



CLC **Genomics** Workbench

USER MANUAL

Manual for
CLC Genomics Workbench 25.0.2
Windows, macOS and Linux

June 11, 2025

This software is for research purposes only.

QIAGEN Aarhus A/S
Kalkværksvej 5, 11.
DK - 8000 Aarhus C
Denmark



Contents

I	Introduction	14
1	Introduction to CLC Genomics Workbench	15
1.1	Contact information and citation	16
1.2	Download and installation	17
1.3	System requirements	21
1.4	Workbench Licenses	23
1.5	Plugins	39
1.6	Network configuration	41
II	Core Functionalities	43
2	User interface	44
2.1	View Area	45
2.2	Zoom functionality in the View Area	55
2.3	Toolbox panel	56
2.4	Processes tab and Status bar	59
2.5	History and Element Info views	61
2.6	Workspace	63
2.7	List of shortcuts	64
3	Data management and search	67
3.1	Navigation Area	68
3.2	Working with non-CLC format files	77
3.3	Customized attributes on data locations	79
3.4	Searching for data in CLC Locations	83

3.5	Backing up data from the CLC Workbench	88
3.6	Working with AWS S3 using the Remote Files tab	89
4	User preferences and settings	94
4.1	General preferences	95
4.2	View preferences	96
4.3	Data preferences	98
4.4	Advanced preferences	98
4.5	Export/import of preferences	101
4.6	Side Panel view settings	101
5	Printing	105
5.1	Selecting which part of the view to print	106
5.2	Page setup	107
5.3	Print preview	109
6	Connections to other systems	110
6.1	CLC Server connection	110
6.2	AWS Connections	114
7	Importing data	119
7.1	Standard import	120
7.2	Import tracks	121
7.3	Import NGS Reads	131
7.4	Import other high-throughput sequencing data	149
7.5	Import RNA spike-in controls	155
7.6	Import Primer Pairs	156
8	Exporting data and graphics	158
8.1	Data export	159
8.2	Export graphics to files	180
8.3	Export graph data points to a file	184
8.4	Copy/paste view output	185
9	Working with tables	187

9.1	Table view settings and column ordering	188
9.2	Filtering tables	190
10	Data download	195
10.1	Search for Sequences at NCBI	195
10.2	Search for PDB Structures at NCBI	198
10.3	Search for Sequences in UniProt (Swiss-Prot/TrEMBL)	201
10.4	Search for Reads in SRA	205
10.5	Sequence web info	214
11	References management	217
11.1	Download Genomes	218
11.2	QIAGEN Sets	221
11.3	Reference Data Sets and defining Custom Sets	224
11.4	Storing, managing and moving reference data	233
12	Running tools, handling results and batching	238
12.1	Running tools	238
12.2	Handling results	243
12.3	Batch processing	244
13	Metadata	249
13.1	Creating metadata tables	251
13.2	Associating data elements with metadata	257
13.3	Working with data and metadata	261
13.4	Moving, copying and exporting metadata	268
14	Workflows	270
14.1	Creating and editing workflows	272
14.2	Workflow elements	292
14.3	Launching workflows individually and in batches	325
14.4	Advanced workflow batching	338
14.5	Template workflows	342
14.6	Managing workflows	358

14.7	QIAseq Panel Analysis Assistant	368
III	Basic sequence analysis	379
15	Viewing and editing sequences	380
15.1	Sequence Lists	381
15.2	View sequences	387
15.3	Working with annotations	399
15.4	Element information	408
15.5	View as text	409
16	BLAST search	411
16.1	Running BLAST searches	412
16.2	Output from BLAST searches	419
16.3	Local BLAST databases	425
16.4	Manage BLAST databases	428
16.5	Bioinformatics explained: BLAST	429
17	3D Molecule Viewer	436
17.1	Importing molecule structure files	437
17.2	Viewing molecular structures in 3D	441
17.3	Customizing the visualization	442
17.4	Tools for linking sequence and structure	451
17.5	Align Protein Structure	455
17.6	Generate Biomolecule	460
18	General sequence analyses	463
18.1	Annotate with GFF/GTF/GVF file	463
18.2	Extract sequences	466
18.3	Shuffle sequence	467
18.4	Dot plots	469
18.5	Local complexity plot	477
18.6	Sequence statistics	477
18.7	Join Sequences	482

18.8	Pattern discovery	483
18.9	Motif Search	485
18.10	Create motif list	490
19	Nucleotide analyses	492
19.1	Convert DNA to RNA	492
19.2	Convert RNA to DNA	493
19.3	Reverse complements of sequences	493
19.4	Translation of DNA or RNA to protein	494
19.5	Find open reading frames	496
20	Protein analyses	499
20.1	Protein charge	499
20.2	Antigenicity	501
20.3	Hydrophobicity	502
20.4	Download Pfam Database	507
20.5	Pfam domain search	507
20.6	Find and Model Structure	510
20.7	Secondary structure prediction	518
20.8	Protein report	519
20.9	Reverse translation from protein into DNA	522
20.10	Proteolytic cleavage detection	525
21	Primers	529
21.1	Primer design - an introduction	530
21.2	Setting parameters for primers and probes	532
21.3	Graphical display of primer information	534
21.4	Output from primer design	536
21.5	Standard PCR	537
21.6	Nested PCR	541
21.7	TaqMan	542
21.8	Sequencing primers	544
21.9	Alignment-based primer and probe design	545

21.10 Analyze primer properties	550
21.11 Find binding sites and create fragments	551
21.12 Order primers	555
22 Sequencing data analyses	557
22.1 Importing and viewing trace data	558
22.2 Trim sequences	559
22.3 Assemble sequences	562
22.4 Assemble sequences to reference	564
22.5 Sort sequences by name	567
22.6 Add sequences to an existing contig	570
22.7 View and edit contigs and read mappings	571
22.8 Reassemble contig	580
22.9 Secondary peak calling	581
22.10 Extract Consensus Sequence	583
23 Cutting and cloning	587
23.1 Restriction site analyses	587
23.2 Restriction enzyme lists	596
23.3 Restriction Based Cloning	598
23.4 Homology Based Cloning	606
23.5 Gateway cloning	615
23.6 Gel electrophoresis	622
24 Sequence alignment	625
24.1 Create an alignment	625
24.2 View alignments	630
24.3 Edit alignments	636
24.4 Join alignments	640
24.5 Pairwise comparison	641
25 Phylogenetic trees	646
25.1 K-mer Based Tree Construction	648
25.2 Create tree	649

25.3	Model Testing	650
25.4	Maximum Likelihood Phylogeny	652
25.5	Tree Settings	660
25.6	Metadata and phylogenetic trees	669
26	RNA structure	675
26.1	RNA secondary structure prediction	676
26.2	View and edit secondary structures	682
26.3	Evaluate structure hypothesis	690
26.4	Structure scanning plot	692
26.5	Bioinformatics explained: RNA structure prediction by minimum free energy minimization	694
IV	High-throughput sequencing	700
27	Tracks	701
27.1	Track types	703
27.2	Track lists	704
27.3	Working with tracks	708
27.4	Reference data as tracks	716
27.5	Modify tracks	716
27.6	Filter tracks	724
27.7	Merge tracks	726
27.8	Convert tracks	728
27.9	Graph tracks	731
28	Prepare sequencing data	738
28.1	QC for Sequencing Reads	738
28.2	Trim Reads	743
28.3	Demultiplex Reads	758
29	Quality control for resequencing analysis	766
29.1	QC for Targeted Sequencing	766
29.2	Target Region Coverage Analysis	777

29.3	QC for Read Mapping	780
29.4	Whole Genome Coverage Analysis	786
30	Read mapping	789
30.1	Map Reads to Reference	790
30.2	Map Long Reads to Reference	801
30.3	Reads tracks and stand-alone read mappings	804
30.4	Local Realignment	820
30.5	Merge Read Mappings	827
30.6	Remove Duplicate Mapped Reads	828
30.7	Extract Consensus Sequence	831
31	Variant detection	835
31.1	Variant Detection tools	836
31.2	Fixed Ploidy Variant Detection	841
31.3	Low Frequency Variant Detection	843
31.4	Basic Variant Detection	843
31.5	Variant Detection - filters	844
31.6	Variant Detection - the outputs	852
31.7	Fixed Ploidy and Low Frequency Detection tools: detailed descriptions	861
31.8	Copy Number Variant Detection	869
31.9	Identify Known Mutations from Sample Mappings	885
31.10	InDels and Structural Variants	890
31.11	Structural Variant Caller for Long Reads	905
32	Resequencing	912
32.1	Variant filtering	913
32.2	Variant annotation	917
32.3	Variants comparison	922
32.4	Variant quality control	927
32.5	Functional consequences	929
32.6	Create Consensus Sequences from Variants	946
33	RNA-Seq and Small RNA analysis	948

33.1	RNA-Seq normalization	949
33.2	Create Expression Browser	951
33.3	miRNA analysis	960
33.4	RNA-Seq Tools	971
33.5	Expression Plots	1018
33.6	Differential Expression	1036
34	Microarray analysis	1061
34.1	Experimental design	1062
34.2	Transformation and normalization	1075
34.3	Quality control	1078
34.4	Feature clustering	1090
34.5	Statistical analysis - identifying differential expression	1098
34.6	Annotation tests	1105
34.7	General plots	1111
35	De Novo sequencing	1117
35.1	De Novo Assembly	1117
35.2	Map Reads to Contigs	1138
35.3	De Novo Assemble Long Reads	1141
35.4	Polish Contigs with Reads	1144
36	Epigenomics analysis	1148
36.1	Histone Chip-Seq	1148
36.2	ChIP-Seq Analysis	1151
36.3	Bisulfite Sequencing	1157
36.4	Advanced Peak Shape Tools	1175
37	Utility tools	1181
37.1	Extract Annotated Regions	1181
37.2	Extract Reads	1183
37.3	Filter on Custom Criteria	1187
37.4	Merge Overlapping Pairs	1189
37.5	Combine Reports	1192

37.6 Create Sample Report	1198
37.7 Create Report from Table	1205
37.8 Modify Report Type	1207
37.9 Track tools	1213
37.10 Create Sequence List	1213
37.11 Update Sequence Attributes in Lists	1213
37.12 Split Sequence List	1217
37.13 Subsample Sequence List	1219
37.14 Rename Elements	1221
37.15 Rename Sequences in Lists	1225
38 Legacy tools	1230
38.1 Correct Long Reads	1230
V Appendix	1234
A Use of multi-core computers	1235
B Graph preferences	1237
C BLAST databases	1239
C.1 Peptide sequence databases	1239
C.2 Nucleotide sequence databases	1239
C.3 Adding more databases	1240
D Proteolytic cleavage enzymes	1242
E Restriction enzymes database configuration	1244
F Technical information about modifying Gateway cloning sites	1245
G IUPAC codes for amino acids	1246
H IUPAC codes for nucleotides	1247
I Formats for import and export	1248

I.1	List of bioinformatic data formats	1248
I.2	List of graphics data formats	1256
J	SAM/BAM/CRAM export format specification	1257
J.1	Flags	1258
K	Gene expression annotation files and microarray data formats	1262
K.1	GEO (Gene Expression Omnibus)	1262
K.2	Affymetrix GeneChip	1265
K.3	Illumina BeadChip	1267
K.4	Gene ontology annotation files	1268
K.5	Generic expression and annotation data file formats	1268
L	Custom codon frequency tables	1272
M	Comparison of track comparison tools	1273
	Bibliography	1275

Part I

Introduction

Chapter 1

Introduction to *CLC Genomics Workbench*

Contents

1.1	Contact information and citation	16
1.2	Download and installation	17
1.2.1	General information about installing and upgrading Workbenches	18
1.2.2	Installation on Microsoft Windows	19
1.2.3	Installation on macOS	20
1.2.4	Installation on Linux with an installer	20
1.3	System requirements	21
1.3.1	Limitations on maximum number of cores	22
1.4	Workbench Licenses	23
1.4.1	Request an evaluation license	24
1.4.2	Download a license using a license order ID	26
1.4.3	Import a license from a file	28
1.4.4	Upgrade license	28
1.4.5	Configure license manager connection	31
1.4.6	Viewing or updating license information	35
1.4.7	Download a static license on a non-networked machine	36
1.4.8	Viewing mode	37
1.4.9	Start in safe mode	39
1.5	Plugins	39
1.5.1	Install	39
1.5.2	Uninstall	40
1.5.3	Updating plugins	41
1.6	Network configuration	41

Welcome to *CLC Genomics Workbench 25.0.2* — a software package supporting your daily bioinformatics work.

CLC Genomics Workbench 25.0.2 **is for research purposes only.**

The *CLC Genomics Workbench* provides an easy-to-use graphical interface for running bioinformatics analyses. Tools can be run individually, or chained together in a workflow, making running

complex analyses simple and efficient. The functionality of the *CLC Genomics Workbench* can also be extended using plugins. The built-in Plugin Manager provides an up-to-date listing. A list is also available on our plugin webpage: <https://digitalinsights.qiagen.com/products-overview/plugins/>.

Supporting documentation and links for the *CLC Genomics Workbench* can be found under the **Help** menu in the top toolbar. Of particular note when getting started:

- The built-in Workbench user manual can be opened by choosing the **Help** option or by clicking on the **F1** key.
- Manuals for installed plugins can be accessed under the **Plugin Help** option.
- The **Online Tutorials** option opens our tutorials webpage in a browser. Tutorials offer hands-on examples of how to use features of the *CLC Genomics Workbench*. Alternatively, click on the following link to visit that webpage: <https://digitalinsights.qiagen.com/support/tutorials/>.

Watch product specialists demonstrates our software in the videos offered via our **Online presentations** area: <https://tv.qiagenbioinformatics.com/>.

The latest version of this user manual can be found in pdf and html formats at <https://digitalinsights.qiagen.com/technical-support/manuals/>

The *CLC Genomics Workbench* is being constantly developed and improved. A detailed list of new features, improvements, bug fixes, and changes for the current version of *CLC Genomics Workbench* can be found at <https://digitalinsights.qiagen.com/technical-support/latest-improvements/>.

1.1 Contact information and citation

CLC Genomics Workbench is developed by:

QIAGEN Aarhus A/S
Kalkværksvej 5, 11.
DK - 8000 Aarhus C
Denmark

<https://digitalinsights.qiagen.com/>

Email: ts-bioinformatics@qiagen.com

The QIAGEN Aarhus team is continuously improving *CLC Genomics Workbench* with your interests in mind. We welcome all requests and feedback from users, as well as suggestions for new features or more general improvements to the program.

Getting help via the Workbench If you encounter a problem or need help understanding how *CLC Genomics Workbench* works, and the license you are using is covered by our Maintenance, Upgrades and Support (MUS) program (<https://digitalinsights.qiagen.com/>

[technical-support/maintenance-and-support/](#)), please contact us by clicking on the Support button at the right hand side of the top toolbar (figure 1.1) or by choosing **Contact Support** under the **Help** menu.

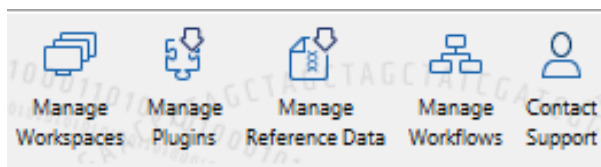


Figure 1.1: Contact our Support team by clicking on the button at the right hand side of the top Toolbar

This will open a dialog where you can enter your contact information, and a text field for writing the question or problem you have. On a second dialog you will be given the chance to attach screenshots or even small datasets that can help explain or troubleshoot the problem. When you send a support request this way, it will automatically include helpful technical information about your installation and your license information so that you do not have to look this up yourself. Our support staff will reply to you by email.

Other ways to contact the support team You can also contact the support team by email: ts-bioinformatics@qiagen.com

Please provide your contact information, your license information, some technical information about your installation, and describe the question or problem you have. You can also attach screenshots or even small data sets that can help explain or troubleshoot the problem.

Information about the license(s) being used by a *CLC Workbench* and any installed modules can be found by opening the License Manager:

Help | License Manager...

Information about MUS cover on particular licenses is provided in your myCLC account: <https://secure.clcbio.com/myclc/login>.

How to cite us To cite a CLC Workbench or Server product, use the name of the product, the version number. For example QIAGEN CLC Main Workbench 24.0 or QIAGEN CLC Genomics Workbench 24.0. If a location is required by the publisher of the publication, use (QIAGEN, Aarhus, Denmark). Our website is <https://digitalinsights.qiagen.com/>.

Further details about citing QIAGEN Digital Insights software can be found in our FAQ at

<https://qiagen.secure.force.com/KnowledgeBase/KnowledgeNavigatorPage?id=kA41i000000L63hCAC>

1.2 Download and installation

Software installers for Windows, macOS and Linux systems for the current *CLC Genomics Workbench* release can be downloaded from <https://digitalinsights.qiagen.com/downloads/product-downloads/>.

More ways to get installers are described in the Frequently Asked Question entry "Where can I get installer files for the software?": <https://qiagen.secure.force.com/KnowledgeBase/KnowledgeNavigatorPage?id=kA41i000000L5uQCAS>

To check for available updates from within the software, go to the menu option: **Help | Check for Updates...** (🔍).

General information about running software installers, including differences between upgrading to a new minor version compared to upgrading to a new major version, are covered in section 1.2.1. Detailed instructions for running the software installer in interactive mode on each supported operating system then follows.

Information about running the software installers in console mode and silent mode are provided in the Workbench Deployment manual at https://resources.qiagenbioinformatics.com/manuals/workbenchdeployment/current/index.php?manual=Installation_modes_console_silent.html.

1.2.1 General information about installing and upgrading Workbenches

When a Workbench installer is run, it performs the following tasks:

1. **Extracts and copies files to the installation directory** The Workbench software is installed into a directory. It is self contained. The suggested folder name to install into reflects the software name and the major version line. For example, for a *CLC Genomics Workbench* with major version 25, the default installation location offered on each platform would be:

Windows C:\Program files\CLC Genomics Workbench 25

macOS /Applications/CLC Genomics Workbench 25

Linux /opt/CLCGenomicsWorkbench 25

To install the software into central locations, like those listed above, generally requires administrator rights. Administrator rights will also be needed to install licenses and plugins for installations in central locations. The software can be installed to another location, if desired. When only a single person will use the software, this can be useful. Installing to an area they have permission to write to means that licenses and plugins can then be installed without needing administrator rights.

General recommendations for installation locations

- **For minor updates**, you will be asked whether you wish to:
 - **Update the existing installation** *Generally recommended for minor updates.* New files will be installed into the same directory as the existing installation. Licensing information and installed plugins remain in place from the installation already present.
OR
 - **Install to a different directory.** Configuration will be needed after installation. E.g. licensing needs to be configured, any desired plugins will need to be installed, etc.
- **For major updates.** The suggested installation directory will reflect the new major version number of the software. *Please do not install a new major version into the same folder as an existing, older version of the Workbench.* Configuration will be needed after

installation. E.g. licensing needs to be configured, any desired plugins will need to be installed, etc.

2. **Sets the amount of memory** The installer investigates the amount of RAM on the machine during installation and sets the amount of memory that the Workbench can use.
3. **Establishes shortcuts (optional)** On Windows and Mac systems, an option is provided during installation to create a shortcut for starting the Workbench. On Linux systems, this option is also presented, but it has no effect.

On Macs without Rosetta present on the system, the option of installing it is offered during the installation process. Rosetta enables Intel-based features to run on Apple Silicon Macs. While not needed for the majority of tools, some require it, for example De Novo Assembly, BLAST, Sample Reads and tools for analyzing small RNA.

Updating workflows after upgrading a *CLC Workbench* is described in section [14.6.1](#).

1.2.2 Installation on Microsoft Windows

To install to a central location, the installer must be run in *administrator mode*. On Windows, right-click on the of the downloaded installer file and choose "Run as Administrator".

To install to a location you have write permission to, double click the icon of the downloaded installer file.

Then walk through the following steps. (The exact order options are presented may differ to that described.)

- Read and accept the License agreement and click **Next**.
- Unless you are installing a minor update to the same folder as an existing installation, you will be prompted to choose where you would like to install the Workbench. If you are upgrading from an earlier version, please refer to section [1.2.1](#) for information about installing to an existing or different directory. Click on **Next**.
- Choose where you want the program's shortcuts to be placed. Click on **Next**.
- Choose if you would like to associate .clc files to the *CLC Genomics Workbench*. If you check this option, double-clicking a file with a ".clc" extension will open the *CLC Genomics Workbench*.
- Choose if a desktop icon should be created, and choose whether clc://URLs should be opened by this program by default. Click on **Next**.
- Wait for the installation process to complete, and then choose whether you would like to launch *CLC Genomics Workbench* right away. Click on **Finish**.

When the installation is complete the program can be launched from the Start Menu or from one of the shortcuts you chose to create.

1.2.3 Installation on macOS

To install to a location you have write permission to, double click the icon of the downloaded installer file.

Then walk through the following steps. (The exact order options are presented may differ to that described.)

- Read and accept the License agreement and click **Next**.
- Choose where you would like to install the application. If you are upgrading from an earlier version, please refer to section [1.2.1](#) for information about installing to an existing or different directory. Click on **Next**.
- Specify other options associated with the installation such as whether a desktop icon should be created, whether the software should open clc:// URLs. whether .clc files should be associated with the software and whether it should be added to the dock. Click on **Next**.
- Wait for the installation process to complete, choose whether you would like to launch *CLC Genomics Workbench* right away, and click on **Finish**.

On Apple Silicon Macs without Rosetta present on the system, the option of installing it is offered during the installation process. Rosetta enables Intel-based features to run on Apple Silicon Macs. While not needed for the majority of tools, some require it, for example De Novo Assembly, BLAST, Sample Reads and tools for analyzing small RNA.

When the installation is complete, the program can be launched from the dock, if present there, or by clicking on the desktop shortcut if you chose to create one. The software can also be launched from within the installation folder.

1.2.4 Installation on Linux with an installer

Navigate to the directory containing the installer and launch it. This can be done by running a command similar to:

```
# sh CLCGenomicsWorkbench_25_0_2_64.sh
```

To install to a central location such as /opt or /usr/local, you will normally need to run the above command using sudo. If you do not have sudo privileges you can choose to install in your home directory, or any other location you have write permission for.

Then walk through the following steps. (The exact order options are presented may differ to that described.)

- Read and accept the License agreement and click **Next**.
- Choose where you would like to install the application. If you are upgrading from an earlier version, please refer to section [1.2.1](#) for information about installing to an existing or different directory. Click on **Next**.

- Choose where you would like to create symbolic links to the program. Click on **Next**.
DO NOT create symbolic links in the same location as the application.
Symbolic links should be installed in a location which is included in your environment PATH. For a system-wide installation you can choose for example /usr/local/bin. If you do not have root privileges you can create a 'bin' directory in your home directory and install symbolic links there. You can also choose not to create symbolic links.
- Wait for the installation process to complete and click on **Finish**.

If you choose to create symbolic links in a location which is included in your PATH, the program can be executed by running the command:

```
# clcgenomicswb25
```

Otherwise you start the application by navigating to the location where you choose to install it and running the command:

```
# ./clcgenomicswb25
```

1.3 System requirements

- Windows: Supported versions of Windows 10, Windows 11, Windows Server 2016, Windows Server 2019, Windows Server 2022 and Windows Server 2025
- Mac: macOS 13, 14 and 15
- Linux: RHEL 8 and later and supported versions of SUSE Linux Enterprise Server 12.5 and later. The software is expected to run without problem on other recent Linux systems, but we do not guarantee this. To use BLAST related functionality, libnsl.so.1 is required.
- 64 bit operating system
- 16 GB RAM recommended (8 GB RAM required)
- 1024 x 768 display required
- 1600 x 1200 display recommended

See section [1.3.1](#) for information pertaining to working on systems with >64 cores.

Memory and CPU settings for mapping reads

For mapping reads to the human genome (3.2 gigabases), or genomes of a similar size, 16 GB RAM is required. Smaller systems can be used when mapping to small genomes.

Larger amounts of memory can help the overall speed of the analysis when working with large datasets, but little gain is expected above about 32 GB of RAM.

Increasing the number of cpus can decrease the time a read mapping takes, however performance gain is expected to be limited above approximately 40 threads.

System requirements for de novo assembly

De novo assembly may need more memory than stated above - this depends both on the number of reads, error profile and the complexity and size of the genome.

When assembling PacBio HiFi reads with the De Novo Assemble Long Reads tool, at least 32 GB RAM is recommended.

For examples of the memory usage of various data sets when using the De Novo Assembly tool, see https://resources.qiagenbioinformatics.com/white-papers/White_paper_on_de_novo_assembly_4.pdf.

Other requirements for long read analysis

The following tools for working with long reads require an AMD/Intel CPU that supports AVX2 or an Apple M series CPU:

- De Novo Assemble Long Reads
- Polish Contigs with Reads
- Correct Reads (legacy)

Polish Contigs with Reads and Correct Long Reads (legacy) use Racon [Vaser et al., 2017], for which the following is noted: *Memory consumption of polishing equals the size of the mandatory input files, while the memory is somewhat larger for error correction.* See <https://complex.zesoi.fer.hr/en/blog-en/58-racon-1-0-release>.

System requirements for 3D viewers

A graphics card that supports OpenGL 2.0.

Note: 3D rendering is only supported when the *CLC Genomics Workbench* is installed on the same machine the viewer is opened on. Indirect rendering (such as X11 forwarding through ssh, remote desktop connection/VNC, and running in virtual machines) is not supported.

CLC data location requirements

Requirements for CLC data locations are provided in section 3.1.2.

1.3.1 Limitations on maximum number of cores

Most modern CPUs implement hyper threading or a similar technology that makes each physical CPU core appear as two logical cores on a system. In this manual the term "core" always refers to a logical core unless otherwise stated.

Licensing: Static licenses can be used on systems with 64 or fewer logical cores. On systems with more than 64 logical cores a network license is needed. See <https://digitalinsights.qiagen.com/licensing/>.

Performance: Tools that take advantage of multiple cores do not scale linearly with high numbers of cores. If you plan to use a large system (>64 cores), a *CLC Genomics Server* with job

nodes running on virtual machines provides potential to use more of the compute capacity in a controlled manner than a *CLC Genomics Workbench*. Jobs submitted to a *CLC Genomics Server* job node setup can be run in parallel, with appropriate CPU limits configurable for each node. For further details, see <https://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=Introduction.html>.

1.4 Workbench Licenses

When you start up the *CLC Genomics Workbench* for the first time on your system, or after installing a new major release, the **License Assistant**, shown in figure 1.2, will be presented to you. The **License Assistant** can also be launched during an active Workbench session by clicking on the "Upgrade Workbench License" button at the bottom of the **License Manager**. The **License Manager** can be started up using the Workbench menu item:

Help | License Manager ()

You need a license...

In order to use this application you need a valid license.
Please choose how you would like to obtain a license for your workbench.

- ☒ **Request an evaluation license**
Try out the application for 14 days. A static license will be downloaded to your local machine. Use with remote or virtual machines is not supported.
- ☐ **Download a license**
Use a license order ID to download a static license.
- ☐ **Import a license from a file**
Import a static license from an existing license file.
- ☐ **Upgrade from an existing Workbench installation**
Upgrade an existing license for an older version of the software. Your license must be covered by Maintenance, Upgrades and Support to use this option.
- ☐ **Configure license manager connection**
Configure a connection to a CLC Network License Manager that hosts network license(s) for this product, or update or disable an existing connection configuration.

Figure 1.2: The License Assistant provides access to licensing options.

The options available in the **License Assistant** window are described in brief below, and then in detail in the sections that follow.

- **Request an evaluation license** Request a fully functional, time-limited static license.
- **Download a license** Use the license order ID provided when you purchase the software to download and install a static license file.
- **Import a license from a file** Import an existing static license file, for example a file downloaded from the license download webpage.
- **Upgrade from an existing Workbench installation** If you have used a previous version of

the *CLC Genomics Workbench*, and you are entitled to upgrade to a new major version, select this option to upgrade your static license file.

- **Configure license manager connection** If your organization has a *CLC Network License Manager*, select this option to configure the connection to it.

Select the appropriate option and then click on the **Next** button.

To use the **Request an evaluation license**, **Download a license** or the **Upgrade from an existing Workbench installation** options, your machine must be able to access the external network. If this is not the case, please see section 1.4.7.

When using a *CLC Genomics Workbench* installed in a central location on your system, you must be running the program in administrative mode to license the software. On Linux and Mac, this means you must be logged in as an administrator. On Windows, you can right-click the program shortcut and choose "Run as Administrator".

If you do not have a license order ID or access to a license, you can still use the Workbench in **Viewing Mode**. See section 1.4.8 for further information about this.

Note: Static licenses are tied to the host ID of the machine they were downloaded to. If your license is covered by Maintenance, Upgrades and Support (MUS), please contact our Support team (ts-bioinformatics@qiagen.com) if you need to start using a different machine for working with the *CLC Genomics Workbench*.

1.4.1 Request an evaluation license

We offer a fully functional version of the *CLC Genomics Workbench* free of charge for a 14 day period for evaluation purposes. The 14 day period commences when the evaluation license is downloaded. If you have questions about *CLC Genomics Workbench* features or product licensing options, please send an email to bioinformaticssales@qiagen.com.

When you choose the option **Request an evaluation license**, you will see the dialog shown in figure 1.3.

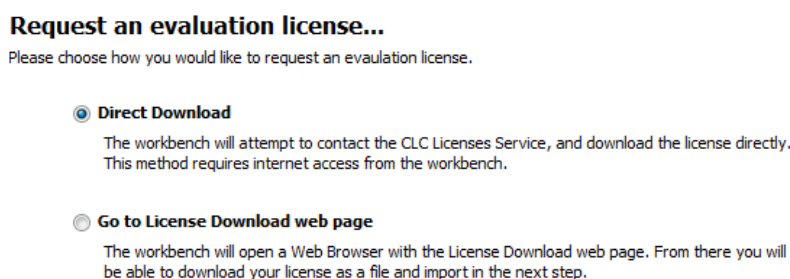


Figure 1.3: Choose between downloading a license directly, or opening the license download form in a web browser.

In this dialog, there are two options:

- **Direct Download.** Download the license directly. This method requires that the Workbench has access to the external network.

- **Go to CLC License Download web page.** The online license download form will be opened in a web browser. This option is suitable for when downloading a license for use on another machine that does not have access to the external network, and thus cannot access the QIAGEN Aarhus servers.

After selecting your method of choice, click on the button labeled **Next**.

Direct download

After choosing the **Direct Download** option and clicking on the button labeled **Next**, a dialog similar to that shown in figure 1.4 will appear if the license is successfully downloaded and installed.

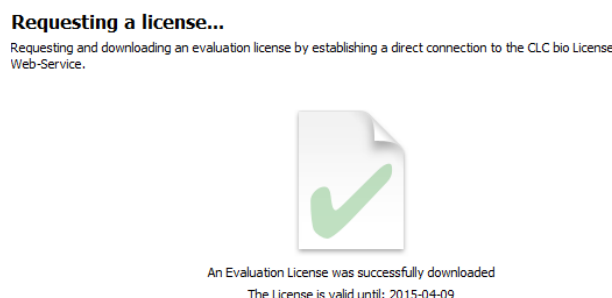


Figure 1.4: A license has been successfully downloaded and installed for use.

When the license has been downloaded and installed, the **Next** button will be enabled.

If there is a problem, a dialog will appear indicating this.

Go to license download web page

After choosing the **Go to CLC License Download web page** option and clicking on the button labeled **Next**, the license download form will be opened in a web browser, as shown in figure 1.5.

Download a license

This page can be used to download a license if you are not able to contact the license server directly from your CLC Workbench.

You have requested the following license:

License Order-ID:	CLC-LICENSE-SRENMINSTED-0D43CA9EDF-60000D844A4C0C48XXXX
Product:	
Product Version:	1
Host-ID(s):	0SEDE85CEFD, A81AEFF919F, 2A378AC18863
Host name:	laptop-32

To download your license, please click the button below.

Download License

If the request is successful a file containing the license will be downloaded to your computer.

To begin using your license you must import the file into the license assistant wizard. Do this by clicking on the **Choose License File** button and locate the file on your computer.

Figure 1.5: The license download form opened in a web browser.

Click on the **Download License** button and then save the license file.

Back in the Workbench window, you will now see the dialog shown in 1.6.

Click on the **Choose License File** button, find the saved license file and select it. Then click on the **Next** button.

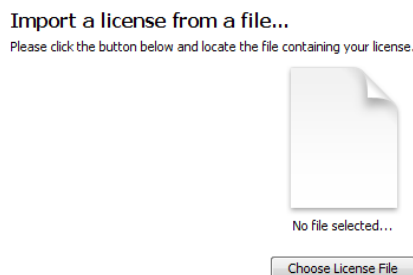


Figure 1.6: Importing the license file downloaded from the web page.

Accepting the license agreement

The End User License Agreement (EULA) must be read and accepted as part of the installation process. Please read the EULA text carefully, and if you agree to it, check the box next to the text **I accept these terms**. If further information is requested from you, please fill this in before clicking on the **Finish** button.

1.4.2 Download a license using a license order ID

Using a license order ID, you can download a license file via the Workbench or using an online form. When you have chosen this option and clicked on the **Next** button, you will see the dialog shown in 1.7. Enter your license order ID into the License Order ID text field. (The ID can be pasted into the box after copying it and then right clicking in the text field and choosing Paste from the context menu, or using a key combination like Ctrl+V, or on a Mac, ⌘ + V).

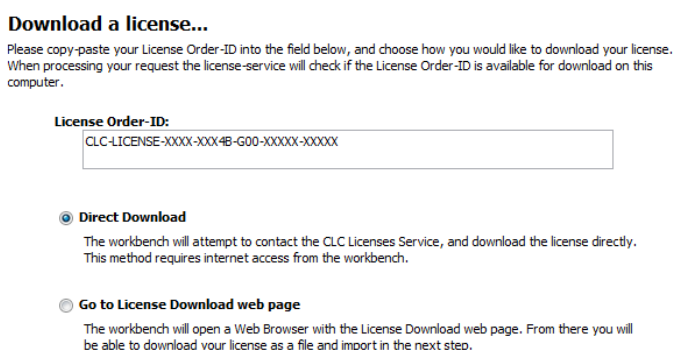


Figure 1.7: Enter a license order ID into the text field and then click on the Next button.

In this dialog, there are two options:

- **Direct Download.** Download the license directly. This method requires that the Workbench has access to the external network.
- **Go to CLC License Download web page.** The online license download form will be opened in a web browser. This option is suitable for when downloading a license for use on another machine that does not have access to the external network, and thus cannot access the QIAGEN Aarhus servers.

After selecting your method of choice, click on the button labeled **Next**.

Direct download

After choosing the **Direct Download** option and clicking on the button labeled **Next**, a dialog similar to that shown in figure 1.8 will appear if the license is successfully downloaded and installed.

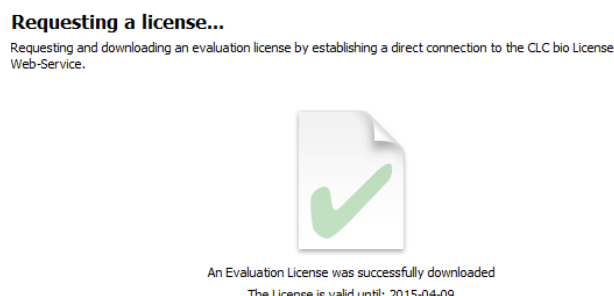


Figure 1.8: A license has been successfully downloaded and installed for use.

When the license has been downloaded and installed, the **Next** button will be enabled.

If there is a problem, a dialog will appear indicating this.

Go to license download web page

After choosing the **Go to CLC License Download web page** option and clicking on the button labeled **Next**, the license download form will be opened in a web browser, as shown in figure 1.9.

The image shows a web browser window with a dark background. The title of the page is "Download a license". Below the title, there is a paragraph of text: "This page can be used to download a license if you are not able to contact the license server directly from your CLC Workbench." followed by "You have requested the following license:". Below this, there is a table with license details:

License Order-ID:	CLC-LICENSE-SRENMINSTED-0D43CA9EDF40000D844A4C0C480000X		
Product:			
Product Version:	1		
Host-ID(s):	05ED85CEFD	A81AEFF919F	2A378AC18863
Host name:	laptop-32		

Below the table, there is a paragraph: "To download your license, please click the button below." followed by a button labeled "Download License". At the bottom, there is a paragraph: "If the request is successful a file containing the license will be downloaded to your computer. To begin using your license you must import the file into the license assistant wizard. Do this by clicking on the Choose License File button and locate the file on your computer."

Figure 1.9: The license download form opened in a web browser.

Click on the **Download License** button and then save the license file.

Back in the Workbench window, you will now see the dialog shown in 1.10.

Click on the **Choose License File** button, find the saved license file and select it. Then click on the **Next** button.

Accepting the license agreement

The End User License Agreement (EULA) must be read and accepted as part of the installation process. Please read the EULA text carefully, and if you agree to it, check the box next to the text **I accept these terms**. If further information is requested from you, please fill this in before clicking on the **Finish** button.

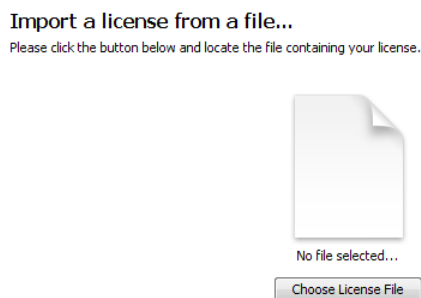


Figure 1.10: Importing the license file downloaded from the web page.

1.4.3 Import a license from a file

If you already have a license file associated with the host ID of your machine, it can be imported using this option.

When you have clicked on the **Next** button, you will see the dialog shown in 1.11.

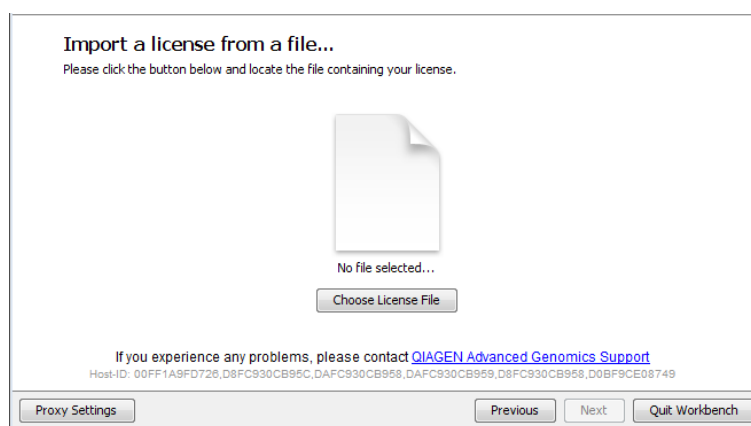


Figure 1.11: Selecting a license file.

Click on the **Choose License File** button, locate the license file and selected it. Then click on the **Next** button.

Accepting the license agreement

The End User License Agreement (EULA) must be read and accepted as part of the installation process. Please read the EULA text carefully, and if you agree to it, check the box next to the text **I accept these terms**. If further information is requested from you, please fill this in before clicking on the **Finish** button.

1.4.4 Upgrade license

The option "Upgrade from an existing Workbench installation" can be convenient when you have been using another version of a licensed Workbench and the license is covered by our Maintenance, Upgrades and Support (MUS) program. Licenses not covered by MUS cannot be updated to support a new major Workbench release line.

If your license is covered our Maintenance, Upgrades and Support (MUS) program but you

experience problems downloading a license for the new version of the software, please contact bioinformaticslicense@qiagen.com.

The Workbench will need direct access to the external network to use this option. If the Workbench cannot connect to the external network directly, please see section 1.4.7.

After selecting the "Upgrade from an existing Workbench installation" option, click on the **Next** button. The Workbench will search for an earlier installation of the same Workbench product you are upgrading to.

If it finds that installation, it will locate the existing license file and show information like that in figure 1.12.

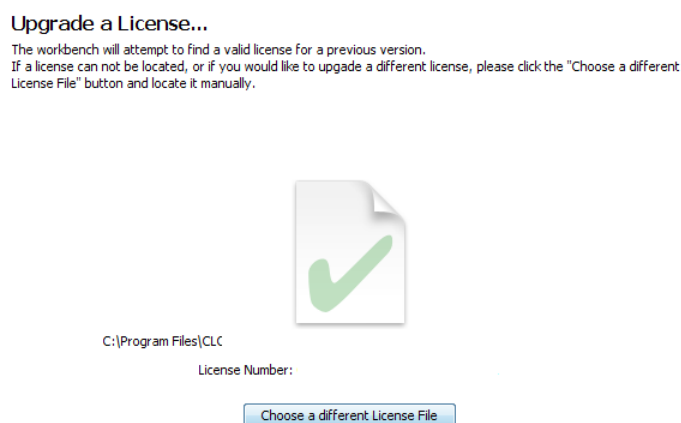


Figure 1.12: An license from an older installation was found.

When you click on the **Next** button, the Workbench checks if you are entitled to upgrade your license. This is done by contacting QIAGEN Aarhus servers.

If the earlier Workbench version could not be found, which can be the case if you have installed to a custom location or are upgrading from one Workbench product to another product replacing it¹, then click on the "Choose a different License File" button. Navigate to where the older license file is, which will be in a subfolder called "licenses" within the installation area of the Workbench you are upgrading from. Select the license file and click on the "Open" button.

If the license selected can be updated, a message similar to that shown in figure 1.13 will be displayed. If there is a problem updating the selected license, a dialog will appear indicating this.

Click on the **Next** button and then choose how to proceed to get the updated license file.

In this dialog, there are two options:

- **Direct Download.** Download the license directly. This method requires that the Workbench has access to the external network.
- **Go to CLC License Download web page.** The online license download form will be opened in a web browser. This option is suitable for when downloading a license for use on another

¹In November 2018, the Biomedical Genomics Workbench was replaced by the CLC Genomics Workbench and a free plugin, Biomedical Genomics Analysis. Licenses for the Biomedical Genomics Workbench covered by MUS at that time can be used to download a valid license for the CLC Genomics Workbench, but the upgrade functionality is not able to automatically find the older license file.

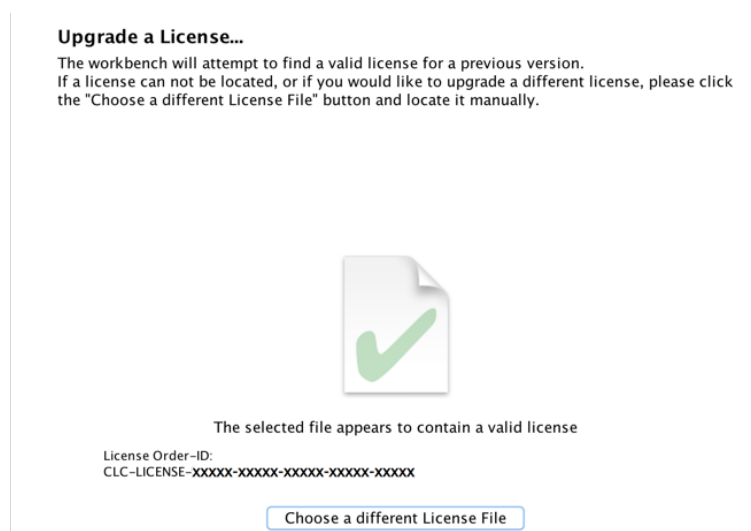


Figure 1.13: A license in an older installation was found.

machine that does not have access to the external network, and thus cannot access the QIAGEN Aarhus servers.

After selecting your method of choice, click on the button labeled **Next**.

Direct download

After choosing the **Direct Download** option and clicking on the button labeled **Next**, a dialog similar to that shown in figure 1.14 will appear if the license is successfully downloaded and installed.

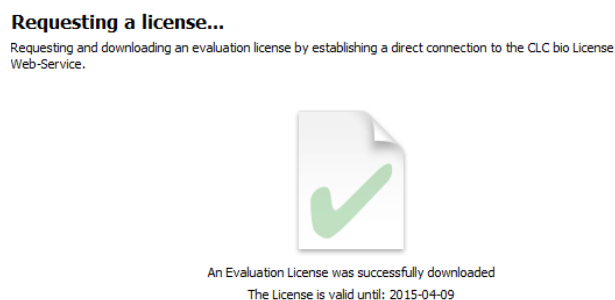


Figure 1.14: A license has been successfully downloaded and installed for use.

When the license has been downloaded and installed, the **Next** button will be enabled. If there is a problem, a dialog will appear indicating this.

Go to license download web page

After choosing the **Go to CLC License Download web page** option and clicking on the button labeled **Next**, the license download form will be opened in a web browser, as shown in figure 1.15.

Click on the **Download License** button and then save the license file.

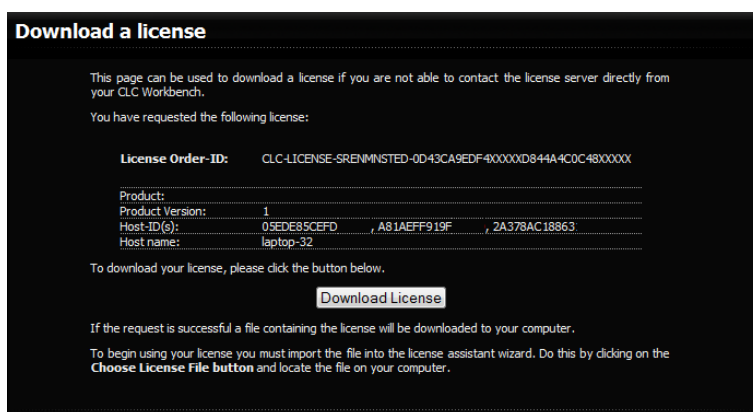


Figure 1.15: The license download form opened in a web browser.

Back in the Workbench window, you will now see the dialog shown in 1.16.

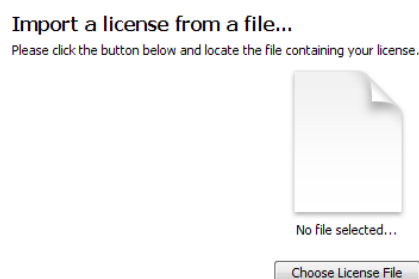


Figure 1.16: Importing the license file downloaded from the web page.

Click on the **Choose License File** button, find the saved license file and select it. Then click on the **Next** button.

Accepting the license agreement

The End User License Agreement (EULA) must be read and accepted as part of the installation process. Please read the EULA text carefully, and if you agree to it, check the box next to the text **I accept these terms**. If further information is requested from you, please fill this in before clicking on the **Finish** button.

1.4.5 Configure license manager connection

If your organization is running a *CLC Network License Manager*, you can configure your Workbench to connect to it to get a license.

To configure a Workbench to connect to a *CLC Network License Manager*, select the **Configure license manager connection** option and click on the **Next** button. A dialog appears, as shown in figure 1.17.

The options in that dialog are:

- **Enable license manager connection.** This box must be checked for the Workbench to contact the *CLC Network License Manager* to get a license for the *CLC Genomics Workbench*.

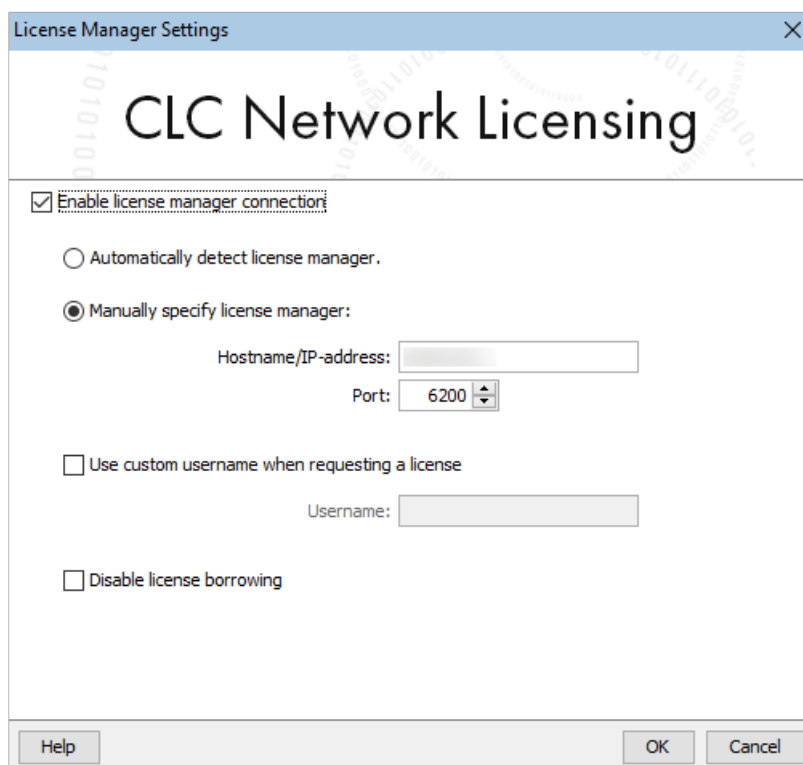


Figure 1.17: Configuring the connection to a CLC Network License Manager.

- **Automatically detect license manager.** By checking this option the Workbench will look for a *CLC Network License Manager* accessible from the Workbench. Automatic server discovery sends UDP broadcasts from the Workbench on port 6200. Available license servers respond to the broadcast. The Workbench then uses TCP communication for to get a license, if one is available. Automatic server discovery works only on local networks and will not work on WAN or VPN connections. Automatic server discovery is not guaranteed to work on all networks. If you are working on an enterprise network on where local firewalls or routers cut off UDP broadcast traffic, then you may need to configure the details of the *CLC Network License Manager* using the **Manually specify license manager** option instead.
- **Manually specify license manager.** Select this option to enter the details of the machine the *CLC Network License Manager* is running on, specifically:
 - **Host name.** The address of the machine the *CLC Network License Manager* is running on.
 - **Port.** The port used by the *CLC Network License Manager* to receive requests.
- **Use custom username when requesting a license.** Optional. When unchecked (the default), the username of the account being used to run the Workbench is the username used when contacting the license manager. When this option is checked, a different username can be entered for that purpose. Note that borrowing licenses is not supported with custom usernames.
- **Disable license borrowing on this computer.** Check this box if you do not want users of this Workbench to borrow a license. See section 1.4.5 for further details.

Releasing Workbench network licenses

Once a network license for a *CLC Workbench* has been obtained, the Workbench must be shut down to release that license for others to use. While it is possible for a license to be pulled from a running Workbench, in practice, that Workbench will immediately retrieve the license just released.

Modules and network licenses

A relevant license is needed to run tools delivered by modules, and to submit jobs to be run on the cloud via a *CLC Workbench*.

To maximize availability, module licenses are first checked out when a job requiring the module is undertaken. Module licenses are checked back in (returned) when the *CLC Workbench* is closed, or four hours after the most recent job requiring that license was launched, whichever is shortest.

Job completion does not depend on the *CLC Workbench* having a module license checked out.

A note about borrowing module licenses: If you plan to borrow network module licenses and the *CLC Network License Manager* will not be continuously accessible from the *CLC Workbench*, then while there is still a connection, launch a job that requires the relevant module. If available, the relevant license will be checked out, and can then be borrowed. When a module license is borrowed, the four hour validity period mentioned above is not relevant. The module licenses are borrowed for the time period that you specify.

See section 1.4.5 for further information about borrowing a license.

Borrowing a license

A *CLC Genomics Workbench* using a network license normally needs to maintain a connection to the *CLC Network License Manager*. However, if allowed by the network license administrator, network licenses can be *borrowed* for offline use for a period of time. While the license is borrowed, there is one less network license available for other users. Borrowed licenses can be returned early.

The Workbench must be connected to the *CLC Network License Manager* at the point when the license is borrowed or returned. The procedure for borrowing a license is:

1. Go to the Workbench menu option:
Help | License Manager...
2. Click on the "Borrow License" tab to display the license borrowing settings (figure 1.18).
3. Select the license(s) that you wish to borrow by clicking in the checkboxes in the Borrow column in the License overview panel.

If you plan to borrow module licenses but they are not listed, start a job that requires that module. This will check out the relevant module license, so that it becomes available to borrow.
4. Choose the length of time you wish to borrow the license(s) for using the drop down list in the Borrow License tab. By default the maximum is 7 days, but network license administrators can specify a lower limit than this.

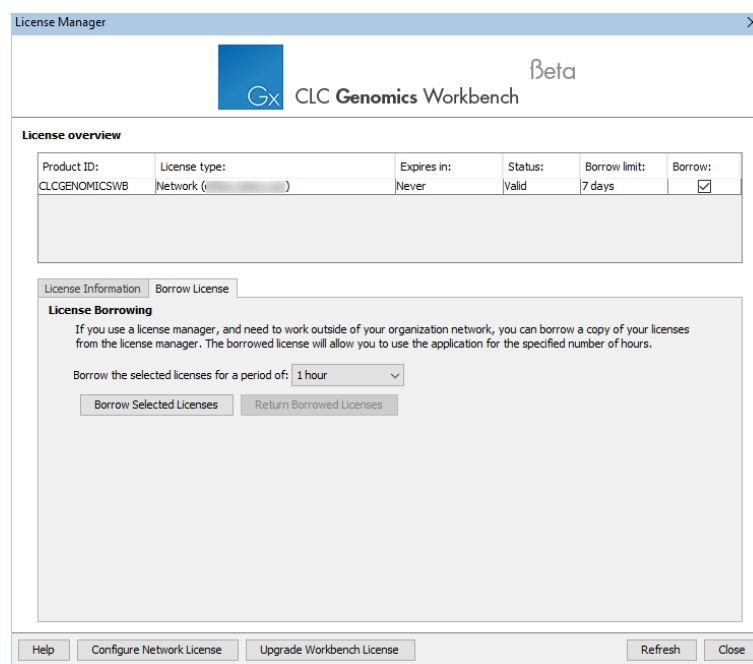


Figure 1.18: Borrowing a license from a CLC Network License Manager.

5. Click **Borrow Selected Licenses**.
6. Close the License Manager when you are done.

You can now go offline and continue working with the *CLC Genomics Workbench*. When the time period you borrowed the license for has elapsed, the network license will be again made available for other users. To continue using *CLC Genomics Workbench* with a license, you will need to connect to the network again so the Workbench can request another license.

You can return borrowed licenses early opening up the **License Manager**, going to the "Borrow License" tab, and clicking on the **Return Borrowed Licenses** button.

Common issues when using a network license

Some issues that may be experienced when using network licenses are:

- No license is available at that point in time.
If all licenses for your product are in use, a message saying "No license available at the moment" will appear (figure 1.19).
After at least one license is returned to the pool, you will be able to run the software and get the necessary license. If running out of licenses is a frequent issue, you may wish to discuss this with your administrator.
Data can be viewed, imported and exported, and very basic analyses launched, by running the Workbench in Viewing Mode. Click on the **Viewing Mode** button in that dialog to launch the Workbench in this mode.
- The connection to the CLC Network License Manager is lost.

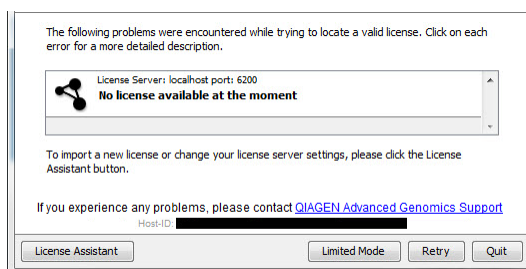


Figure 1.19: When there are no available network licenses for the software, a message appears to indicate this.

If the Workbench connection to the *CLC Network License Manager* is lost, you will see a dialog like that shown in figure 1.20.

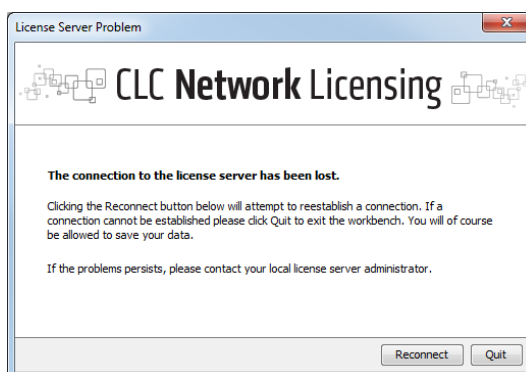


Figure 1.20: This Workbench was unable to establish a connection to obtain a network license.

If you have chosen the option to **Automatically detect license manager** and you have not succeeded in connecting to the *CLC Network License Manager* before, please check with your local IT support that automatic detection is possible at your site. If it is not, you will need to specify the settings, as described earlier in this section.

If you have successfully contacted the *CLC Network License Manager* from your Workbench previously, please contact your local administrator. Common issues include that the *CLC Network License Manager* is not running or that network details have changed.

1.4.6 Viewing or updating license information

To view or edit information about the n license(s) the Workbench is currently using, open the Workbench's License Manager (figure 1.21) using the menu option:

Help | License Manager (📄)

The Workbench's License Manager can be used to:

- See information about its license (e.g. the license type, when it expires, etc.)
- Configure the connection to a *CLC Network License Manager*. Click on the **Configure Network License** button at the lower left corner to open the dialog seen in figure 1.17.
- Upgrade from an evaluation license. Click on the **Upgrade Workbench License** button to open the dialog shown in figure 1.2.

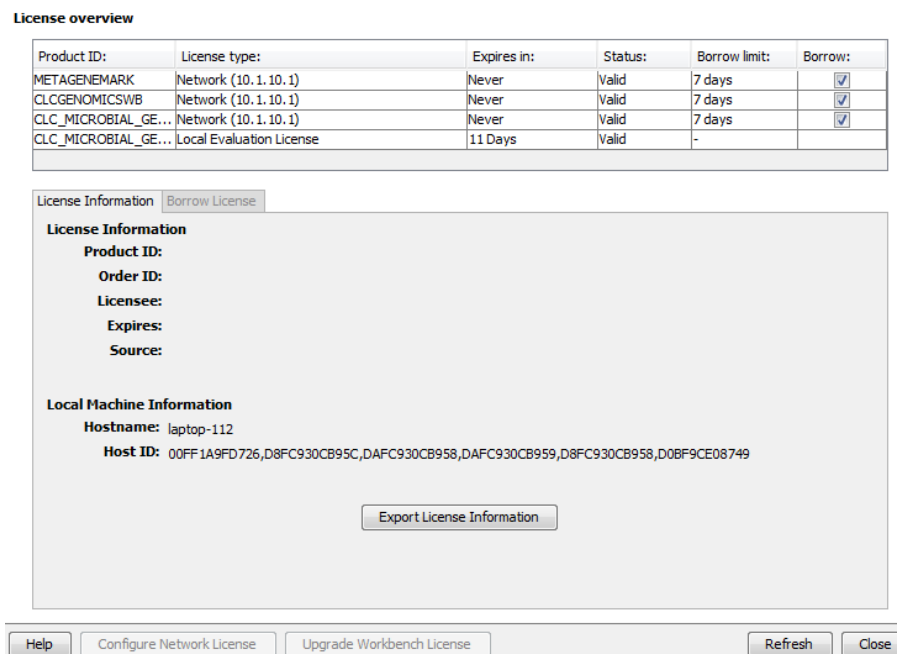


Figure 1.21: License information and license-related functionality is available in the Workbench License Manager.

- Export license information to a text file.
- Borrow a license from a *CLC Network License Manager* when network licenses are in use.

If you wish to switch away from using a network license, click on the button to **Configure Network License** and uncheck the box beside the text **Enable license manager connection** in the dialog. When you restart the Workbench, you can set up the new license as described in section 1.4.

1.4.7 Download a static license on a non-networked machine

To download a static license for a machine that does not have direct access to the external network, you can follow the steps below:

- Install the *CLC Genomics Workbench* on the machine you wish to run the software on.
- Start up the software as an administrative user and find the host ID of the machine that you will run the CLC Workbench on. You can see the host ID of the machine at the bottom of the License Assistant window in grey text, or, if working in Viewing Mode, by launching the **License Manager** from under the Workbench **Help** menu option.
- Make a copy of this host ID such that you can use it on a machine that has internet access.
- Go to a computer with internet access, open a browser window and go to the network license download web page:
<https://secure.clcbio.com/LmxWSv3/GetLicenseFile>
- Paste in your license order ID and the host ID that you noted down in the relevant boxes on the web page.

- Click on 'Download License' and save the resulting .lic file.
- Open the Workbench on your non-networked machine. In the Workbench license manager choose 'Import a license from a file'. In the resulting dialog click on the 'Choose License File' button and then locate and select the .lic file you have just downloaded.

If the License Manager does not start up by default, you can start it up by going to the menu option:

Help | License Manager (🔑)

- Click on the **Next** button and go through the remaining steps to install the license.

1.4.8 Viewing mode

Using a CLC Workbench in Viewing Mode is a free and easy way to access extensive data viewing capabilities, basic bioinformatics analysis tools, as well as import and export functionality.

Data viewing

Any data type supported by the Workbench being used can be viewed in Viewing Mode. Plugins or modules can also be installed when in Viewing Mode, expanding the range of data types supported.

Viewing Mode of the CLC Workbenches can be particularly useful when sharing data with colleagues or reviewers who wish to view and investigate data you have generated but who do not have access to a Workbench license.

Data import, export and analysis in Viewing Mode

When working in Viewing Mode, the Import and Export buttons in the top Toolbar are enabled, and standard import and export functionality for many bioinformatics data types is supported. Tools available can be seen in the Workbench Tools menu, as illustrated in figure 1.22.

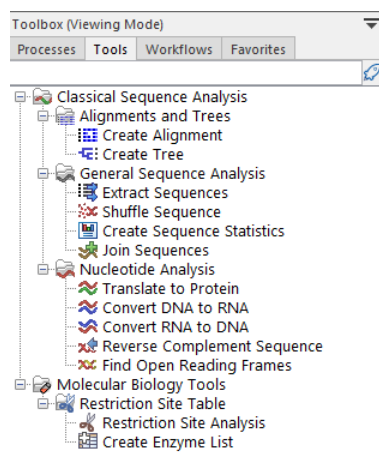


Figure 1.22: Bioinformatics tools available when using Viewing Mode are found under the Tools menu.

Starting a CLC Workbench in Viewing Mode

A button labeled **Viewing Mode** is presented in the Workbench License Manager when a Workbench is started up without a license installed, as shown in figure 1.23. This button is also visible in message windows that appear if a Workbench is started up that has an expired license or that is configured to use a network license but all the available licenses have been checked out by others, as described in section 1.4.5.

Click on the **Viewing Mode** button to start up the Workbench in Viewing Mode.

You need a license...

In order to use this application you need a valid license.
Please choose how you would like to obtain a license for your workbench.

- ☒ **Request an evaluation license**
Try out the application for 14 days. A static license will be downloaded to your local machine. Use with remote or virtual machines is not supported.
- ☐ **Download a license**
Use a license order ID to download a static license.
- ☐ **Import a license from a file**
Import a static license from an existing license file.
- ☐ **Upgrade from an existing Workbench installation**
Upgrade an existing license for an older version of the software. Your license must be covered by Maintenance, Upgrades and Support to use this option.
- ☐ **Configure license manager connection**
Configure a connection to a CLC Network License Manager that hosts network license(s) for this product, or update or disable an existing connection configuration.

If you experience any problems, please contact [QIAGEN Digital Insights Support](#)

Host-ID:

Proxy Settings Viewing Mode Previous **Next** Quit

Figure 1.23: Click on the *Viewing Mode* button at the bottom of the License Manager window to launch the Workbench in Viewing Mode.

To go from running in Viewing Mode to running a Workbench with its full functionality, it just needs to have access to a valid license. This can be done by installing a static license, or when using a network license, by restarting the Workbench when licenses are once again available.

1.4.9 Start in safe mode


If the program becomes unstable on start-up, you can start it in **Safe mode**. This is done by pressing and holding down the Shift button while the program starts.

When starting in safe mode, the user settings (e.g. the settings in the **Side Panel**) are deleted and cannot be restored. Your data stored in the **Navigation Area** is not deleted. When started in safe mode, some of the functionalities are missing, and you will have to restart the *CLC Genomics Workbench* again (without pressing Shift).

1.5 Plugins

The analysis functionality of the *CLC Genomics Workbench* can be extended substantially by installing plugins. The built-in Plugin Manager provides an up-to-date listing of the plugins available. These include commercial modules, such as those that are part of the QIAGEN CLC Genomics Workbench Premium product.

Alternatively, visit our plugin webpage for a list: <https://digitalinsights.qiagen.com/products-overview/plugins/>.

Plugins are installed and uninstalled using the **Plugin Manager**, which can be opened using the **Manage Plugins** () button in the **Toolbar**, or by going to the top level menu:

Utilities | Manage Plugins... ()

Note: To install plugins and modules using a centrally installed *CLC Workbench*, the software must be run in *administrator mode*. On Windows, right-click on the program shortcut and choose "Run as Administrator". On Linux, this usually means running the software with sudo privileges.

The Plugin Manager has two tabs at the top:

- **Manage Plugins** An overview of plugins that are installed.
- **Download Plugins** An overview of plugins available from QIAGEN that are not yet installed on your Workbench.

1.5.1 Install

To install a plugin, open the Plugin Manager and click on the **Download Plugins** tab. This will display an overview of the plugins available (figure 1.24).

Select a plugin in the list to display additional information about it in the right hand pane. Click on **Download and Install** to to install the plugin.

Accepting the license agreement

The End User License Agreement (EULA) must be read and accepted as part of the installation process. Please read the EULA text carefully, and if you agree to it, check the box next to the text **I accept these terms**. If further information is requested from you, please fill this in before clicking on the **Finish** button.

If you have a .cpa plugin installer file on your computer, for example if you have downloaded it from our website, install the plugin by clicking on the **Install from File** button at the bottom of the dialog and specifying the plugin *.cpa file.

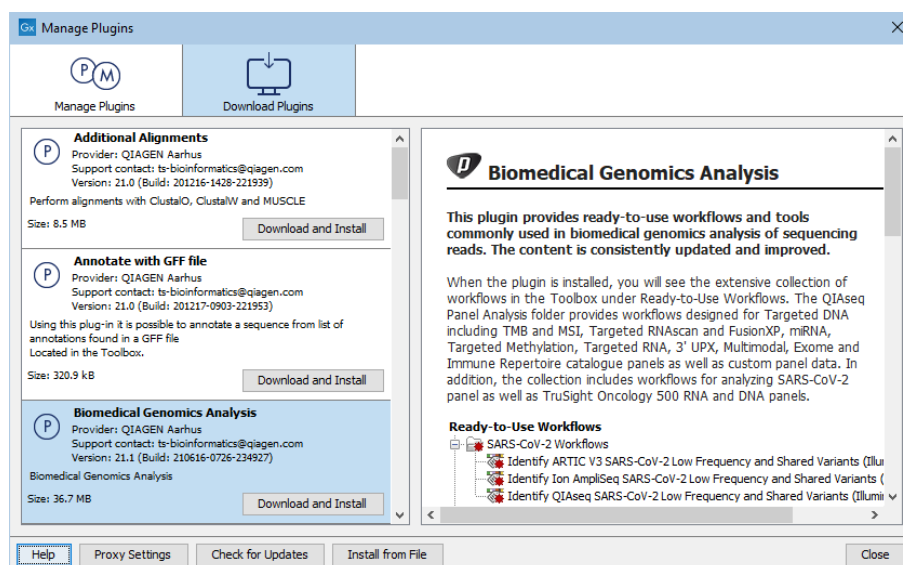


Figure 1.24: The plugins that are available for download.

When you close the Plugin Manager after making changes, you will be prompted to restart the software. Plugins will not be fully installed, or removed, until the *CLC Workbench* has been restarted.

1.5.2 Uninstall

Plugins are uninstalled using the Plugin Manager (figure 1.25). This can be opened using the **Manage Plugins (P)** button in the **Toolbar**, or by going to the top level menu:

Utilities | Manage Plugins... (P)

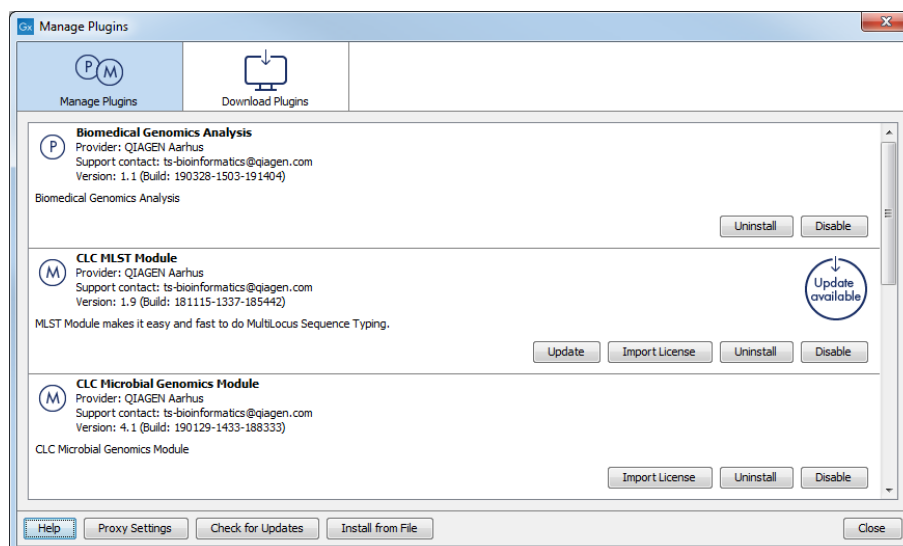


Figure 1.25: The plugin manager with plugins installed.

The installed plugins are shown in the **Manage plugins** tab of the plugin manager. To uninstall, select the plugin in the list and click **Uninstall**.

If you do not wish to completely uninstall the plugin, but you do not want it to be used next time

you start the Workbench, click the **Disable** button.

When you close the dialog, you will be asked whether you wish to restart the workbench. The plugin will not be uninstalled until the workbench is restarted.

1.5.3 Updating plugins

If a new version of a plugin is available, you will get a notification during start-up (figure 1.26).

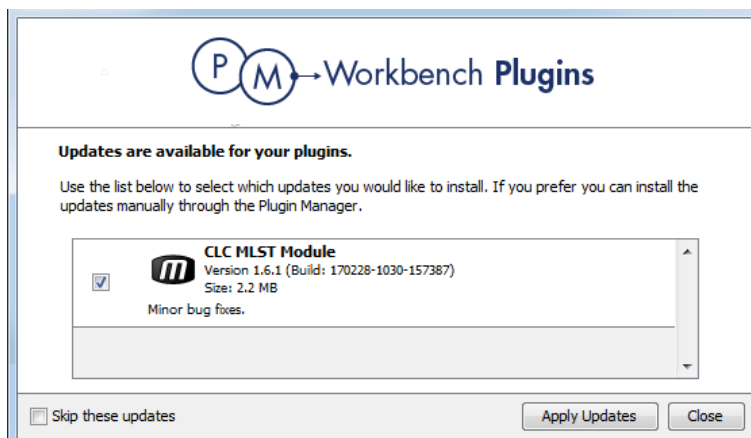


Figure 1.26: Plugin updates.

In this list, select which plugins you wish to update, and click **Install Updates**. If you press **Cancel** you will be able to install the plugins later by clicking **Check for Updates** in the Plugin manager (see figure 1.25).

1.6 Network configuration

If you use a proxy server to access the Internet you must configure *CLC Genomics Workbench* to use this. Otherwise you will not be able to perform any online activities.

CLC Genomics Workbench supports the use of an HTTP-proxy and an anonymous SOCKS-proxy.

To configure your proxy settings, go to **Edit | Preferences** and choose the **Advanced** tab (figure 1.27).

You have the choice between an HTTP-proxy and a SOCKS-proxy. The *CLC Workbench* only supports the use of a SOCKS-proxy that does not require authorization.

You can select whether the proxy should also be used for FTP and HTTPS connections.

List hosts that should be contacted directly, i.e. not via the proxy server, in the **Exclude hosts** field. The value can be a list, with each host separated by a `|` symbol. The wildcard character `*` can also be used. For example: `*.foo.com|localhost`.

The proxy can be bypassed when connecting to a *CLC Server* by checking the box next to **Bypass proxy when connecting to a CLC Server**.

Workbenches can be preconfigured to bypass the proxy settings when connecting to a *CLC Server* by configuring this setting in a `proxy.properties` file, where the IP address (not the host name) of the *CLC Server* is provided in the `proxyexclude` field. See <https://resources.qiagenbioinformatics.com>.

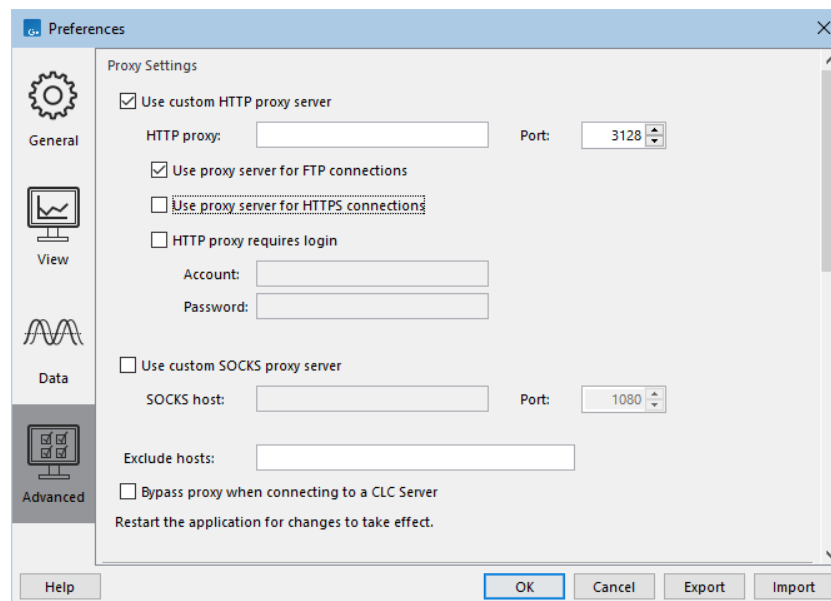


Figure 1.27: Advanced preference settings include proxy configuration options.

com/manuals/workbenchdeployment/current/index.php?manual=Per_computer_Workbench_information.html for further details.

If you have problems with these settings, please contact your systems administrator.

Part II

Core Functionalities

Chapter 2

User interface

Contents

2.1 View Area	45
2.1.1 Close views	48
2.1.2 Save changes in a view	48
2.1.3 Undo/Redo	49
2.1.4 Arrange views in View Area	50
2.1.5 Moving a view to a different screen	51
2.1.6 Side Panel	52
2.2 Zoom functionality in the View Area	55
2.3 Toolbox panel	56
2.4 Processes tab and Status bar	59
2.5 History and Element Info views	61
2.6 Workspace	63
2.7 List of shortcuts	64

The user interface of the *CLC Genomics Workbench* when it is first opened looks like that shown in figure 2.1.

Key areas are listed below with a brief description and links to further information.

- **Navigation Area** Data elements stored in File Locations are listed in the Navigation Area. (Section 3.1).
- **Toolbox Area** This area contains 4 tabs:
 - **Processes** Running and finished processes are listed under this tab. (Section 2.4)
 - **Tools** Analysis tools are available under this tab. (Section 2.3)
 - **Workflows** Template workflows and installed workflows are available under this tab. Apart from running workflows from here, you can right-click on them and choose to open a copy of the workflow. This opens a copy in the Workflow Editor (section 14.1).
 - **Favorites** Tools you use most often are listed here, and you can add tools you want, for quick access. (Section 2.3)

- **View Area** Data and workflows can be opened in this area for viewing and editing. (Section 2.1) A **Side Panel** with configuration options is present on the right hand side when items are open in this area (Section 4.6).
- **Menu bar** and **Toolbar** Many tools and associated actions can be launched using buttons and options in these areas.
- **Status Bar** The Workbench status and its connections to other systems is presented in this area. (Section 2.4).

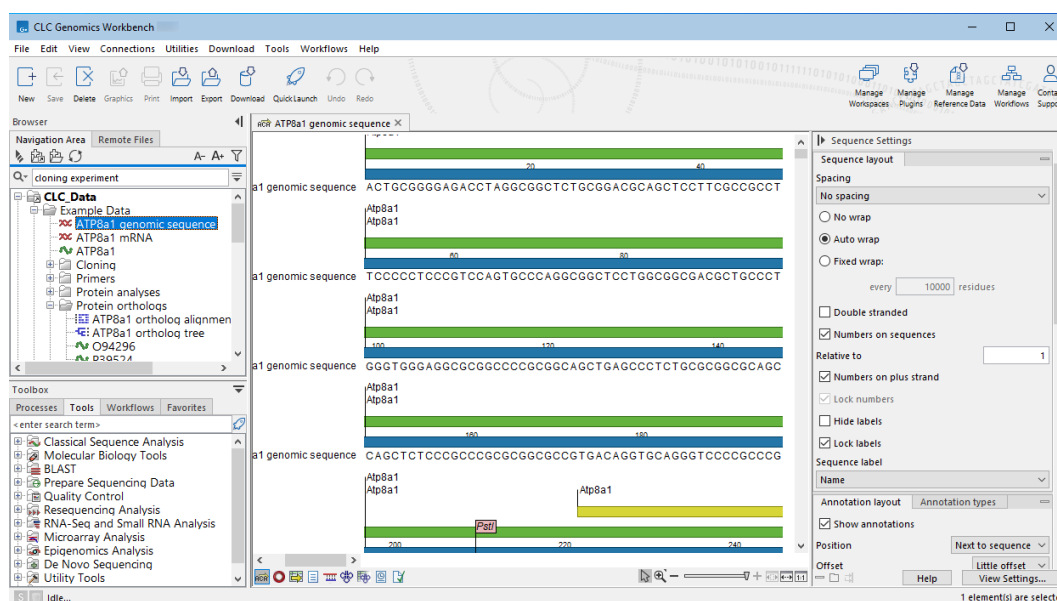


Figure 2.1: The **CLC Workbench** interface includes the **Navigation Area** in the top left, several tabs in the **Toolbox** area at the bottom left, a large viewing area on the right, menus and toolbars at the top, and a status bar at the bottom. A sequence has been opened in the **View** area, and a **Side Panel** containing settings relevant to viewing sequences is thus present on the far right.

Different areas of the interface can be hidden or made visible, as desired. Options controlling this are available under the **View** menu at the top. For example, whether the **Toolbox** panel should be visible, and which tabs should be visible in the **Toolbox** area can be configured using the menu options under:

View | Show/Hide Toolbox

You can also collapse the various areas by clicking on buttons like (▼) or (◀), where they appear. Similar buttons are presented for revealing areas if they are hidden.

2.1 View Area

The **View Area** is where elements are displayed when opened. Each open element is shown in a view, with a tab at the top containing the element's name (figure 2.2). Mouse-over a tab to reveal a tooltip containing summary information about the element.

Tabs can be dragged to put them in the order desired, and to move them to a different area in a split view (section 2.1.4).

Right-clicking on a tab opens a menu with various navigation options, as well as the ability to select tools and viewing options, etc.

Opening and viewing elements in the View Area

There are multiple ways to open an element in the View Area, including:

1. Double click on an element in the Navigation Area.
2. Right-click on an element in the Navigation Area, and choose the **Show** option from the context menu.
3. Drag elements from the Navigation Area into the Viewing Area.
4. Select an element in the Navigation Area and use the keyboard shortcut Ctrl + O (⌘ + O on macs)
5. Choose the option to "Open" results when launching an analysis. (This is only recommended when small numbers of elements will be generated, and where it is not important to save the results directly.)

When opening an element while another element is already open, the newly opened element will become the active tab. Click on any other tab to open it and make it the active view. Alternatively, use the keyboard shortcuts to navigate between tabs: Ctrl + PageUp or PageDown (or ⌘ + PageUp or PageDown on macs).

To provide more space for viewing data, you can hide Navigation Area and Toolbox by clicking the hide icon (⌵) at the top of the Navigation Area. You can also hide the Side Panel using the same icon at the top of the Side Panel.

Tooltips

For some data types and some views, tooltips provide additional useful information. Hover the mouse cursor over an area of interest to reveal these. For example, hover over an annotation on a sequence and a tooltip containing details about that annotation is shown. Hover over a variant in a variant track, and information about that variant is shown.

If you wish to hide such tooltips while moving the mouse around in a view, hold down the Ctrl key.

Tooltips can take a moment to appear. To make them show up immediately while moving the mouse around in a view, hold down the Shift key.

Accessing different views of the same data element

Various views of a data element are available. These can be opened by clicking on the icons at the bottom of an open view. The last two icons are always those for opening the **History** (🕒) and **Element Info** (📄) views. These are described in section 2.5.

For illustration, the icons for views available for sequence elements are shown in figure 2.3. Clicking on the **Show As Circular** (🔄) icon would present the sequence in a circular view. Mouse over any of these icons to see the type of view they represent.



Figure 2.2: Four elements are open in the View Area, organized in 2 areas horizontally - 3 elements in the top area, and one in the bottom. The active view is in the top area, as indicated by the blue bar just under the tabs.



Figure 2.3: The icons presented at the bottom of an open nucleotide sequence. Clicking on each of these presents a different view of the data.

Linked views Different views of the same element, or different elements referred to by another element, can be opened in a "linked view". This is particularly useful with multiple viewing areas open, i.e. split views. When views are linked, selecting an item or region in one view brings the relevant item or region into focus in the linked view(s). See figure 2.4, where a region was selected in one view, and that selection is then also shown in the other view.

To open a linked view, keep the Ctrl key (⌘ on macs) depressed and then click on the item to open. E.g. to open a different view of the same element, click on one of the icons at the bottom of the open view. The new view will open, often in a second, horizontal view area. When the View Area is already split horizontally, the new view is opened in the area not occupied by the original view.

Further information about split views is provided in section 2.1.4.

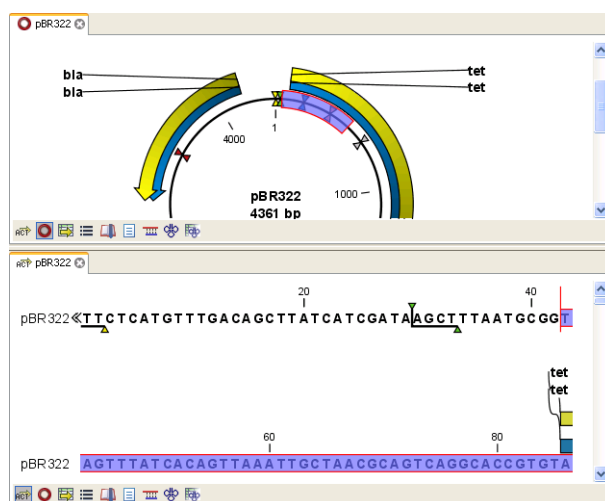


Figure 2.4: A split view of the same sequence element

2.1.1 Close views

To close a view, click on the X to the right of its name in the top tab. Alternatively, right-click on the element tab and choose from the following options (figure 2.5):

- **Close Tab** Close just the selected tab.
- **Close Tab Group** When tabs are open in a split view, all tabs in the same area as the selected tab will be closed. When not in split view, this option has the same effect as **Close All Tabs**.
- **Close All Other Tabs** Close all tabs in all tab areas except the selected tab.
- **Close All Tabs** Close all tabs in all tab areas.

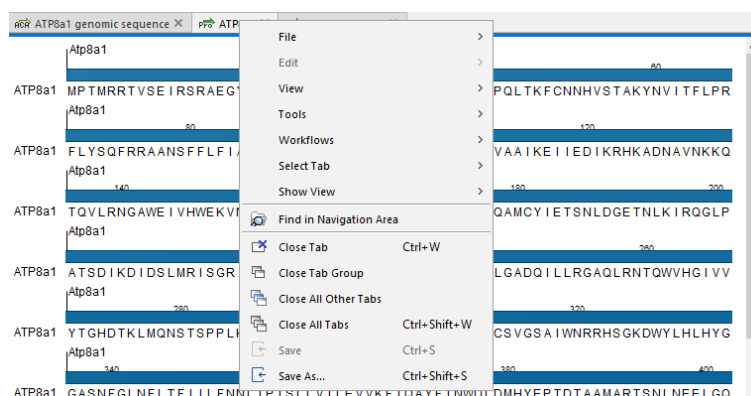


Figure 2.5: Right-click on the tab for a view, to see the options relating to closing open views.

2.1.2 Save changes in a view

When a new view is created, an * in front of the name of the view in the tab indicates that the element has not been saved yet. Similarly, when changes to an element are made in a view, an

* is added before the element name on the tab and the element name is shown in *bold and italic* in the Navigation Area (figure 2.6).



Figure 2.6: The ATP8a1 mRNA element has been edited, but the changes are not saved yet. This is indicated by an * on the tab name in the View Area, and by the use of bold, italic font for the element's name in the Navigation Area.

The **Save** function may be activated in two ways: Select the tab of the view you want to save and **Save** (⌘) or **Ctrl + S** (⌘ + S on Mac)

If you close a tab of a view containing an element that was edited, you will be asked if you want to save.

When saving an element from a new view that has not been opened from the Navigation Area, a save dialog appears (figure 2.7). In this dialog, you can name the element and select the folder in which you want to save the element.

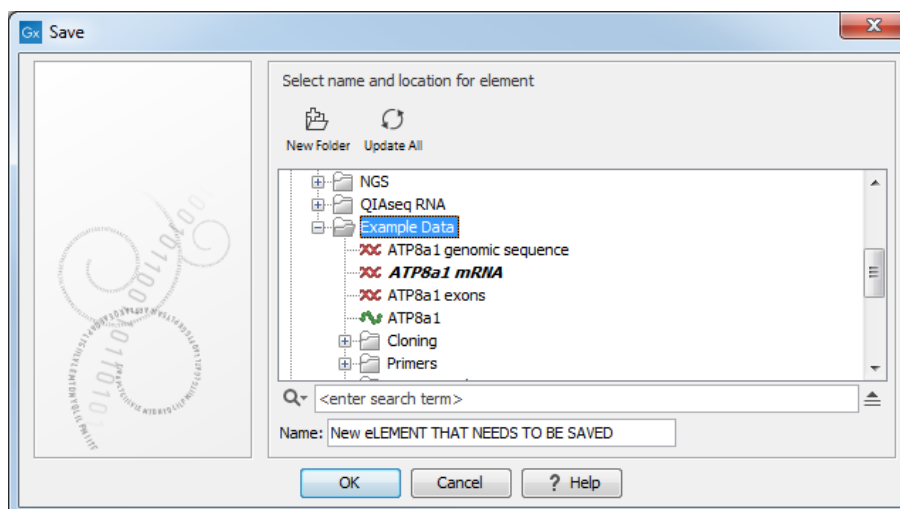


Figure 2.7: Save dialog. The new element has been name "New element that needs to be saved" and will be saved in the "Example Data" folder.

2.1.3 Undo/Redo

If you make a change to an element in a view, e.g. remove an annotation in a sequence or modify a tree, you can undo the action. In general, **Undo** applies to all changes you can make when right-clicking in a view. **Undo** is done by:

Click undo (↶) **in the Toolbar** or **Ctrl + Z**

If you want to undo several actions, just repeat the steps above.

To reverse the undo action:

Click the redo icon in the Toolbar or **Ctrl + Y**

Note! Actions in the Navigation Area, e.g., renaming and moving elements, cannot be undone. However, you can restore deleted elements (see section 3.1.8).

You can set the number of possible undo actions in the Preferences dialog, see section 4.

2.1.4 Arrange views in View Area

Views are arranged in the **View Area** by their tabs. The order of the views can be changed using drag and drop.

If a tab is dragged into a view, the area where the tab will be placed is highlighted blue. The blue area can be a tab bar in another view, or the bottom of an existing view. In that case, the tab will be moved to a new split view.

You can also split a View Area horizontally or vertically using the menus.

Splitting horizontally may be done this way:

right-click a tab of the view | View | Split Horizontally ()

This action opens the chosen view below the existing view. When the split is made vertically, the new view opens to the right of the existing view (see figure 2.8).

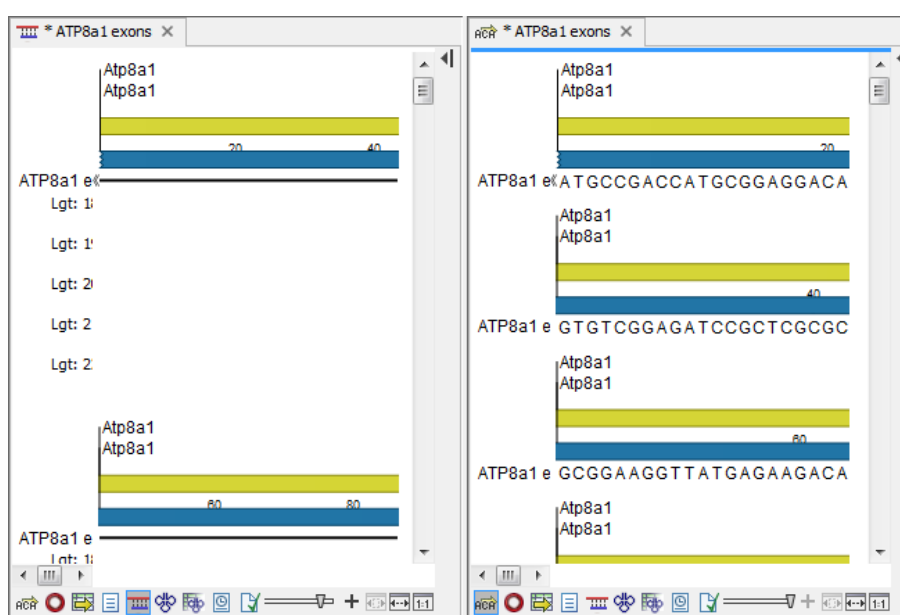


Figure 2.8: A vertical split screen.

A blue bar underneath the tab area indicates the active area, i.e. where the focus is at that time.

Splitting the View Area can be undone by dragging the tab of the bottom view to the tab of the top view, or by using the **Maximize/Restore View** function.

Select the view you want to maximize, and click

View | Maximize/restore View (□) or **Ctrl + M**

right-click the tab | View | Maximize/restore View (□)

or **double-click the tab of view**

The following restores the size of the view:

View | Maximize/restore View (□) or **Ctrl + M**

or **double-click title of view**

2.1.5 Moving a view to a different screen

Using multiple screens can be a great benefit when analyzing data with the *CLC Genomics Workbench*. You can move a view to another screen by dragging the tab of the view and dropping it outside the workbench window. Alternatively, you can right-click in the view area or on the tab itself and select **View | Move to New Window** from the context menu.

An example is shown in figure 2.9, where the main Workbench window shows a table of open reading frames, and the screen to the right is used to display the sequence and annotations.

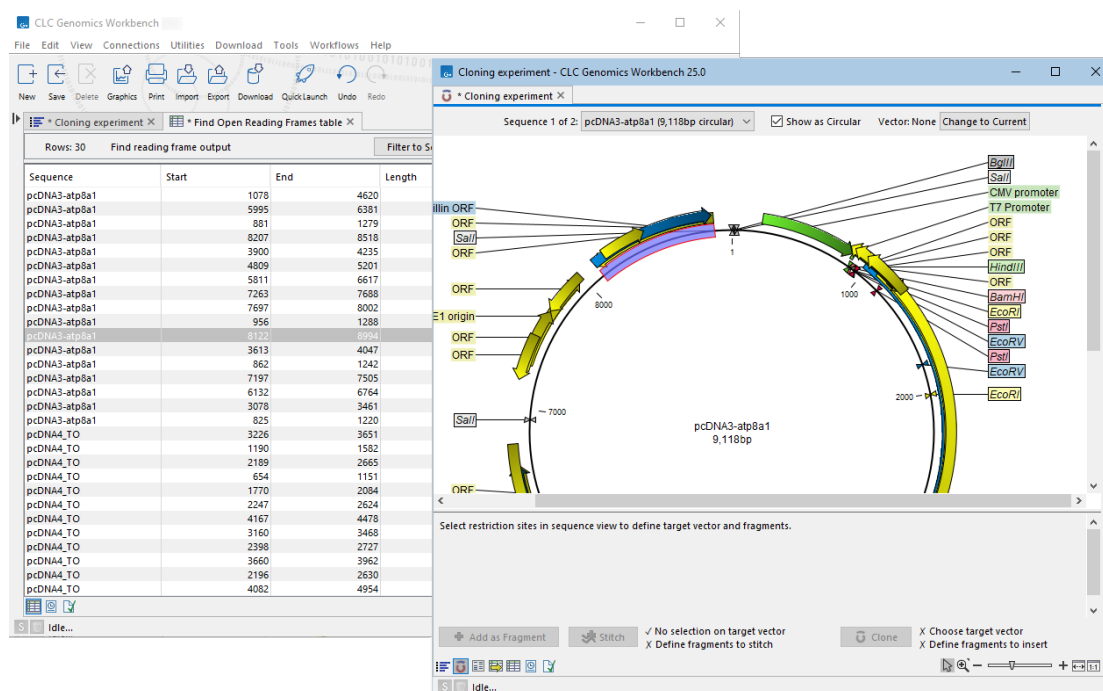


Figure 2.9: Showing the table on one screen while the sequence is displayed on another screen. Clicking the table of open reading frames causes the focus to shift to the corresponding region in the linked view, and that region to be selected.

You can make more detached windows, by dropping tabs outside the open workbench windows, or you can drag more tabs to a detached window. To get a tab back to the main workbench window, just drag the detached tab back, and drop it next to the other tabs in the top of the view area. **Note:** You should not drag the detached window header, just the tab itself.

2.1.6 Side Panel

When an element is open for viewing, options for configuring that view are available in the **Side Panel**. The settings for a given view can be saved and re-used, as described in section 4.6.

The **Side Panel** is organized into palettes, where related settings are grouped (figure 2.10). The specific palettes and options depend on the data type and the view that is open. The buttons in the top right side of each palette can be used for expanding the contents for viewing (\square), or collapsing ($=$). Using the buttons at the bottom left side of the Side Panel, all the palettes can be expanded (\square) or collapsed ($=$).

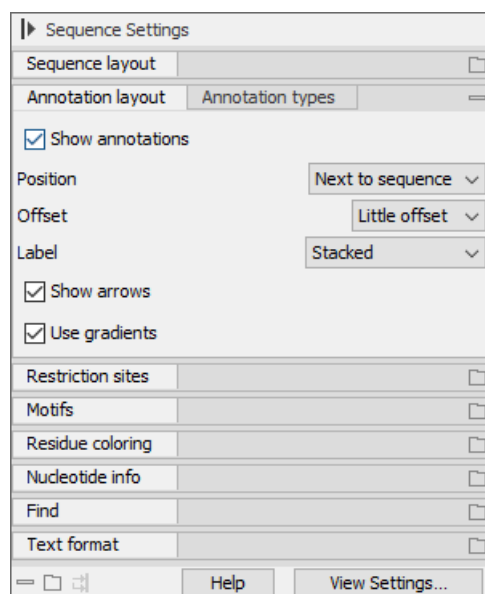


Figure 2.10: Side Panel settings for a nucleotide sequence. The Annotation layout palette is expanded, while the remaining palettes are collapsed. In the bottom left corner of the Side Panel are buttons for expanding, collapsing and re-docking all palettes.

Palettes can be placed at a different level in the Side Panel or undocked to be viewed anywhere on the screen (figure 2.11). For this, click on the palette name and while keeping the mouse depressed, drag the palette up and down, or drag it outside the CLC Workbench.

A palette can be re-docked by clicking on its tab name and dragging it back into the Side Panel. A red highlight indicates where it will be placed. Alternatively, clicking on the (\rightarrow) button at the top left of the floating palette will place it at the bottom of the Side Panel. All floating palettes can be re-docked by clicking on the (\rightarrow) button at the bottom of the Side Panel.

The whole Side Panel can be hidden or revealed using buttons at the top right: (\blacktriangleright) to hide the Side Panel and (\blacktriangleleft) to reveal it, if it was hidden. The keyboard shortcut Ctrl + U (⌘ + U on Mac) can also be used for these actions.

Changing the colors

Where color settings are offered, you have the option to use a fixed color or a gradient. For example, in figure 2.11, each annotation type is displayed using a fixed color, while the quality scores are displayed using a gradient. The color/gradient used can be changed by clicking on it.

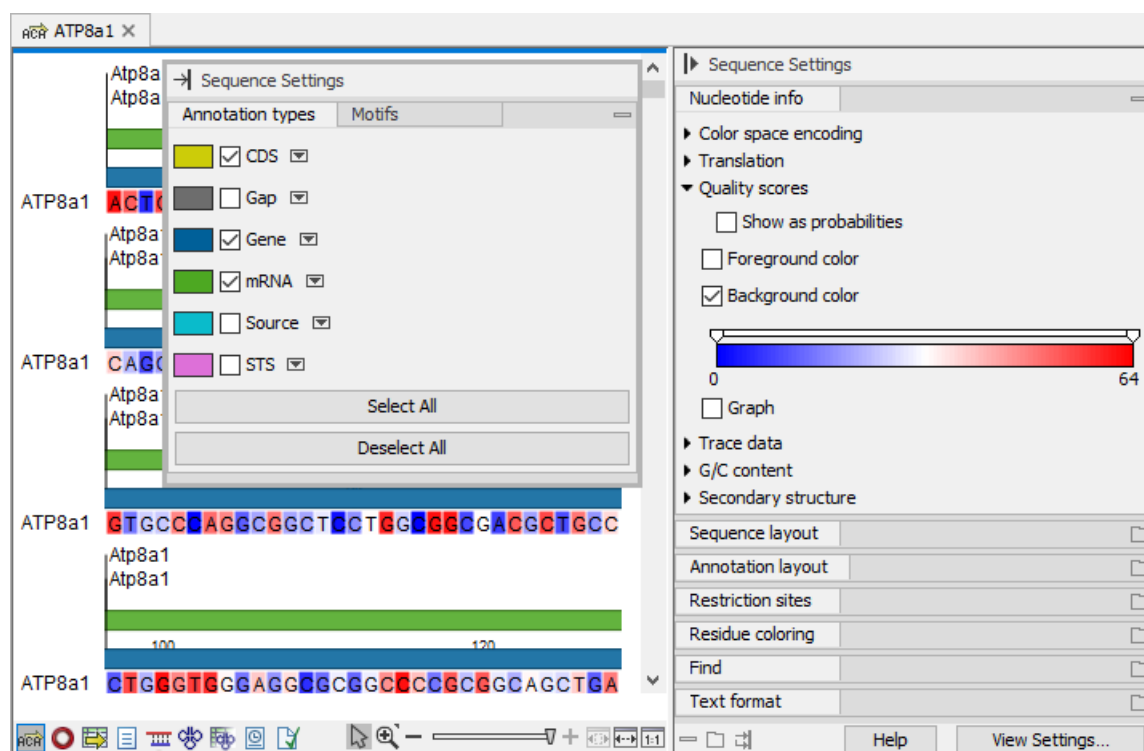


Figure 2.11: The Annotation types and Motifs palettes have been undocked. The Nucleotide info palette has been moved at the top of the Side Panel. The background color of nucleotides reflects the quality scores.

A fixed color can be chosen from a predefined set of swatches (figure 2.12) or defined by setting the:

- Hue, saturation, and value (HSV).
- Hue, saturation, and lightness (HSL).
- Red, green, and blue combination (RGB).
- Cyan, magenta, yellow, and black combination (CMYK).

A gradient can be chosen from a predefined set of gradients (figure 2.13) or customized by setting:

- The type of the gradient.
 - Continuous: the color gradually changes from one set color and location to the next.
 - Discrete: only the set colors are used and they change abruptly at the specified locations.
- Each color in the gradient and its location within the gradient.
 - The location can be provided as a percentage, ranging from 0% to 100%, or as an absolute value, with the minimum and maximum determined by the underlying data.

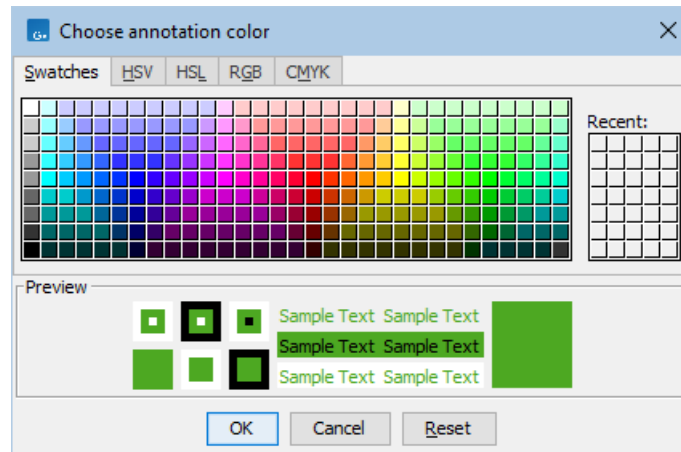





Figure 2.12: Clicking on the color of the mRNA annotation type opens a dialog where the color can be changed.

For example, the range of the gradient from figure 2.13 is 0 – 64. Setting the location to 50% corresponds to the absolute value of 32 ($0 + (64 - 0) * 0.5$).

Add locations by clicking on  or . Remove intermediate locations by clicking on .

- The color can be chosen as described above.

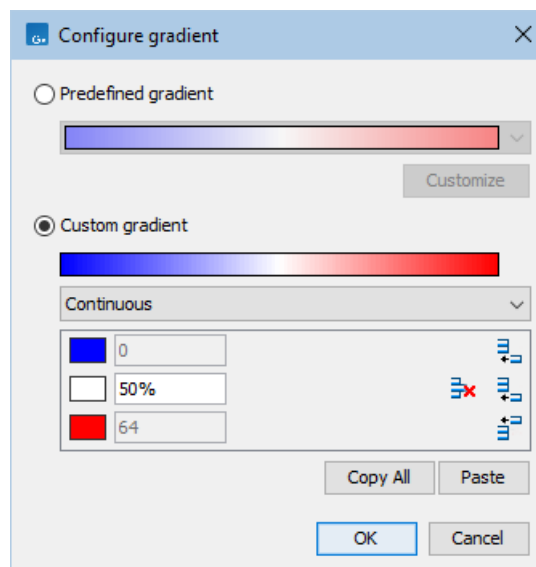


Figure 2.13: Clicking on the gradient of the quality scores opens a dialog where the gradient can be changed.

Gradients settings can be reused, making it easy to apply the same gradient consistently across different views. This is done using buttons in the 'Configure gradient' dialog (figure 2.13):

- Click on **Copy All** to copy the gradient configuration. You can paste this into a text file for later use.

- Click on the **Paste** button to apply copied gradient settings. Colors and locations present in the 'Configure gradient' dialog are overwritten by this action.

2.2 Zoom functionality in the View Area

All views except tabular and text views support zooming in and out. This section describes the most common zoom functionality. For protein 3D structures, see section 17.2.

Zoom tools are located at the bottom right corner of most views (figure 2.14).

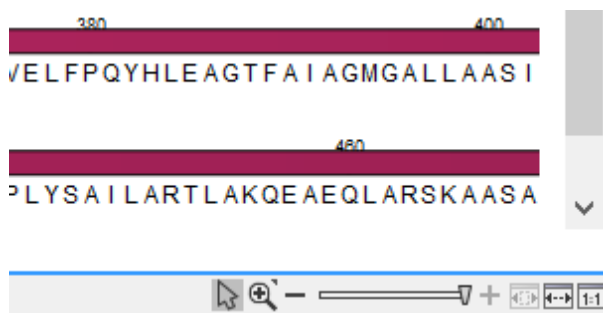


Figure 2.14: Zoom tools are located at the bottom right corner of the view.

The zoom tools include:

- Shortcuts for zooming out to fit the width of the view (↔) or zooming in all the way to see details (1:1).
- A shortcut to zoom to a selection (🔍). Select a region in the view, and then click this icon to zoom in on the selected region. (Keyboard shortcut Ctrl + 1)
- A slider to zoom in and zoom out to any desired level. The slider position reflects the current zoom level. Move the slider left to zoom out, or right to zoom in. For fine grained control, click on the slider and move the mouse up slightly or down slightly.
- Mouse mode buttons:
 - **Selection mode** (🖱️). Used when you wish to select data in a view. This is the default.
 - **Zoom in mode** (🔍) When selected, whenever you click the view, it zooms in. Alternatively, click on a location in the view, and the view will zoom in, with the focus on that location, or drag a box around an area, and the view will be zoomed to that area. (Keyboard shortcut Ctrl + 2)
 - If you press and hold on (🔍) or right-click on it, two other modes become available (figure 2.15).
 - **Panning** (🖱️) When selected, you can pan around in the the view around using the mouse. (Keyboard shortcut Ctrl + 4)
 - **Zoom out** (🔍) When selected, whenever you click the view, it zooms out. (Keyboard shortcut Ctrl + 3)

Additional notes:



Figure 2.15: Additional mouse modes can be found in the zoom tools when right-clicking on the magnifying glass.

- If you hold the mouse over the selection and zoom tools, tooltips will appear that provide further information about how to use the tools.
- If you press the **Shift** button on your keyboard while in zoom mode, the zoom function is reversed.
- You may have to click in the view before you can use the keyboard or the scroll wheel to zoom.

In many views, you can zoom in by pressing '+' on your keyboard, or zoom out by pressing '-' on your keyboard.

If you have a mouse with a scroll wheel, you can also do the following:

Zoom in: **Press and hold Ctrl (⌘ on Mac) | Move the scroll wheel on your mouse forward**

and

Zoom out: **Press and hold Ctrl (⌘ on Mac) | Move the scroll wheel on your mouse backwards**

2.3 Toolbox panel

The Toolbox panel at the bottom, left side of the Workbench, provides access to tools, installed workflows and template workflows, described in more detail below. The Processes tab, also present here, is described in section 2.4.

The font size in the Toolbox tabs can be increased or decreased using the (A-) or (A+) icons at the top, right hand side of the Navigation Area. Changing the font size is described in more detail in section 3.1.

Launching tools and workflows is described in section 12.1, with further details about launching workflows, individually or in batch mode, covered in chapter 14.

Tools tab

Tools available in the *CLC Workbench* are provided under the Tools menu, which is available in the Toolbox panel and as a menu at the top of the Workbench. The Tools menu is also available, in an extended form, in the Add Elements dialog available in the Workflow Editor (see section 14.1.1).

Tools are organized in folders according to their functionality. Tools provided by plugins may be in a plugin-specific folder (figure 2.16). When connected to a *CLC Genomics Server* with external applications configured and available, a folder for these will also be present in the Tools menu (figure 2.17).

You can search for tools of interest in the Tools tab in the Toolbox by entering a search term into the field at the top of the tab.

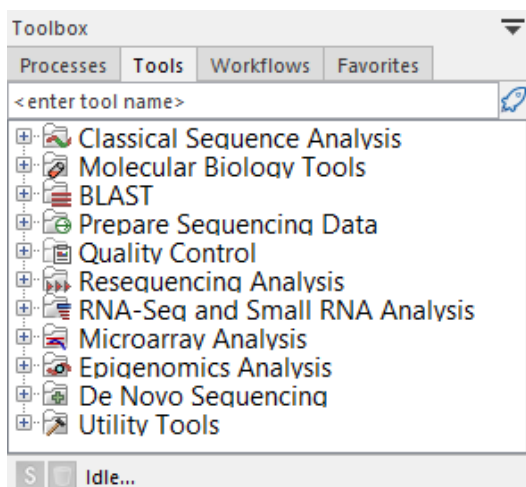


Figure 2.16: The Tools tab in the Toolbox contains folders of available tools. This Workbench is not connected to a CLC Server, as indicated by the grey server icon in the status bar.

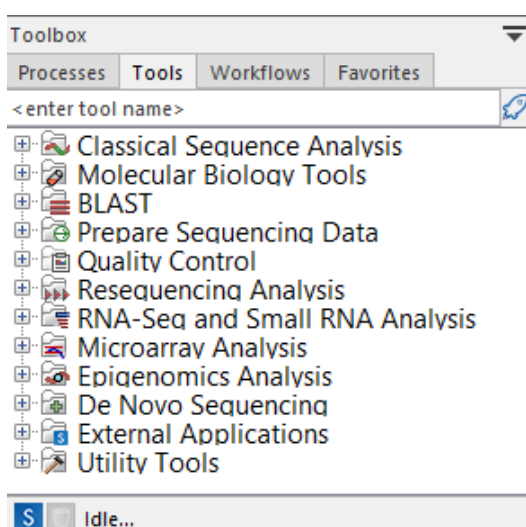


Figure 2.17: This Workbench is connected to a CLC Server, as indicated by the blue server icon in the status bar. External applications have been configured and enabled on that CLC Server, so an External Applications folder is listed, which contains those external applications. The server icon within that folder's icon is a reminder that these are only available when logged into the CLC Server.

Workflows tab

Workflows installed on the Workbench and template workflows (see section 14.5) are listed in the Workflows tab in the Toolbox (figure 2.18). The Workflows menu is also available at the top of the Workbench.

When connected to a CLC Server, workflows installed on that server will be available from a folder in the Workflows menu called **Installed Workflows (Server)** (figure 2.19).

You can search for workflows of interest in the Workflows tab in the Toolbox by entering a search term into the field at the top of the tab.

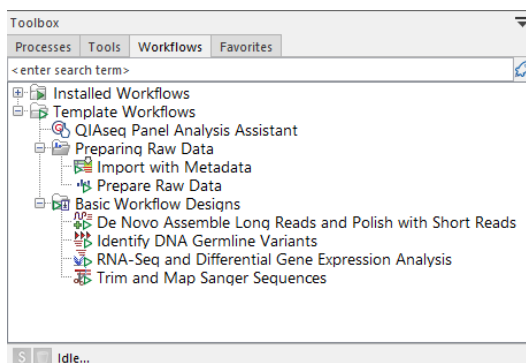


Figure 2.18: The Workflows tab in the Toolbox contains folders for workflows installed on the Workbench and template workflows. This Workbench is not connected to a CLC Server, as indicated by the grey server icon in the status bar. There are also no active AWS Connections, as indicated by the grey cloud icon in the status bar.

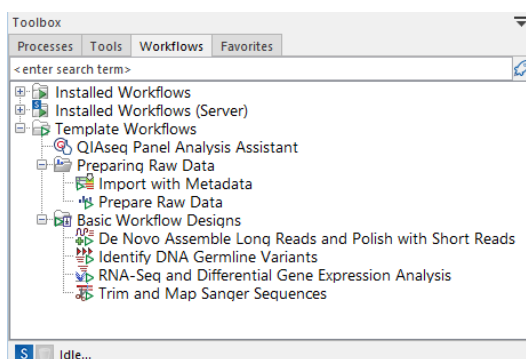


Figure 2.19: This Workbench is connected to a CLC Server, as indicated by the blue server icon in the status bar. This CLC Server has workflows installed on it, so a folder containing those workflows is present in the Workflows menu. (Installed Workflows (Server)).

Favorites tab

You can specify tools, or folders of tools, that you want to find quickly as favorites. In addition, the 10 tools you use most frequently are automatically identified as your frequently used tools or workflows. These lists are also made available in the Quick Launch tool (🔍), started using the Launch button in the toolbar, and in the Add Elements dialog available in the Workflow Editor (see section 14.1.1).

Manually adding tools to the Favorites list is done from tabs in the Toolbox panel:

- Right-click on a tool or folder of tools in the Tools tab and choose the option "Add to Favorites" from the menu that appears (figure 2.21), or
- Open the Favorites tab, right-click in the Favorites folder, choose the option "Add tools" or "Add group of tools". Then select the tool or tool group to add.

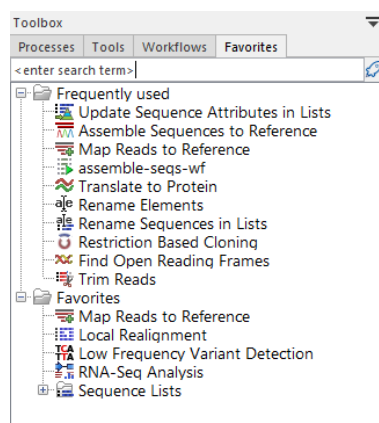


Figure 2.20: Under the Favorites tab is a folder containing your frequently used tools, which are added automatically, based on usage, and a folder containing tools you have specified as favorites.

- From the Favorites tab, click on an item in the Frequently used folder and drag it into the Favorites folder.

Items within the Favorites folder can be re-ordered by opening the Favorites tab in the Toolbox area in the bottom, left hand side of the Workbench and dragging tools up and down within the list. (Folders cannot be repositioned.)

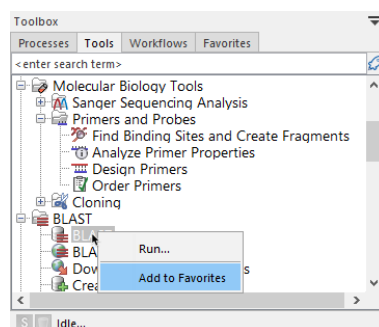


Figure 2.21: Tools or workflows can be added to the Favorites tab by right-clicking on them in the Toolbox area and choosing the "Add to Favorites" option.

To remove an item from the Favorites tab, right-click on it and choose the option **Remove from Favorites** from the menu that appears.

You can search for items of interest in the Favorites tab in the Toolbox by entering a search term into the field at the top of the tab.

2.4 Processes tab and Status bar

The Status bar is located at the bottom of the *CLC Genomics Workbench*. On the left side, information is displayed about whether a process is running or the Workbench is idle. Further to the left is information on the status of connections to other systems, such as a *CLC Server*. On the right hand side, context dependent information is displayed. For example, when selecting part of a sequence, the size of a selected region will be reported, or when mousing over a variant in a variant track, the location of the variant is reported.

Detailed information about running and completed processes for the Workbench session is provided under the Processes tab, found in the lower, left side of the Workbench. When logged into a CLC Server, the status of your jobs that are running, completed or queued on the server, are also displayed.

Several options are available after clicking on the small icon (🔍) next to a given process, as shown in figure 2.22).

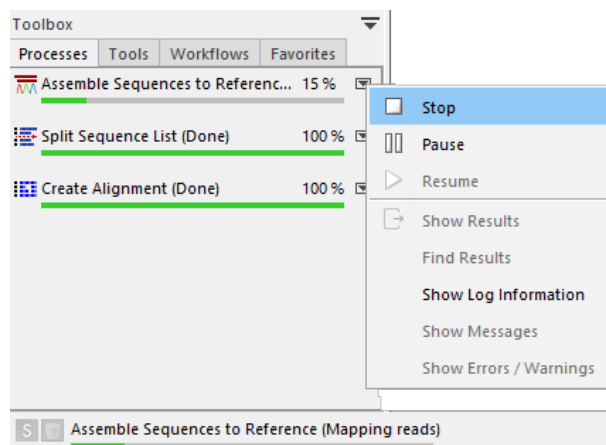


Figure 2.22: Completed jobs run during a Workbench and the progress of running jobs is visible in the Processes tab. The progress of a running job is also visible in the bottom frame of the Workbench. Clicking the small icon next to a process in the Process tab reveals a menu with actions that can be taken.

For completed jobs, these options provide a convenient way to locate results in the Navigation Area:

- **Show results** Open the results generated by that process in the Viewing Area. (Relevant if results were saved, as described in section 12.2.)
- **Find results** Highlight the results in the Navigation Area. (Relevant if results were saved, as described in section 12.2.)
- **Show Log Information** Opens a log of the progress of the process. This is the same log that opens if the option **Open Log** option is selected when launching a task.
- **Show Messages** Show any messages that were produced during the processing of your data.


Stopped, paused and finished processes are not automatically removed from the Processes tab during a Workbench session. They can, however, be removed by right clicking in the Processes tab and selecting the option "Remove Finished Processes" or by going to the option in the main menu system:


Utilities | Remove Finished Processes (X)


If you close the Workbench while jobs are still running on it, a dialog will ask for confirmation before closing. Workbench processes are stopped when the software is closed and these processes are not automatically restarted when you start the Workbench again. Closing the

Workbench does *not* interrupt jobs sent to a *CLC Server*, as described below.

Processes submitted to a CLC Server

Processes submitted to a *CLC Server* are listed in the Processes tab when the Workbench is logged into the server. Such processes have a server icon () to their left, rather than icons specific to the analysis being run. Processes that are queued or running on a *CLC Server* will reappear in the Workbench processes tab if you restart the Workbench (and log into the server). *CLC Server* processes already finished when you close the Workbench will not be shown again in the processes tab when you restart your Workbench.

Like running Workbench processes, processes running on a *CLC Server* can be stopped, by selecting clicking the small icon () next to the process and selecting the option "Stop". However, unlike jobs running on a Workbench, they cannot be paused or resumed.

Of note when running jobs on a *CLC Server*: If you choose the option "On my local disk or a place I have access to" when launching an import task, then the Workbench must maintain its connection to the *CLC Server* during the first part of the import process, data upload. If you try to close the Workbench during this phase, you will see a warning dialog. You can see what stage tasks are at in the **Processes** tab. Data upload from the Workbench to the server runs as a local, Workbench process. When the upload stage is complete, a new process for the import is started. This import process will have a server icon () to the left of it. At this point, you can disconnect or close your Workbench without affecting the import.

2.5 History and Element Info views

History view

The History view shows the log of all operations carried out on the element. This detailed record can be viewed within the *CLC Genomics Workbench*, as described here, or exported to a pdf format file.

To open the History view, click on the **Show History** () icon under the View area.

The table at the top of the History view contains a row for each operation that has affected this data element. When rows are selected in the table, full details for those operations are displayed in the bottom panel (figure 2.23).

Information in the table:

Summary information is shown in the table for each operation carried out in the creation of this data element:

- **Description** The operation performed
- **User** The username of the person who performed the operation. If you import data created by another person, that person's username will be shown.
- **Date and time** Date and time the operation was carried out. These are displayed according to your locale settings (see section 4.1).
- **Version** The software name and version used for that operation.

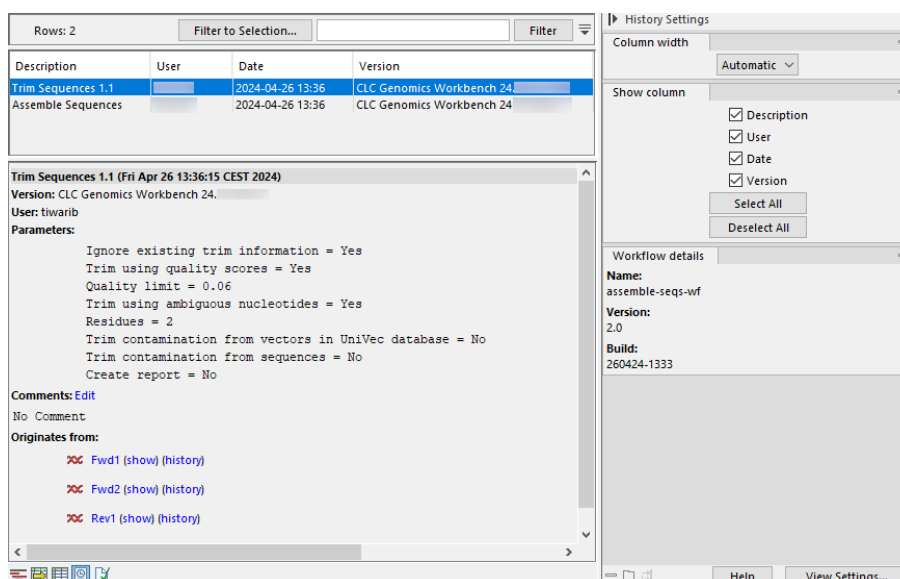


Figure 2.23: The history of an element created by an installed workflow called assemble-seqs-wf.

Information in the lower panel:

- **Parameters** The parameter values used for an analysis.
- **Comments** Additional details added here by tools or details that have been added manually. Click on **Edit** to add information to this field.
- **Originates from** A list of the elements that the current element originated from. Clicking on the name of an originating element selects it in the Navigation Area. Click on the "(show)" link to open the originating element to its default view. Click on "(history)" to open the originating element to its History view.

Information in the side panel:

- **Column width**
- **Show column**
- **Workflow details** Present if the element is an output from a workflow. The name and version of the workflow are listed here, and if the element was generated by an installed workflow (including template workflows), the workflow build id is also reported¹. If the element is output by a workflow launched from the Workflow Editor, the version is reported, but there will be no build id.

If an installer has never been made for a workflow, then data elements created using that workflow (launched from the Workflow Editor), will have 0.1 reported as the workflow version in their history. Workflows that have been used to make an installer inherit the most recent version assigned when creating the workflow installer. See section 14.6.2 for more on creating workflow installers.

¹Workflow build ids are included in the history of elements generated using version 24.0 or later.

Element Info view

To open the Element Info view, click on the **Show Element Info** (📄) icon under the View area.

The Element Info view contains information about the element, such as its name, description and other attributes. If the element is associated with metadata, that association is also reported here.

For further details about element information, please see section 15.4. For further information about metadata associations, see section 13.3.2.

2.6 Workspace

Workspaces are useful when working on more than one project. Open views and their arrangement are saved in a given workspace. Switching between workspaces can thus save much time when working on several different sets of data and results.

Initially, there is a single workspace called "Default". When you set up other workspaces, you assign each a name, which is used when re-opening that workspace, and which is displayed in the title bar of the Workbench when it is the active workspace.

The state of each workspace is saved automatically when the Workbench is closed down. The workspace that was open when closing down is the one that will be opened when the Workbench is started up again.

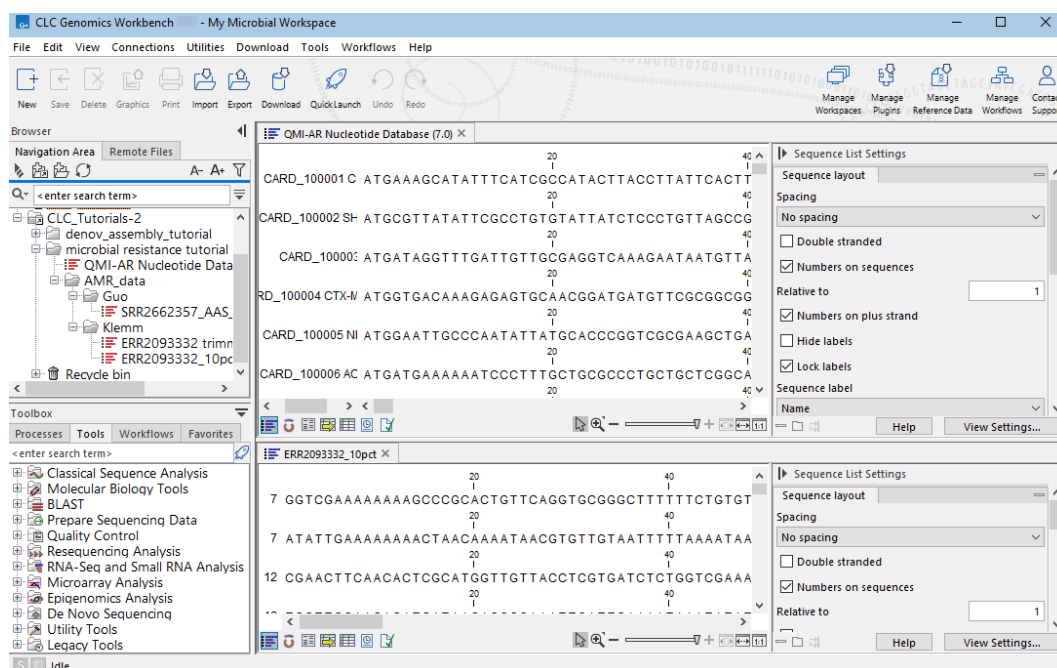


Figure 2.24: The workspace called "My Microbial Workspace" is open after selecting it from the menu opened by clicking on the Manage Workspaces button in the Toolbar. The name of the workspace is visible in the Workbench title bar.

Workspaces do not affect the underlying organization of data, so folders and elements remain the same in the Navigation Area.

Workspaces can be created, opened and deleted using the options available under the **Manage Workspaces** button in the top Toolbar, as described below. This functionality is also present under the View menu.

Creating a workspace

Create a new workspace by clicking in the **Manage Workspaces** button in the top Toolbar.

In the drop-down menu that appears, choose the option **Create Workspace**.

In the dialog that appears, enter a name for the new workspace.

When you click on the OK button, the new workspace is created and opened. The name of the workspace will be in the title bar of the Workbench.

Initially, the **Navigation Area** may be collapsed. Open it up again by clicking in the small black triangle at the top right of the Navigation Area.

Opening a workspace

Switch between workspaces by clicking in the **Manage Workspaces** button in the top Toolbar and selecting the desired workspace from the list presented.

The name of the active workspace will be greyed out in the list.

Deleting a workspace

To delete a workspace, click on the **Manage Workspaces** button in the top Toolbar and select the option **Delete Workspace**.

Workspaces that can be deleted are listed in a drop-down menu in the dialog that appears. Select the one to delete.

Deletion of workspaces cannot be undone.

Note: The Default workspace is not offered, as it cannot be deleted.

2.7 List of shortcuts

The keyboard shortcuts available in *CLC Genomics Workbench* are listed below.

General

Action	Windows/Linux	macOS
Adjust selection	Shift + arrow keys	Shift + arrow keys
Back to Navigation Area	Alt + Home or Alt + fn + left arrow	⌘ + Home or ⌘ + fn + left arrow
BLAST	Ctrl + Shift + L	⌘ + Shift + L
BLAST at NCBI	Ctrl + Shift + B	⌘ + Shift + B
Close	Ctrl + W	⌘ + W
Close all views	Ctrl + Shift + W	⌘ + Shift + W
Copy	Ctrl + C	⌘ + C
Create alignment	Ctrl + Shift + A	⌘ + Shift + A
Create track list	Ctrl + L	⌘ + L
Cut	Ctrl + X	⌘ + X
Delete	Delete	Delete or ⌘ + Backspace
Exit	Alt + F4	⌘ + Q
Export	Ctrl + E	⌘ + E
Export graphics	Ctrl + G	⌘ + G
Find Next Conflict	'.' (dot)	'.' (dot)
Find Previous Conflict	',' (comma)	',' (comma)
Help	F1	F1
Import	Ctrl + I	⌘ + I
Launch tools	Ctrl + Shift + T	⌘ + Shift + T
Maximize/restore View size	Ctrl + M	⌘ + M
Move gaps in alignment	Ctrl + arrow keys	⌘ + arrow keys
New Folder	Ctrl + Shift + N	⌘ + Shift + N
New Sequence	Ctrl + N	⌘ + N
Panning Mode	Ctrl + 4	⌘ + 4
Paste	Ctrl + V	⌘ + V
Print	Ctrl + P	⌘ + P
Redo	Ctrl + Y	⌘ + Y
Rename	F2	F2
Save	Ctrl + S	⌘ + S
Save As	Ctrl + Shift + S	⌘ + Shift + S
Scrolling horizontally	Shift + Scroll wheel	Shift + Scroll wheel
Search local data	Ctrl + Shift + F	⌘ + Shift + F
Search via Side Panel	Ctrl + F	⌘ + F
Search NCBI	Ctrl + B	⌘ + B
Search UniProt	Ctrl + Shift + U	⌘ + Shift + U
Select All	Ctrl + A	⌘ + A
Select Selection Mode	Ctrl + 1 (one)	⌘ + 1 (one)
Show folder content	Ctrl + O	⌘ + O
Show/hide Side Panel	Ctrl + U	⌘ + U
Show tooltip without delay	press and hold Shift	press and hold Shift
Sort folder	Ctrl + Shift + R	⌘ + Shift + R
Split Horizontally	Ctrl + T	⌘ + T
Split Vertically	Ctrl + J	⌘ + J
Suppress tooltip	press and hold Ctrl	press and hold Ctrl
Switch tabs in View Area	Ctrl + PageUp/PageDown or Ctrl + fn + arrow up/down	Ctrl + PageUp/PageDown or Ctrl + fn + arrow up/down
Switch views	Ctrl + Shift + PageUp Ctrl + Shift + PageDown	Ctrl + Shift + PageUp/arrow up Ctrl + Shift + PageDown/arrow down
Translate to Protein	Ctrl + Shift + P	⌘ + Shift + P
Undo	Ctrl + Z	⌘ + Z
Update folder	F5	F5
User Preferences	Ctrl + K	⌘ + ,

Scroll and Zoom shortcuts

Action	Windows/Linux	macOS
Vertical scroll in reads tracks	Alt + Scroll wheel	Alt + Scroll wheel
Vertical scroll in reads tracks, fast	Shift+Alt+Scroll wheel	Shift+Alt+Scroll wheel
Vertical zoom in graph tracks	Ctrl + Scroll wheel	⌘ + Scroll wheel
Zoom	Ctrl + Scroll wheel	⌘ + Scroll wheel
Zoom In Mode	Ctrl + 2	⌘ + 2
Zoom In (without clicking)	'+' (plus)	'+' (plus)
Zoom Out Mode	Ctrl + 3	⌘ + 3
Zoom Out (without clicking)	'-' (minus)	'-' (minus)
Zoom to base level	Ctrl + 0	⌘ + 0
Zoom to fit screen	Ctrl + 6	⌘ + 6
Zoom to selection	Ctrl + 5	⌘ + 5
Reverse zoom mode	press and hold Shift	press and hold Shift

Workflows related shortcuts

Action	Windows/Linux	macOS
Adjust workflow layout	Shift + Alt + L	⌘ + Shift + Alt + L
Workflow, add element	Alt + Shift + E	Alt + Shift + E
Workflow, collapse if its expanded	Alt + Shift + '-' (minus)	Alt + Shift + '-'
Workflow, create installer	Alt + Shift + I	Alt + Shift + I
Workflow, execute	Ctrl + enter	⌘ + enter
Workflow, expand if its collapsed	Alt + Shift + '+' (plus)	Alt + Shift + '+'
Workflow, highlight used elements	Alt + Shift + U	Alt + Shift + U
Workflow, remove all elements	Alt + Shift + R	Alt + Shift + R

Combinations of keys and mouse movements

Action	Windows/Linux	macOS	Mouse movement
Maximize View			Double-click the tab of the View
Restore View			Double-click the View title
Reverse zoom mode	Shift	Shift	Click in view
Select multiple elements not grouped together	Ctrl	⌘	Click elements
Select multiple elements grouped together	Shift	Shift	Click elements
Select Editor and highlight the corresponding element in the Navigation Area	Alt or Ctrl	⌘	Click tab

"Elements" in this context refers to elements and folders in the **Navigation Area** selections on sequences, and rows in tables.

Chapter 3

Data management and search

Contents

3.1	Navigation Area	68
3.1.1	Data structure	70
3.1.2	Adding and removing locations	70
3.1.3	Data sharing information	73
3.1.4	Create new folders	73
3.1.5	Multiselecting elements	73
3.1.6	Copying and moving elements and folders	74
3.1.7	Updating element and folder names	75
3.1.8	Delete, restore and remove elements	75
3.1.9	Show folder elements in a table	76
3.2	Working with non-CLC format files	77
3.3	Customized attributes on data locations	79
3.3.1	Filling in values	81
3.3.2	What happens when a clc object is copied to another data location?	82
3.3.3	Searching custom attributes	83
3.4	Searching for data in CLC Locations	83
3.4.1	Quick Search	84
3.4.2	Local Search	87
3.5	Backing up data from the CLC Workbench	88
3.6	Working with AWS S3 using the Remote Files tab	89

This chapter explains general data management features of *CLC Genomics Workbench*. The first section explains the basics of the data organization and the **Navigation Area**. The next section explains how to set up custom attributes for the data that can be used for more advanced data management. Finally, there is a section about how to search for data in your CLC Locations. The use of metadata tables in *CLC Genomics Workbench* is described separately, in chapter [13](#).

We recommend that data is only added and removed from CLC Data Locations using CLC software. If files are moved using other methods, the data may not be found when launching an analysis, and searches may not find that data. To address issues where data present in a CLC Data Location cannot be found, re-build the index for that location and try again. Information about rebuilding indexes can be found in section [3.4](#).

3.1 Navigation Area

The **Navigation Area** (figure 3.1) is used for organizing and navigating data.

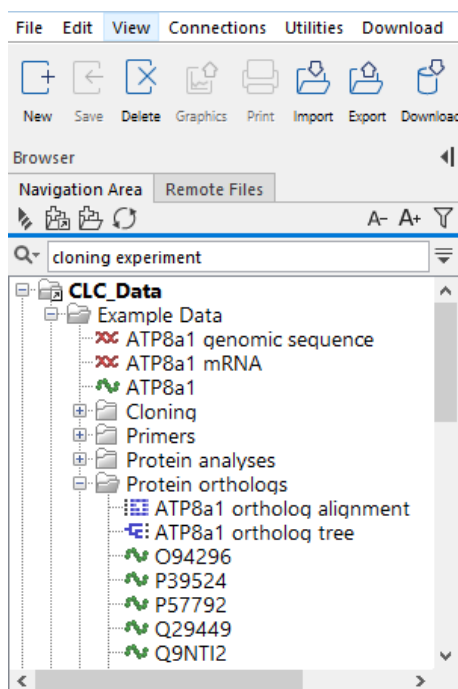


Figure 3.1: The Navigation Area.

Each CLC data element has a name and an icon that represents the type of data in the element. A list of many of the icons and the type of data they represent can be found at <https://qiagen.my.salesforce-sites.com/KnowledgeBase/KnowledgeNavigatorPage?id=kA41i000000L5uFCAS>.

Non-CLC files placed into CLC locations will have generic icons beside them, and any suffix in the original file name will be visible in the Navigation Area. (e.g. .pdf, .xml and so on.)


Elements placed in a folder (e.g. by copy/pasting or dragging) are put at the bottom of the folder listing. If the element is placed on another element, it is put just below that element in the folder listing. If an element of the same name already exists in that folder, a copy is created with the name extension "-1", "-2" etc. See section 3.1.6 for further details.

Elements in a folder can be sorted alphabetically by right-clicking on the folder and choosing the option **Sort Folder** from the menu that appears. When sorting this way on Windows, subfolders are placed at the top of the folder with elements listed below in alphabetical order. On Mac, subfolders and elements are listed together, in alphabetical order.

Opening and viewing CLC data elements is described in section 2.1.

Just above the data listing area is a Quick Search field, which can be used to find elements in your CLC Locations. See section 3.4.1.

Just above the Quick Search field are icons that can be clicked upon. On the left side, from left to right:

- **Collapse all** (). Close all the open folders in the Navigation Area.

- **Add File Location** (📁). Add a new top level location for storing CLC data. See section 3.1.2 for further details.
- **Create Folder** (📁) Create new folders within an existing CLC Location.
- **Update All** (🔄) Refresh the view of the Navigation Area.

On the right side, from left to right:

- **Decrease font size** (A-) and **increase font size** (A+) Decrease or increase the font size in the Navigation Area, both in the left hand side of the Workbench and other locations, such as launch wizards steps where data elements can be selected. The font size in the Tools, Workflows and Favorites tabs in the Toolbox, just below the Navigation Area, are also adjusted.
- **Restrict data types listed** (🔍) Click on this icon to reveal a list of data types. Selecting one of these types will result in only elements of that types, and folders, being shown in the Navigation Area. Click on it again to and select "All Elements" to see all elements listed once more.
- **Hide the Navigation Area and Toolbox** (⏮). This icon is at the top, right hand side. Clicking on it hides the **Navigation Area** and the **Toolbox** panels, providing more area for viewing data. To reveal the panels again, click on the (⏭) icon that is shown when the panels are hidden.

Data held on a CLC Server

If you have logged into a *CLC Server* from your Workbench, then data stored on the *CLC Server* will also be listed in the Navigation Area (figure 3.2).

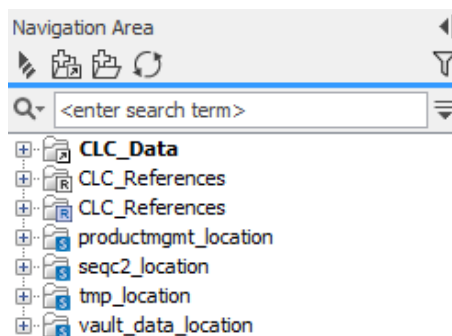


Figure 3.2: Data locations on a CLC Server are highlighted with blue square icons in the Navigation Area.

Locations on a *CLC Server* are configured by the server administrator.

When you launch a job to run on the *CLC Genomics Workbench*, you can choose data located in *CLC Genomics Workbench* File Locations or in *CLC Server* File Locations. When running jobs on a *CLC Server*, you can only select data in locations known to the server (see section 12.1.1).

3.1.1 Data structure

The data in the **Navigation Area** is organized into a number of **Locations**, also known as File Locations or Data Locations. A location represents a folder on the computer: The data shown under a Workbench location in the **Navigation Area** is stored on the computer, in the folder on the file system that the location points to. The full path to the system folder can be seen by mousing over the data location folder icon (figure 3.3).

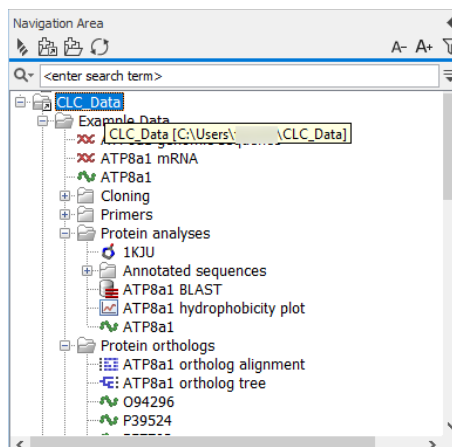


Figure 3.3: Mousing over the 'CLC_Data' location reveals a tooltip showing the full path to the folder on the underlying file system.

When the *CLC Genomics Workbench* is started for the first time, there will be a location called *CLC_Data*, which is the default data location.

Adding more locations and removing locations is described in section 3.1.2. Another location can be specified as the default by right-clicking on the location folder in the Navigation Area and choosing the option **Set as Default Location** from under Location in the menu (figure 3.4). This setting only applies to you. Other people using the same Workbench can set their own default locations.

Administrators can also change the default data location for all users of a Workbench. This is described at https://resources.qiagenbioinformatics.com/manuals/workbenchdeployment/current/index.php?manual=Default_Workbench_data_storage.html.

A location called *CLC_References* is also present by default. It is used for storing reference data downloaded using the Reference Data Manager and for other reference data that has been imported into this area. Further details about this are at section 11.

3.1.2 Adding and removing locations

When a Workbench is first installed a data location is configured by default. Unless your administrator has set things up otherwise, it will be named *CLC_Data*. It points to the following folder on the underlying system:

- Windows: C:\Users\<your_username>\CLC_Data
- Mac: ~/CLC_Data
- Linux: /homefolder/CLC_Data

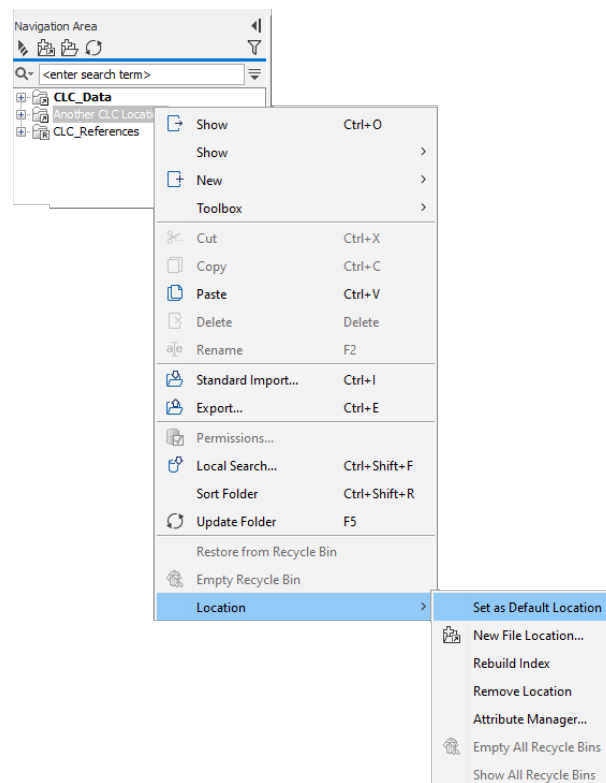




Figure 3.4: Data location options are available in a right-click context menu. Here, a new data location is being specified as the default.

Adding locations

To add a new location, click on the  icon at the top of the Navigation Area, or go to:

File | Location | New File Location 

Navigate to the folder to add as a CLC data location (see figure 3.5).

The name of the new location will be the name of the folder selected. To see the full path to the folder on the file system, hover the mouse cursor over the location icon .

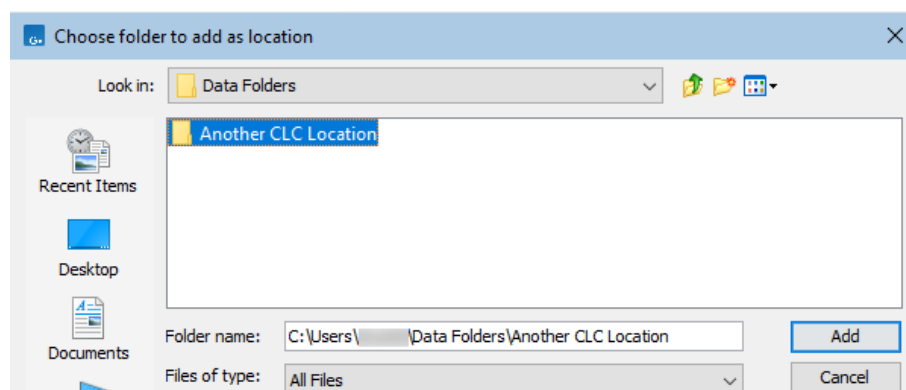


Figure 3.5: Navigating to a folder to use as a new CLC location.

The new location will appear in the **Navigation Area** (figure 3.6).

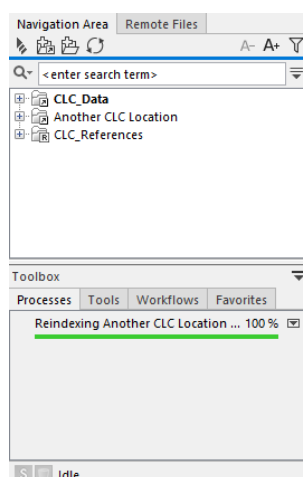


Figure 3.6: A new CLC location has been added. When the selected folder has not been used as a CLC location before, index files will be built, with the index building process listed in the Processes tab below the Navigation Area.

Additional notes about CLC locations

Requirements for folders being used as CLC locations:

- You must have permission to read from that folder, and if you plan to save new data elements or update data elements, you must have permission to write to that folder.
- The folder chosen as a CLC location must not be a subfolder of any area already being used as a CLC Workbench or CLC Server location.

Folders on a network drive or a removable drive can act as CLC locations. Please note though that interruptions to file access can lead to problems. For example, if you have set up a CLC location on One Drive, start editing a cloning experiment, and your laptop goes to sleep, unsaved work may be lost, and errors relating to the lost connection may be reported. If your CLC locations are on such systems, enabling offline line access (aka "always available files") can avoid such issues.

Locations appear inactive in the **Navigation Area** if the relevant drive is not available when you start up the Workbench. Once the drive is available, click on the Update All symbol (🔄) at the top of the Navigation area to refresh the view. All available locations will then be shown as active. There can be sometimes be a short delay before the interface update completes.

See section [3.1.3](#) for information relating to sharing CLC data locations.

Removing locations

To remove a CLC data location, right-click on the location (the top level folder), and select **Location | Remove Location**. The Location menu is also available under the top level **File** menu.

CLC data locations that have been removed can simply be re-added if you wish to access the data via the Workbench Navigation Area again.

After removing the CLC location, standard operating system functionality can be used to remove the folder and its contents from the local file system, if desired.

3.1.3 Data sharing information

The same underlying folder, for example on a network drive, can be added as a CLC data location in multiple Workbenches. This allows data to be easily shared. However, it is important to note that *data sharing is not actively supported*. Specifically:

- We do not support concurrent alteration of data. While the software will often detect this situation and handle it appropriately, by for example only allowing read access to all but the one party editing the file, we do not guarantee this.
- Any functionality that involves using the data search indices, (e.g. search functionality, associating metadata with data), will not work properly for shared data locations. Re-indexing a Data Location can help in the short term, but as soon as a new file is created by another piece of software, the index will be out of date.

If you decide to share data via Workbenches this way, it is vital that when adding a CLC location already used by other Workbenches as a CLC location, **the exact same folder in the file system hierarchy as the other Workbenches have used** is the one selected to add as a location. Indicating a folder higher up or lower down in the hierarchy will cause problems with the indexing of the files. This can lead to newly created objects made by Workbench A not being found when searching from Workbench B and vice versa, as well as issues with associations to CLC Metadata Tables.

Sharing the location of a folder or element with others

When working with others with the same CLC Location available via their *CLC Workbench*, or who connect to the same *CLC Server*, you can easily share the location of a particular element or folder, as described under *Searching using CLC URLs* in section 3.4.1.

3.1.4 Create new folders

Creating a new folder can be done in two ways:

Right-click an element in the Navigation Area | New | Folder 

or **File | New | Folder** 

If a folder is selected in the Navigation Area when adding a new folder, the new folder is added at the bottom of this folder. If an element is selected, the new folder is added right above that element.

You can move the folder manually by selecting it and dragging it to the desired destination.

3.1.5 Multiselecting elements

More than one element can be selected at the same time by:

- Holding down the <Ctrl> key (⌘ on Mac) while clicking on multiple elements selects the elements that have been clicked.
- Selecting one element, and selecting another element while holding down the <Shift> key selects all the elements listed between the two locations (the two end locations included).
- Selecting one element, and moving the cursor with the arrow-keys while holding down the <Shift> key, enables you to increase the number of elements selected.

3.1.6 Copying and moving elements and folders

To copy or move elements or folders in the Navigation Area, start by selecting the elements and/or folders. You can then use explicit actions for cutting, copying and pasting the selections, or you can use drag-and-drop to move them around.

Cut, copy and paste actions

After selecting elements and/or folders, **Cut** (✂) and **Copy** (📋) become available. After cutting or copying, the **Paste** (📄) action becomes available. There are several ways to access these actions:

- In the menu available if you right-click on a selected element or folder.
- In the top level **Edit** menu.
- As keyboard shortcuts:
 - Ctrl + C (⌘ + C on Mac) to copy
 - Ctrl + X (⌘ + X on Mac) to cut
 - Ctrl + V (⌘ + V on Mac) to paste

When you cut an element, it will appear "grayed out" until you activate the paste function. You can revert the cut command by copying another element.

Drag and drop for copying or moving

To use drag and drop to copy or move elements and/or folders, select the desired items, keep the mouse button depressed, and drag the selection to the new position.

When elements are moved *between different CLC Locations*, copies of the original elements are made and placed in the new location.

When elements are moved *within the same CLC Location*, no copy is created. The elements are just moved from one folder to another.

To make a copy of an element within the same CLC Location, use the **Copy** (📋) and **Paste** (📄) actions, or depress the Ctrl / Command key while dragging the data element to the new folder.

Copies of an element open in the View area can also be made by clicking on its tab in the View Area and dragging the tab to the desired location in the Navigation Area. This is *not* a way to save updates to an existing element. Any unsaved changes to the original element (the one open in the View area) remain unsaved until an explicit save action is taken on the original.

Names of copied or moved elements

CLC element names must be unique in a given CLC Location. This means that the same name can be used for different elements in different folders, but not for CLC elements in the same folder.

For non-CLC files available in the Navigation Area, the file must have a name unique within that CLC Location.

If an action results in a file with a name that clashes with an existing file, the name of the new element will have a numeric extension appended to guarantee uniqueness, e.g. "-1", "-2", etc.

3.1.7 Updating element and folder names

Renaming elements and folders

To start editing the name of an element or folder in the **Navigation Area**, do one of the following:

- Slow double-click on the item's name. I.e. Click on the name once, leave a short pause and click on the name again.
The speed of a slow double-click is usually defined at the system level. Double-clicking quickly on an element's name will open it in the viewing area, and double-clicking quickly on a folder name will open a closed folder or close an open folder.
- Click on the item's name to select it and then click on the function key F2.
- Click on the item's name to select it, and then select the option Rename from the top-level Edit menu.

When you have finished editing the name, click on the **Enter** key or select another element in the **Navigation Area**. To disregard changes before saving them, click on the **Esc** key.



If you update the name of an item you do not have permission to change, the new name will not be kept. The original name will be retained.

Renaming annotations is described in section [15.3.3](#).

3.1.8 Delete, restore and remove elements

When data in a *CLC Genomics Workbench* File Location is deleted, it is moved to the recycle bin within that File Location. From the recycle bin, data can be restored, i.e. taken back out of the recycle bin, or it can be deleted from the disk by emptying the recycle bin. Disk space is not freed up until data is deleted from the disk.

To delete an element, or a folder of elements:

1. Move it to the recycle bin by using the **Delete** () option from the **Edit** menu, the right-click menu of an element, or in the **Toolbar**, or use the **Delete key** on your keyboard.
2. Empty the recycle bin using the **Empty Recycle Bin** command available under the **Edit** menu or in the menu presented when you right-click on a **Recycle Bin** ()

Note! Emptying the recycle bin cannot be undone. Data is not recoverable after it has been removed by emptying the recycle bin.

For deleting annotations from sequences, see section 15.3.5.

To restore items in a recycle bin:

- Drag the items using your mouse into the folder where they used to be, or
- Right-click on the element and choose the option **Restore from Recycle Bin**.

Deleting data held on a CLC Server

Deleting data and folders works the same way for data in CLC Server File Locations as described above. The following differences should, however, be noted:

- The contents of your server-based recycle bin can be accessed by you and by your server administrator.
- CLC Server settings can affect how you work with server-based recycle bins. For example:
 - Recycle bins can be configured to be automatically emptied periodically.
 - The ability to empty server-based recycle bins can be restricted to something only administrators can do.

3.1.9 Show folder elements in a table

A location or a folder might contain large amounts of elements. It is possible to view their elements in the View Area:

select a folder or location | Show (📁) in the Toolbar

or **select a folder or location | right click on the folder and select Show (📁) | Contents**

An example is shown in figure 3.7.

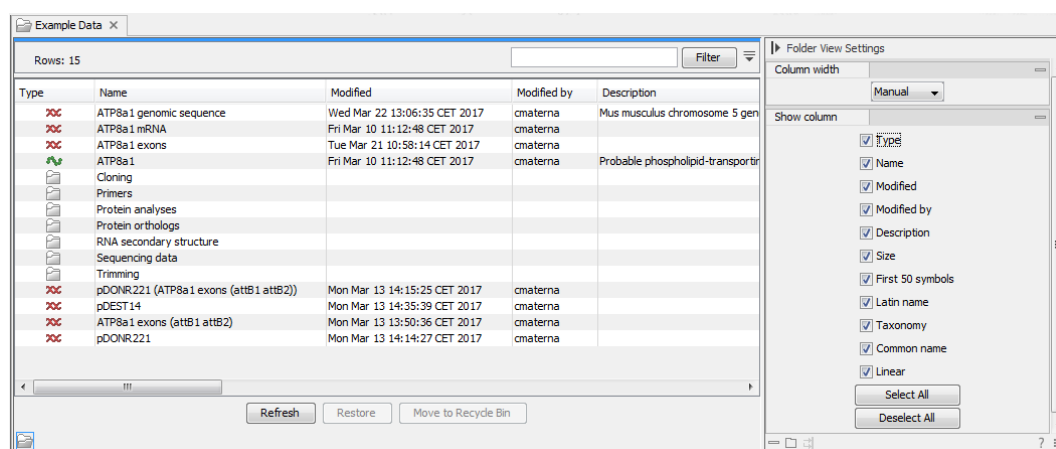


Figure 3.7: Viewing the elements in a folder.

When the elements are shown in the view, they can be sorted by clicking the heading of each of the columns. You can further refine the sorting by pressing Ctrl (⌘ on Mac) while clicking the heading of another column.

Sorting the elements in a view does not affect the ordering of the elements in the **Navigation Area**.

Note! The view only displays one "layer" at a time: the content of subfolders is not visible in this view. Also note that only sequences have the full span of information like organism etc.

Batch edit folder elements You can select a number of elements in the table, right-click and choose **Edit** to batch edit the elements. In this way, you can change for example the description or name of several elements in one go.

In figure 3.8 you can see an example where the name of two sequence are renamed in one go. In this example, a dialog with a text field will be shown, letting you enter a new name for these two sequences.

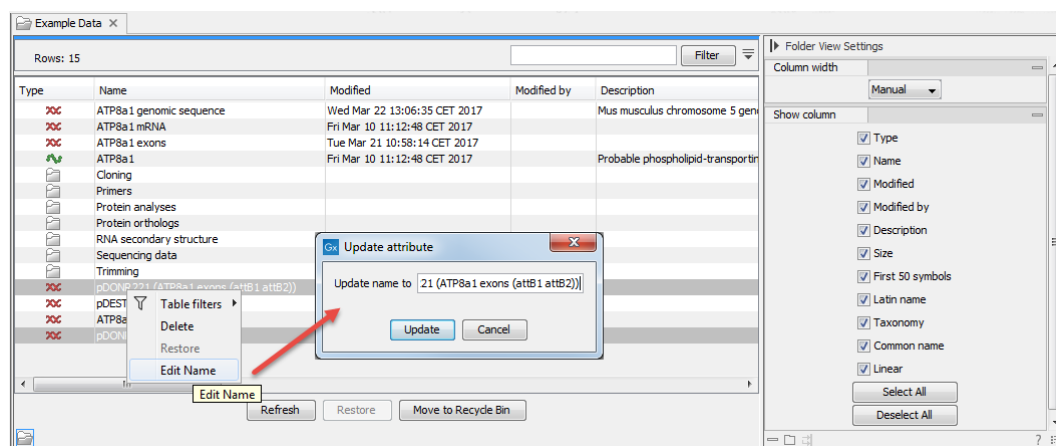


Figure 3.8: Changing the common name of two sequences.

Note! This information is directly saved and you cannot undo.

Drag and drop folder elements You can drag and drop objects from the folder editor to the Navigation area. This will create a copy of the objects at the selected destination. New elements can be included in the folder editor in the view area by dragging and dropping an element from a destination in the Navigation Area to the folder in the Navigation Area that you have open in the view area. It is not possible to drag elements directly from the Navigation Area to the folder editor in the View area.

3.2 Working with non-CLC format files

This section covers working with non-CLC format files stored in a CLC Location. Non-CLC format files can be recognized by their icons (e.g. representing a pdf or Word document), or by having a generic icon, when no specific icon could be assigned. Non-clc format files also usually have the original file extension in their names, such as .pdf, .docx, .xml and so on.

Non-CLC format files can be imported to a CLC Location using the Standard Import tool

(section 7.1). Any format not recognized is imported as an external (non-CLC format) file. Such files cannot be opened directly in the *CLC Workbench*, but double clicking on its name in the Navigation Area will open it in a relevant program, if your system supports this. Alternatively, if your system supports it, click on the file in the Navigation Area and, keeping the mouse button depressed, drag it onto a relevant program, for example one represented by an icon in a toolbar, to indicate the file should be opened by that tool.

Copying non-CLC format files to your file system

A copy of non-CLC format files in CLC Locations can be saved to other areas of the file system easily:

1. Select the non-CLC format files in the Navigation Area.
2. Right-click on a selected file and choose the option **Save to disk...** from the menu (figure 3.9).

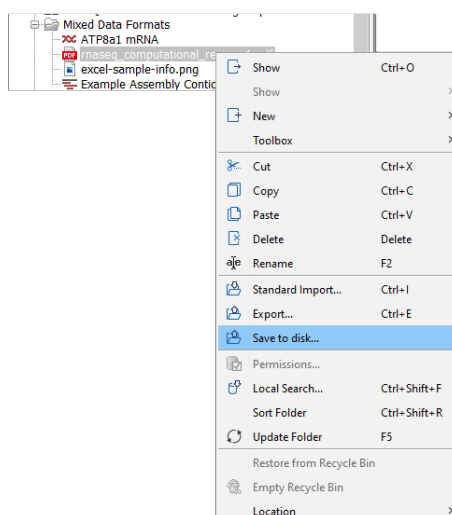


Figure 3.9: Select one or more non-CLC format files, right-click and choose the option "Save to disk..."

Using drag and drop for copying or moving non-CLC format files

Non-CLC format files can be saved to another place accessible on your system using drag and drop. To do this:

1. Select the non-CLC format file(s) in the Workbench Navigation Area.
2. Keeping the mouse button depressed, drag the selection to a local file browser.

Note: Dragging to a place on the same file system as the CLC Location results in the file(s) being *moved* from the CLC Location to the new location. Dragging to a location on a different file system results in the file(s) being *copied*, thus leaving the original file(s) in place in the CLC Location.

The "Save to disk..." functionality described in the section above always makes a copy.

3.3 Customized attributes on data locations

Location-specific attributes can be set on all elements stored in a given data location. Attributes could be things like company-specific information such as LIMS id, freezer position etc. Attributes are set using a CLC Workbench acting as a client to the CLC Server.

Note that the attributes scheme belongs to a particular data location, so if there are multiple data locations, each will have its own set of attributes.

To configure which fields that should be available¹ go to the Workbench:

right-click the data location | Location | Attribute Manager

This will display the dialog shown in figure 3.10.

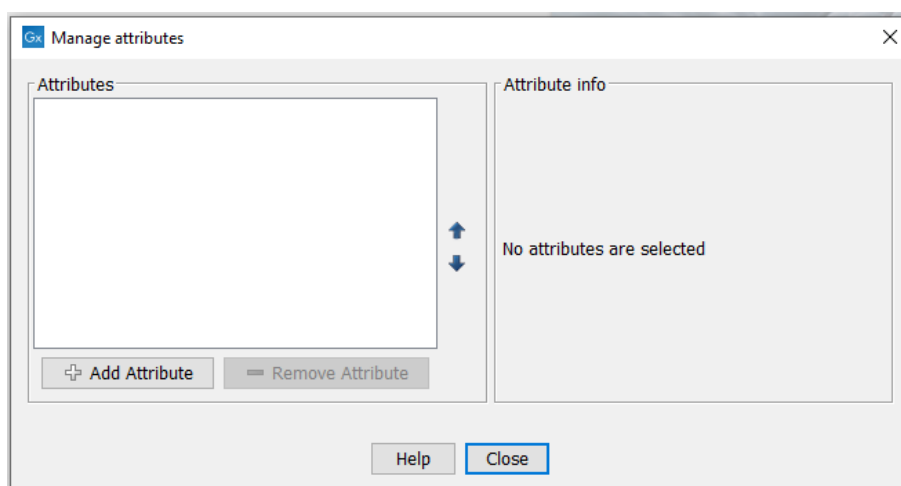


Figure 3.10: Adding attributes.

Click the **Add Attribute** (+) button to create a new attribute. This will display the dialog shown in figure 3.11.

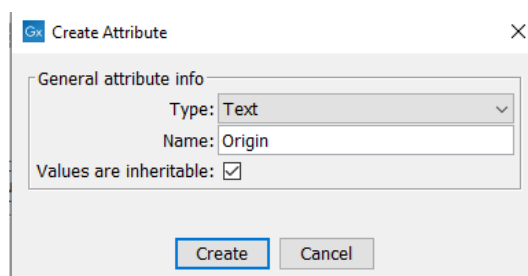


Figure 3.11: The list of attribute types.

First, select what kind of attribute you wish to create. This affects the type of information that can be entered by the end users, and it also affects the way the data can be searched. The following types are available:

- **Checkbox.** This is used for attributes that are binary (e.g. true/false, checked/unchecked and yes/no).
- **Text.** For simple text with no constraints on what can be entered.

¹If the data location is a server location, you need to be a server administrator to do this.

- **Hyper Link.** This can be used if the attribute is a reference to a web page. A value of this type will appear to the end user as a hyper link that can be clicked. Note that this attribute can only contain one hyper link. If you need more, you will have to create additional attributes.
- **List.** Lets you define a list of items that can be selected (explained in further detail below).
- **Number.** Any positive or negative integer.
- **Bounded number.** Same as number, but you can define the minimum and maximum values that should be accepted. If you designate some kind of ID to your sequences, you can use the bounded number to define that it should be at least 1 and max 99999 if that is the range of your IDs.
- **Decimal number.** Same as number, but it will also accept decimal numbers.
- **Bounded decimal number.** Same as bounded number, but it will also accept decimal numbers.

When a data element is copied, attribute values are transferred to the copy of the element by default. To prevent the values for an attribute from being copied, uncheck the **Values are inheritable** checkbox.

When you click **OK**, the attribute will appear in the list to the left. Clicking the attribute will allow you to see information on its type in the panel to the right.

Lists are a little special, since you have to define the items in the list. When you choose to add the list attribute in the left side of the dialog, you can define the items of the list in the panel to the right by clicking **Add Item** (+) (see figure 3.12).

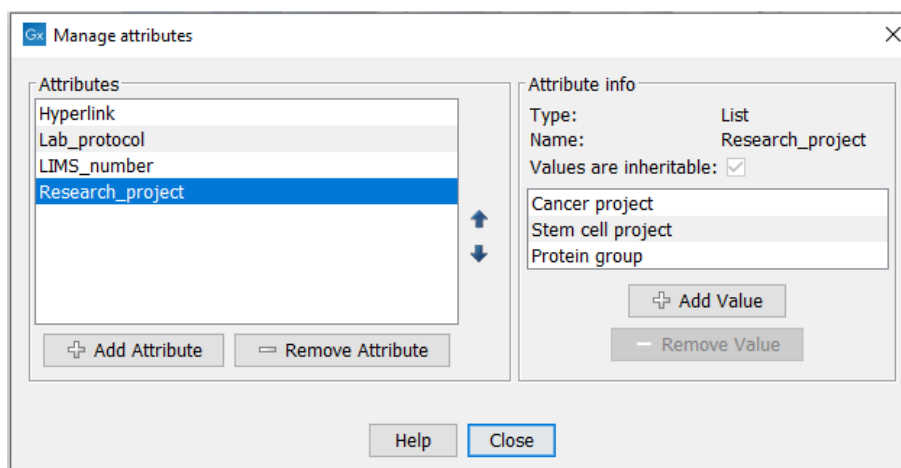


Figure 3.12: Defining items in a list.

Remove items in the list by pressing **Remove Item** (=).

Removing attributes To remove an attribute, select the attribute in the list and click **Remove Attribute** (=). This can be done without any further implications if the attribute has just been created, but if you remove an attribute where values have already been given for elements in the data location, it will have implications for these elements: The values will not be removed, but they will become static, which means that they cannot be edited anymore.

If you accidentally removed an attribute and wish to restore it, this can be done by creating a new attribute of exactly the same name and type as the one you removed. All the "static" values will now become editable again.

When you remove an attribute, it will no longer be possible to search for it, even if there is "static" information on elements in the data location.

Renaming and changing the type of an attribute is not possible - you will have to create a new one.

Changing the order of the attributes You can change the order of the attributes by selecting an attribute and click the **Up** and **Down** arrows in the dialog. This will affect the way the attributes are presented for the user.

3.3.1 Filling in values

When a set of attributes has been created (as shown in figure 3.13), the end users can start filling in information.

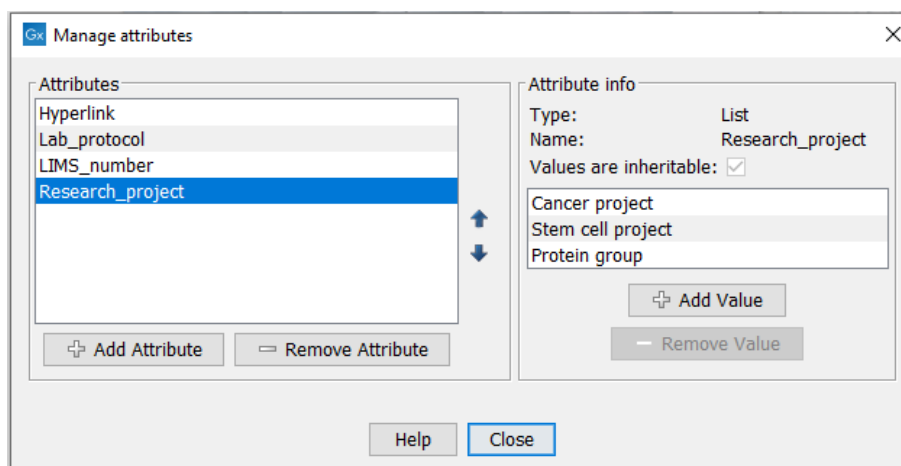


Figure 3.13: A set of attributes defined in the attribute manager.

This is done in the element info view:

right-click a sequence or another element in the Navigation Area | Show (🔍) | Element info (📄)

This will open a view similar to the one shown in figure 3.14.

You can now enter the appropriate information and **Save**. When you have saved the information, you will be able to search for it (see below).

Note that the element (e.g. sequence) needs to be saved in the data location before you can edit the attribute values.

When nobody has entered information, the attribute will have a "Not set" written in red next to the attribute (see figure 3.15).

This is particularly useful for attribute types like checkboxes and lists where you cannot tell, from the displayed value, if it has been set or not. Note that when an attribute has not been set, you cannot search for it, even if it looks like it has a value. In figure 3.15, you will *not* be able to find

1WCL_A x

Fixed Fields

- ▼ Name Edit
1WCL_A
- ▼ Description Edit
- ▼ Metadata Refresh
This element is not associated with any metadata
- Local Attribute Fields
- ▼ Research_project Clear
Cancer project
- ▼ Hyper_link Edit Clear
<http://www.clcbio.com>
- ▼ Is_confirmed Clear
☒
- ▼ Lab_protocol Edit Clear
P123
- ▼ LIMS_number Clear
412
- ▼ Location Clear
Lab 23
- ▼ Patient_number Clear
651

Figure 3.14: Adding values to the attributes.

Local Attribute Fields

- ▼ Research_project Clear
Cancer project
- ▼ Hyper_link Edit Clear
Not set

Figure 3.15: An attribute which has not been set.

this sequence if you search for research projects with the value "Cancer project", because it has not been set. To set it, simply click in the list and you will see the red "Not set" disappear.

If you wish to reset the information that has been entered for an attribute, press "Clear" (written in blue next to the attribute). This will return it to the "Not set" state.

The **Folder editor**, invoked by pressing **Show** on a given folder from the context menu, provides a quick way of changing the attributes of many elements in one go. For details, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Show_folder_elements_in_table.html.

3.3.2 What happens when a clc object is copied to another data location?

The user supplied information, which has been entered in the **Element info**, is attached to the attributes that have been defined in this particular data location. If you copy the sequence to another data location or to a data location containing another attribute set, the information will become fixed, meaning that it is no longer editable and cannot be searched for. Note that attributes that were "Not set" will disappear when you copy data to another location.

If the element (e.g. sequence) is moved back to the original data location, the information will again be editable and searchable.

If the e.g. Molecule Project or Molecule Table is moved back to the original data location, the information will again be editable and searchable.


3.3.3 Searching custom attributes

When a text-based attribute is created, it becomes available for searching using Quick Search or Local Search.

Quick Search: Precede the name of the attribute with "attrib_" and then add a colon followed by the query term, e.g. "attrib_color:blue".

You must use whole words when searching for elements containing particular values for local attributes.

For searching purposes, **words** are the terms on either side of a space, hyphen or underscore in a name. The names of elements and folders are split into words when indexing.

In the **Local Search** () tool, the local attributes are available to select from a drop down list, in addition to Name and Path.

Read more about search here: https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Local_Search.html.

3.4 Searching for data in CLC Locations

There are two ways of searching for data in your CLC Locations:

- **Quick Search** Available above the **Navigation Area** and described in section 3.4.1. By default, terms entered are used to search for the names of data elements and folders across all available CLC Locations.
- **Local Search** Available under the Utilities menu and described in section 3.4.2. All CLC Locations can be searched, or an individual Location can be searched. Local searches can be saved, so that the same search can be run again easily.

Re-indexing CLC Locations

The *CLC Genomics Workbench* automatically indexes data added to a CLC Data Location using CLC software tools. These indexes are used when searching in the Navigation Area, to fetch the relevant data elements when launching an analysis and for resolving associations between data elements and CLC Metadata Tables. When problems with any of these are experienced, it may be resolved by re-indexing that CLC Location.

To re-index, right-click on the relevant CLC Location in the Navigation Area and then, from the menu that appears, select the option:

Location | Rebuild Index

Rebuilding the index for locations with a lot of data can take some time.

The index building process can be stopped under the Processes tab, described in section 2.4.

Indexes are updated automatically when data is moved between CLC Locations, but not when data elements are moved within a given CLC Location. When searching based on names, this does not matter. However, for searches based on information in the path, the index may need to be rebuilt before searching.

3.4.1 Quick Search

The search field just above the Navigation Area on the left side of the *CLC Genomics Workbench* (figure 3.16) can be used to search for data elements and folders in all available CLC Locations. By default matches against names are reported. Searches are case insensitive.

A CLC URL, which specifies a specific element or folder, can also be entered. This is described at the end of this section.

Using Quick Search for searches based on names

The examples below refer to searches where the following elements are present in a CLC Location.

- E. coli reference sequence
- Broccoli sequence
- Coliform set

Search with a single term to look for any element or folder with a name containing that term.

Example 1: A search for *coli* would return all 3 elements listed above.

Search with two or more terms to look for any element or folder with a name containing all of those terms.

Example 2: A search for *coli set* would return "Coliform set" but not the other two entries listed in the earlier example.

Search with two or more words in quotes to look for any element or folder name containing those words, appearing consecutively, in the order provided. Whole words must be used within quotes, rather than partial terms.

For searching purposes, **words** are the terms on either side of a space, hyphen or underscore in a name. The names of elements and folders are split into words when indexing.

Example 3: A search for "*coli reference*" would find an element called "E. coli reference sequence".

Example 4: A search for "*coli sequence*" would not return any of the elements in the example list. In the name "E. coli reference sequence", the words *coli* and *sequence* are not placed consecutively, and in "Broccoli sequence", "*coli*" is a partial term rather than a whole word.

Why only words when searching with quotes? The use of quotes allows quite specific searches to be run quickly, but only using words, as defined by the indexing system.

Tip: Searches with whole words are faster than searching with partial terms. If a term is a word in some names but a partial term in others, the hits found using the complete word are returned first. E.g. searches with the term *cancer* would return elements with names like "cancer reads" and "my cancer sample" before an element with a names like "cancerreads".

Note: Wildcards (* ? ~) are ignored in basic searches. If you wish to define a search using wildcards, use the advanced search functionality of Quick Search.

Advanced searching using Quick Search

When a search term is preceded by "name:", more specific searches can be defined, using standard wild cards. If multiple query terms are provided, elements and folders containing all terms are returned. Searches are case insensitive.

Terms in the path can also be searched for by preceding the term with "path:".

If you have defined local (custom) attributes for any of your CLC Locations, the contents of these can also be searched with using the name of the attribute preceded by "attrib_", e.g. "attrib_color:blue".

Example 5: A search for **path:tutorials name:cancer** would find all data elements or folders where at least one folder in its path contained "tutorials" and the name of the folder or element included the word "cancer". Word here is as defined earlier in this section. In this example, an element named "cancer tissue reads" in a subfolder under "CLC_Tutorials" would be found, but an element called "cancerreads" in that subfolder would not be.

Example 6: A search for **path:tutorials cancer** would find all data elements or folders where at least one folder in its path contained "tutorials" (with capital or small letters) and the name of the folder or element reported *including* the word "cancer". In this example, an element named "cancer tissue reads" in a subfolder under "CLC_Tutorials" would be found, and so would an element called "cancerreads" in that subfolder.

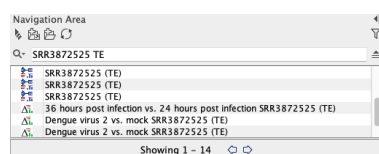


Figure 3.16: Enter terms in the Quick Search field to look for elements and folders.

Advanced search expressions

The terms below can only be used when the field being searched has been specified explicitly, e.g. by preceding the search term with "name:" or "path:".

- **Wildcard multiple character search (*).** Appending an asterisk * to the search term find matches that start with that term. E.g. A search for **BRCA*** will find terms like **BRCA1**, **BRCA2**, and **BRCA1gene**.
- **Wildcard single character search (?).** The ? character represents exactly one character. For example, searching for **BRCA?** would find **BRCA1** and **BRCA2**, but would not find **BRCA1gene**.
- **Search related words (~).** Appending a tilde to the search term looks for fuzzy matches, that is, terms that almost match the search term, but are not necessarily exact matches. For example, **:ADRAA~** will find terms similar to **ADRA1A**.

Search results

When there are many hits, only the first 50 are shown initially. To see the next 50, click on the **Next** (➡) arrow just under the list of results.

The number to return initially can be configured in the Workbench Preferences, as described in section 4.

Finding the element or folder in the Navigation Area

You can move the focus from a search result to its location in the Navigation Area by doing one of the following:

- Click on a search result while keeping the Alt key depressed.
- Right-click on a search result and click on **Show Location** in the menu presented.

Quick Search history

You can access the 10 most recent searches by clicking the icon (Q-) to the left of the search field (figure 3.17). When one of these is selected, the search is automatically run again.

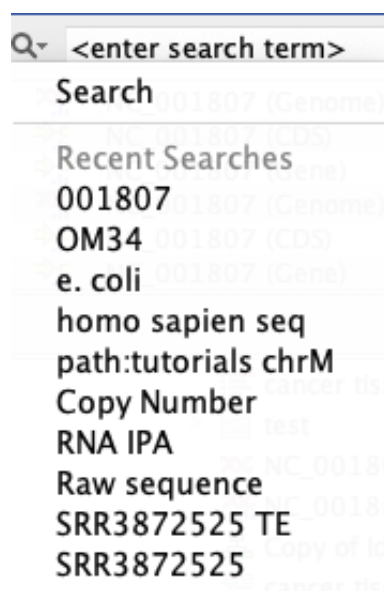


Figure 3.17: Recent searches are listed and can be selected to be re-run by clicking on the icon to the left of the search field.

Searching using CLC URLs

A particular data element or folder can be quickly found by entering its CLC URL into the Quick Search box (figure 3.18). An example of when this is useful is when working using a CLC Server and sharing the location of particular data elements with another person.

To obtain and use a CLC URL:

1. Copy the element or folder in your Navigation Area (section 3.1.6) .
2. Paste the contents of the clipboard (i.e. the copied information) to a place that expects text. The text that will be pasted is the CLC URL for that element or folder.

Examples of where text is expected include a text editor, email, messaging system, etc. It also includes the Quick Search field.

3. Paste the CLC URL into the Quick Search field above the Navigation Area to locate the element or folder it refers to.

If you move the element or folder within the *same* CLC Location, the CLC URL will continue to work.

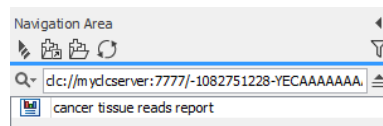


Figure 3.18: Data elements can be located using a CLC URL.

3.4.2 Local Search

The **Local Search** tool is useful when searching large data locations and if you wish to save searches to be run multiple times.

To launch the **Local Search** tool, go to:

Utilities | Local Search (🔍)

or **Ctrl + Shift + F** (⌘ + Shift + F on Mac)

You can choose to search across all CLC Locations or you can specify a single Location to search.

You can search for terms in the names of elements or folders, as well as terms in the path to those elements or folders. If you have defined local (custom) attributes for any of your CLC Locations, the contents of these can also be searched.

More than one search term can be added by clicking on the **Add search parameters** button. To search for terms of the same type (e.g. terms in names), you can just add multiple terms in the same search field, as described below.

Click on the **Search** button to start the search.

Using Local Search for searches based on names

The examples below refer to searches where the following elements are present in a CLC Location.

- E. coli reference sequence
- Broccoli sequence
- Coliform set

Search with a single term to look for any element or folder with a name containing that term.

Example 1: A search for *coli* would return all 3 elements listed above.

Search with two or more terms to look for any element or folder with a name containing all of those terms.

Example 2: A search for *coli* set would return "Coliform set" but not the other two entries listed in the earlier example.

Search with two or more words in quotes to look for any element or folder name containing those words, appearing consecutively, in the order provided. Whole words must be used within quotes, rather than partial terms.

For searching purposes, **words** are the terms on either side of a space, hyphen or underscore in a name. The names of elements and folders are split into words when indexing.

Example 3: A search for "coli reference" would find an element called "E. coli reference sequence".


Example 4: A search for "coli sequence" would not return any of the elements in the example list. In the name "E. coli reference sequence", the words coli and sequence are not placed consecutively, and in "Broccoli sequence", "coli" is a partial term rather than a whole word.

Why only words when searching with quotes? The use of quotes allows quite specific searches to be run quickly, but only using words, as defined by the indexing system.

Tip: Searches with whole words are faster than searching with partial terms. If a term is a word in some names but a partial term in others, the hits found using the complete word are returned first. E.g. searches with the term *cancer* would return elements with names like "cancer reads" and "my cancer sample" before an element with a names like "cancerreads".

Note: Wildcards (* ? ~) are ignored in basic searches. If you wish to define a search using wildcards, use the advanced search functionality of Quick Search.

Saving Local Searches

Local searches can be saved () , so that the same search can be run again easily. Saving a search saves the query, not the results.

The search you just set up can be saved by doing one of the following:

- Choose **Save As...** from under the File menu, or
- Click on the tab of the search view and drag and drop it into a folder in the Navigation Area.


These actions save the search query. (It does not save the search results.)

This can be useful when you run the same searches periodically.

3.5 Backing up data from the CLC Workbench

CLC Workbench data is stored in CLC File Locations. To back up all of your CLC data, back up each of your CLC File Locations (option 1). To back up smaller amounts of data, export selected data elements to a zip file (option 2).

Option 1: Backing up a CLC File Location

Each CLC File Location is associated with a folder on the underlying file system. Hovering the mouse cursor over a CLC File Location () , i.e. a top level folder in the Navigation Area, reveals a tooltip containing the location of that file system folder. Backing up that folder equates to backing up a given CLC File Location.

To recover the data later, retrieve the folder from backup and add it as a CLC File Location (see section 3.1.2).

Note: If the original CLC File Location is still present, the copy from backup will *not* be added. If you wish to add the folder from backup as a new CLC File Location, the options are:

- Remove the original CLC File Location, and then add the folder from backup as a new CLC File Location,
or
- Remove the file called ".clcinfo" from the top level of the folder from backup, and then add the folder as a CLC File Location.

CLC File Location information is stored in an XML file called `model_settings_300.xml` located in the settings folder in the user home area. Further details about this file and how it pertains to data locations in the Workbench can be found in the Workbench Deployment Manual: https://resources.qiagenbioinformatics.com/manuals/workbenchdeployment/current/index.php?manual=Changing_default_location.html.

Option 2: Export a folder of data or individual data elements to a CLC zip file

This option is for backing up smaller amounts of data, for example, backing up certain results, or backing up a whole CLC File Location, that contains a small amount of data.


To export data, click on the Export () button in the top toolbar, or go to:


File | Export ()

Choose zip as the format to export to.

The data to export to the zip file can then be selected.

Further details about exporting data this way is provided in section 8.1.4.

To imported the zip file back into a *CLC Workbench*, click on the Import () button in the top toolbar and select **Standard Import**, or go to:


File | Import () | **Standard Import**

and select *Automatic import* in the Options area.

3.6 Working with AWS S3 using the Remote Files tab

Browsing data in AWS S3 using the Remote Files tab

Available AWS S3 buckets are listed under the **Remote Files** tab, to the right of the Navigation Area tab (figure 3.19).

If AWS S3 buckets are available from more than one source, those sources are listed in a drop-down menu. Possible sources are AWS Connections or public S3 buckets configured in the *CLC Workbench*, or configured in a *CLC Server* that the Workbench is connected to. A server icon () is shown beside sources available via a *CLC Server* (figure 3.20).

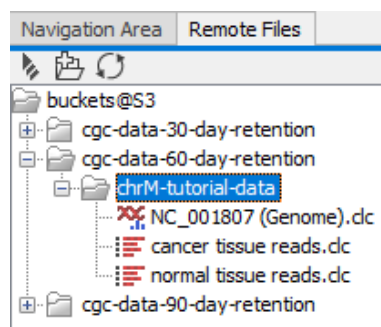


Figure 3.19: AWS S3 buckets you have access to are available under the Remote Files tab.

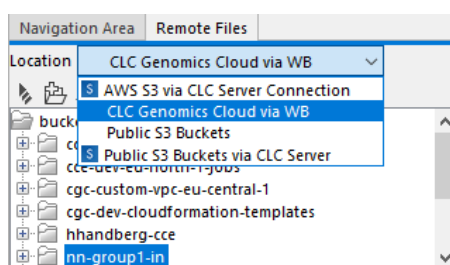


Figure 3.20: This Workbench has a valid AWS Connection and is connected to a CLC Server with a valid AWS Connection. At least one public S3 bucket has been configured in the Workbench and in the CLC Server. The S3 buckets available from the selected source are listed in the Remote Files tab.

Uploading data to AWS S3 using the Remote Files tab

To upload data from your Navigation Area to AWS S3, right-click on a folder in the Remote Files tab and choose the option **Upload to this folder** (figure 3.21).

Upload is sequential. Information about the data upload is shown in the Processes tab, at the bottom left of the Workbench (figure 3.22).

Downloading results via the Remote Files tab

Right-click on an element or elements under the Remote Files tab to download from AWS S3 (figure 3.23). When only a file or files are selected, options to **Download and Open** and **Download and Save** will be available. When a folder has been selected, only the **Download and Save** is available.

After choosing to download the data, you can choose to download it to a CLC File Location or to another area on your system.

To see all the outputs of a particular job that has been run on a *CLC Genomics Cloud* setup, double-click on a `workflow-result.json` file. All the results can then be downloaded and

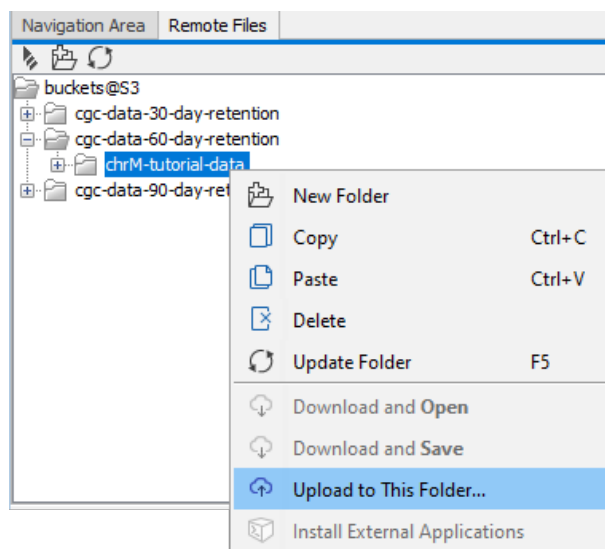


Figure 3.21: To upload data from your Navigation Area to AWS S3, open the Remote Files tab, right-click on the folder you wish to upload data to, and select the option "Upload to This Folder".

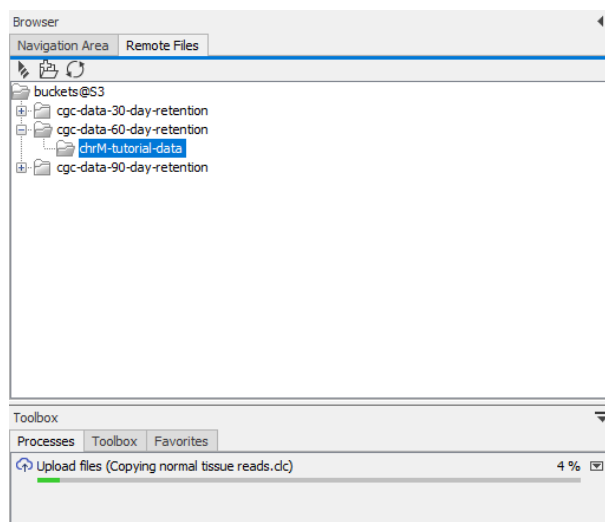


Figure 3.22: After choosing to upload data to S3, the progress of the upload is reported in the Processes tab.

opened from that list, or individual elements can be selected and downloaded. The Execution Log is also available from this list (see figure 3.24).

If the Navigation Tools plugin is installed, bookmarks for items in the Remote Files tab can be made. Double-clicking on bookmarks for individual results files or folders opens the bookmarked items, as standard. Double-clicking on a bookmark for a workflow-result.json file reveals the same list of options as double-clicking on the workflow-result.json file in the Remote Files tab directly. Further details about bookmarks are provided in the Navigation Tools manual at <https://resources.qiagenbioinformatics.com/manuals/navigationtools/current/index.php?manual=Introduction.html>.

Note: AWS charges for downloading data from S3. By default, when the download size exceeds 1 GB, you are prompted for confirmation that you wish to proceed. The size required to trigger

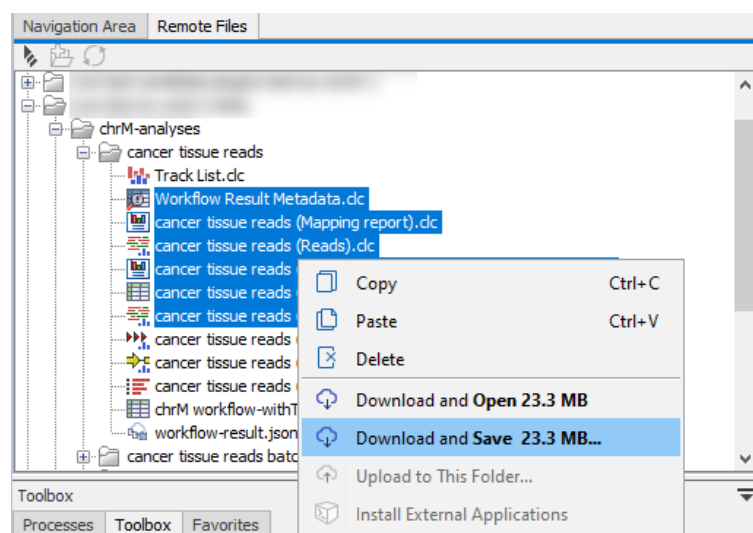


Figure 3.23: Select folders and/or files in the Remote Files tab and right-click to reveal options for downloading that data.

this warning can be changed in the General section of the Workbench Preferences (figure 3.25).

Downloading data using a URL

Pasting an URL into the Workbench Navigation Area will import the files, or a folder containing files, using Standard Import. File types are automatically detected. Thus, as well as CLC format files being available to view and work further with in the Workbench, other files in formats recognized by Standard Import will be imported as CLC format files, allowing them to be worked with in the Workbench. Further details about Standard Import are provided at: https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Standard_import.html.

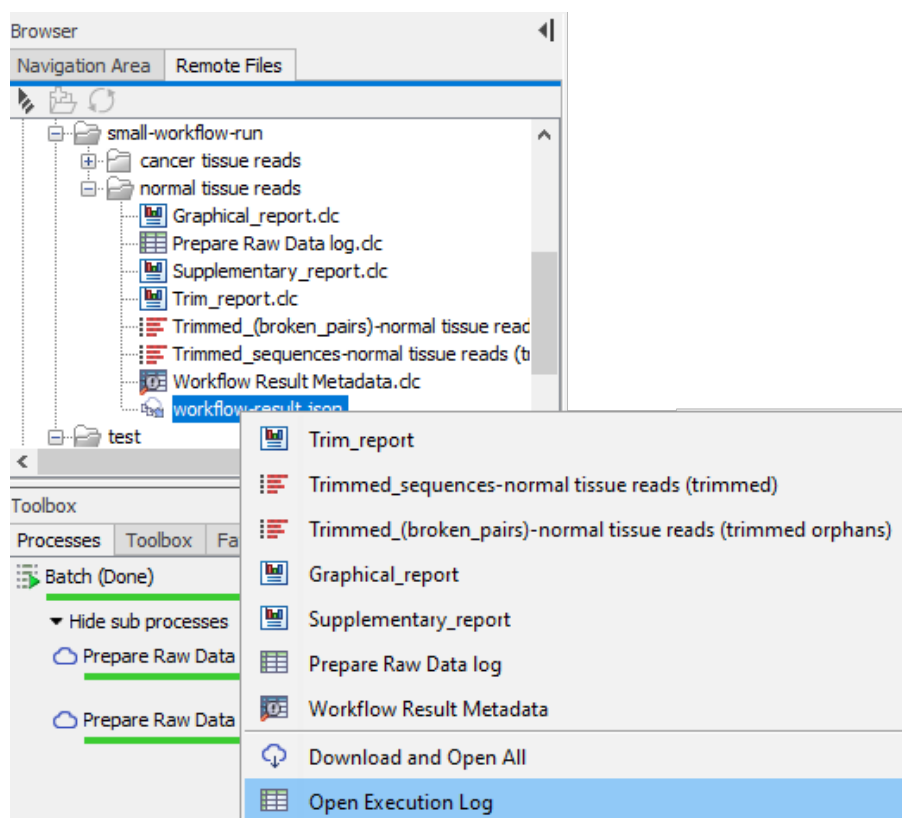


Figure 3.24: Double-click on a workflow-result.json file in the Remote Files tab in the Workbench to reveal a list of all results from a job run in the cloud, as well as the Execution Log. All items can be downloaded and opened from this menu, or individual items can be selected and downloaded.

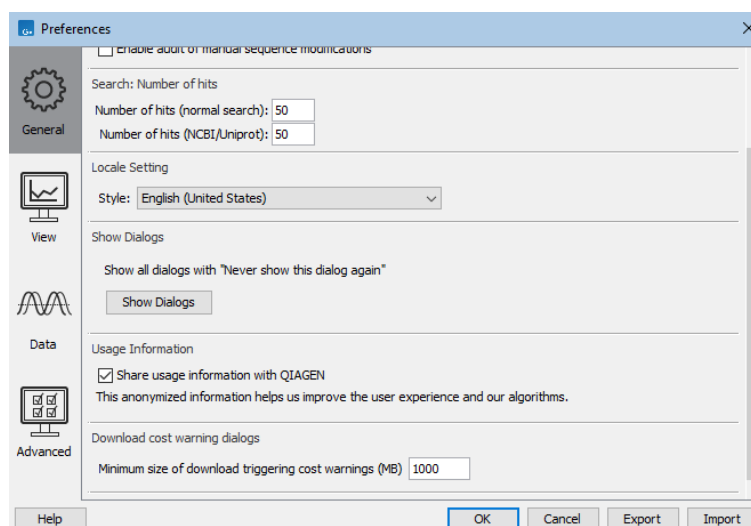


Figure 3.25: The download size above which a cost warning dialog is shown can be adjusted in the Workbench Preferences. The default value is 1000 MB.

Chapter 4

User preferences and settings

Contents

4.1	General preferences	95
4.2	View preferences	96
4.3	Data preferences	98
4.4	Advanced preferences	98
4.5	Export/import of preferences	101
4.6	Side Panel view settings	101

The **Preferences** dialog (figure 4.1) offers opportunities for changing the default settings for different features of the program. Preference settings are grouped under four tabs, each of which is described in the sections to follow.

The **Preferences** dialog is opened in one of the following ways:

Edit | Preferences (⚙)

or **Ctrl + K** (⌘ + ; on Mac)

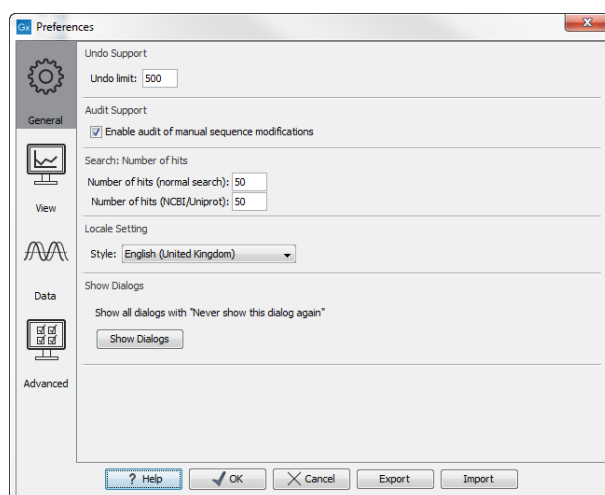


Figure 4.1: Preference settings are grouped under the General, View, Data , and Advanced tabs.

4.1 General preferences

The **General preferences** include:

- **Undo Limit** The default number of undo actions is 500. Undoing and redoing actions is described in section 2.1.3).
- **Audit Support** If this option is checked, all manual editing of sequences will be marked with an annotation on the sequence (figure 4.2). Placing the mouse on the annotation will reveal additional details about the change made to the sequence (see figure 4.3). Note that no matter whether **Audit Support** is checked or not, all changes are recorded in the History log (📄) (see section 2.5).



Figure 4.2: Annotations added when the sequence is edited.

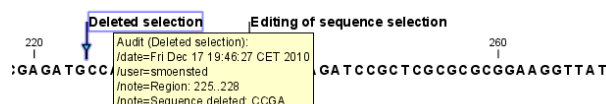


Figure 4.3: Details of the editing.

- **Number of hits** The number of hits shown in *CLC Genomics Workbench*, when e.g. searching NCBI. (The sequences shown in the program are not downloaded, until they are opened or dragged/saved into the Navigation Area).
- **Locale Setting** Specify which country you are located in. This determines how punctuation is used in numbers.
- **Show Dialogs** Many information dialogs have a checkbox with the option: "Never show this dialog again". If you have checked such a box, but later decide you wish to see these notifications, click on the **Show Dialogs** button.
- **Usage information** When this item is checked, anonymous information is shared with QIAGEN about how the Workbench is used. This option is enabled by default.

The information shared with QIAGEN is:

- Launch information (operating system, product, version, and memory available)
- The names of the tools and workflows launched (but not the parameters or the data used)
- Errors (but without any information that could lead to loss of privacy: file names and organisms will not be logged)
- Installation and removal of plugins and modules

The following information is also sent:

- An installation ID. This allows us to group events coming from the same installation. It is not possible to connect this ID to personal or license information.
- A geographic location. This is predicted based on the IP-address. We do not store IP-addresses after location information has been extracted.
- A time stamp

4.2 View preferences

There are five groups of default **View** settings (figure 4.4):

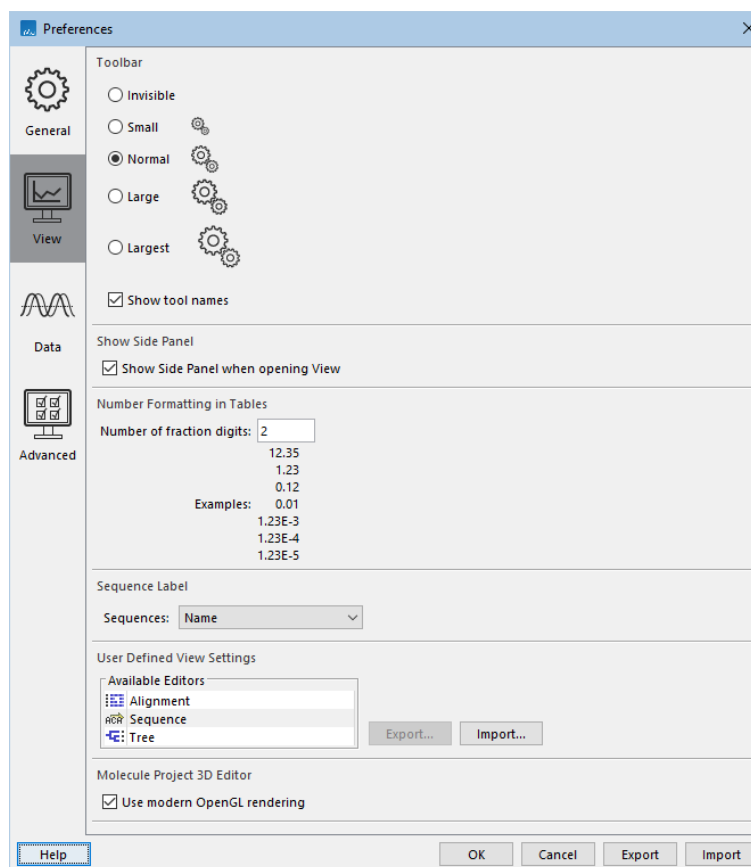


Figure 4.4: Settings relating to views and formatting are found under the View tab in Preferences.

1. **Toolbar** Specify Toolbar icon size, and whether to display names below those icons.
2. **Show Side Panel** Choose whether to display the Side Panel by default when opening a new view.

For any open view, the Side Panel can be collapsed by clicking on the small triangle at the top left side of the settings area or by using the key combination Ctrl + U (⌘ + U on Mac).

3. **Number formatting in tables** Specify the number formatting to use in tables.
The examples below the text field are updated to reflect the option selected.
Open tables must be closed and re-opened for these changes to be applied to them.
4. **Sequence Label** allows you to change the source of the name to use when listing sequence elements in the Navigation Area.
 - Name (the default)
 - Accession (sequences downloaded from databases like GenBank have an accession number).
 - Latin name.

- Latin name (accession).
 - Common name.
 - Common name (accession).
5. **User Defined View Settings** Data types for which custom view settings have been defined are listed here. The default settings to apply to a given data type can be specified. Custom view settings can be exported to a file and imported from a file using the **Export...** and **Import...** button, respectively. To export, select items in the "Available Editors" list and then click on the Export button. A `.vsf` file will be saved to the location you specify. You will have the opportunity to deselect any custom view settings you do not wish to export.

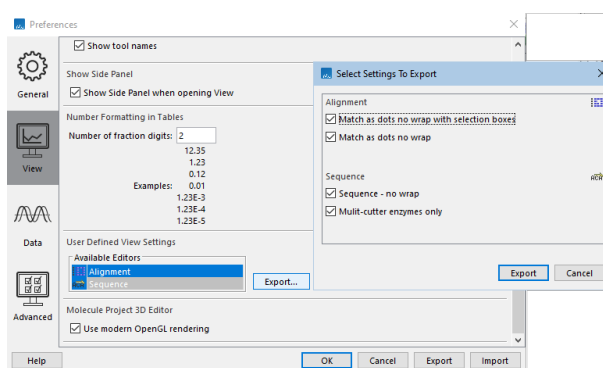


Figure 4.5: Data types for which custom view settings have been defined are listed in the View tab. Settings for multiple views can be exported by selecting them in the list and clicking on the Export... button. Any custom views that should not be included can be deselected before exporting.

To import view settings, select a `.vsf` file and click on the **Import...** button. Specify whether the new settings should be merged with the existing settings or whether they should overwrite the existing settings (figure 4.6). **Note:** If you choose to overwrite existing settings, all existing custom view settings are deleted.

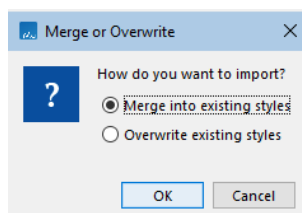


Figure 4.6: When importing view settings, specify whether to merge the new settings with the existing ones or whether to overwrite existing custom settings.

Note: The Export and Import buttons directly under the list of view settings are for exporting and importing just view settings. The buttons at the bottom of the **Preferences** dialog are for exporting all preferences (see section 4.5).

Specifying default view settings for a given data type can also be done using the **Manage View Settings** dialog, described in section 4.6. Export and import can also be done there.

6. **Molecule Project 3D Editor** gives you the option to turn off the modern OpenGL rendering for **Molecule Projects** (see section 17.2).

4.3 Data preferences

The data preferences contain preferences related to interpretation of data:

- Multisite Gateway Cloning primer additions, a list of predefined primer additions for Gateway cloning (see section 23.5.1).

4.4 Advanced preferences

Proxy Settings CLC Genomics Workbench supports the use of an HTTP-proxy and an anonymous SOCKS-proxy.

You have the choice between an HTTP-proxy and a SOCKS-proxy. The CLC Workbench only supports the use of a SOCKS-proxy that does not require authorization.

You can select whether the proxy should also be used for FTP and HTTPS connections.

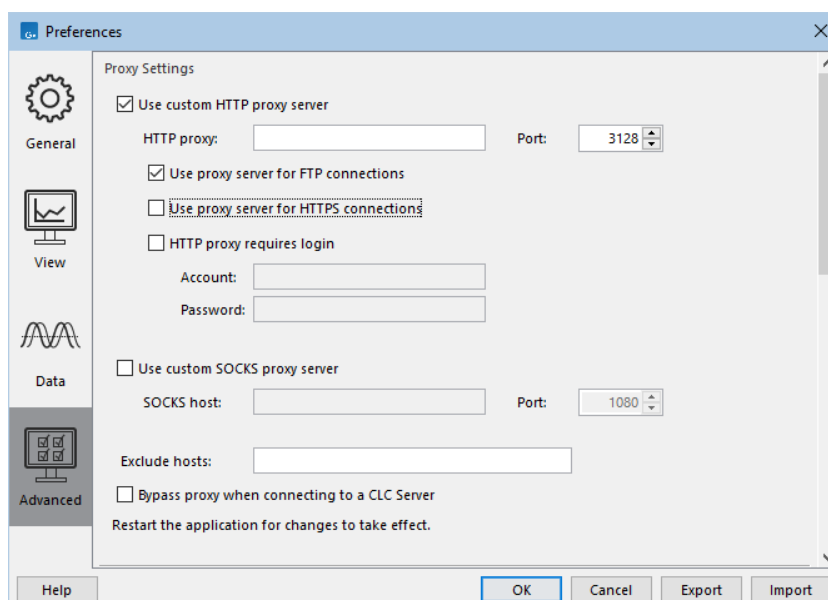


Figure 4.7: Advanced preference settings include proxy configuration options.

List hosts that should be contacted directly, i.e. not via the proxy server, in the **Exclude hosts** field. The value can be a list, with each host separated by a | symbol. The wildcard character * can also be used. For example: *.foo.com|localhost.

The proxy can be bypassed when connecting to a CLC Server by checking the box next to **Bypass proxy when connecting to a CLC Server**.

Workbenches can be preconfigured to bypass the proxy settings when connecting to a CLC Server by configuring this setting in a `proxy.properties` file, where the IP address (not the host name) of the CLC Server is provided in the `proxyexclude` field. See https://resources.qiagenbioinformatics.com/manuals/workbenchdeployment/current/index.php?manual=Per_computer_Workbench_information.html for further details.

If you have problems with these settings, please contact your systems administrator.

Default data location The default location is used when you import a file without selecting a folder or element in the Navigation Area first. It is set to the folder called CLC_Data in the Navigation Area, but can be changed to another data location using a drop down list of data locations already added (see section 3.1.2). Note that the default location cannot be removed, but only changed to another location.

Data Compression CLC format data is stored in an internally compressed format. The application of internal compression can be disabled by unchecking the option "Save CLC data elements in a compressed format". This option is enabled by default. Turning this option off means that data created may be larger than it otherwise would be.

Enabling data compression may impose a performance penalty depending on the characteristics of the hardware used. However, this penalty is typically small, and we generally recommend that this option remains enabled. Turning this option off is likely to be of interest only at sites running a mix of older and newer CLC software, where the same data is accessed by different versions of the software.

Compatibility information:

- A new compression method was introduced with version 22.0 of the CLC Genomics Workbench, CLC Main Workbench and CLC Genomics Server. Compressed data created using those versions can be read by version 21.0.5 and above, but not earlier versions.
- Internal compression of CLC data was introduced in CLC Genomics Workbench 12.0, CLC Main Workbench 8.1 and CLC Genomics Server 11.0. Compressed data created using these versions is not compatible with older versions of the software. Data created using these versions can be opened by later versions of the software, including versions 22.0 and above.

To share specific data sets for use with software versions that do not support the compression applied by default, we recommend exporting the data to CLC or zip format and turning on the export option "Maximize compatibility with older CLC products". See section 8.1.4.

NCBI Integration Without an API key, access to NCBI from a single IP-address is limited to 3 requests per second; if many workbenches use the same IP address when running the Search for Reads in SRA..., Search for Sequences at NCBI and Search for PDB Structures at NCBI tools they may hit this limit. In this case, you can create an API key for NCBI E-utilities in your NCBI account and enter it here.

NCBI BLAST The standard URL for the BLAST server at NCBI is: <https://blast.ncbi.nlm.nih.gov/Blast.cgi>, but it is possible to specify an alternate server URL to use for BLAST searches. Be careful to specify a valid URL, otherwise BLAST will not work.

Read Mapper It is possible to change the size (in MB) of the Read Mapper reference cache.

BaseSpace configuration Import from BaseSpace requires no configuration. The settings in this area are optional.

- **Illumina regional instance ID** Leave this empty to use the default BaseSpace region (USA). Mouse over the field to see the supported regional instances.
- **Client ID and Client secret** Credentials for a BaseSpace app can be entered here.

Reference Data URL to use: Reference data sets available under the QIAGEN Sets tab of the Reference Data Manager are downloaded from the URL provided here. In most cases, this setting should not be changed.

Download to CLC Server via: Relevant when the "On Server" option is chosen in the Reference Data Manager, so that data in the CLC_References area of a *CLC Genomics Server* is used. With the "CLC Server" option chosen (figure 4.8), data downloaded to a *CLC Genomics Server* using the Reference Data Manager is downloaded directly to the Server, without going via the Workbench.

If the *CLC Genomics Server* has no access to the external network, but the *CLC Workbench* does, choose the "CLC Workbench" option. Data is then downloaded to the Workbench and then moved to the reference data area on the Server. The Workbench must be left running throughout the data download process when this option setting is selected.

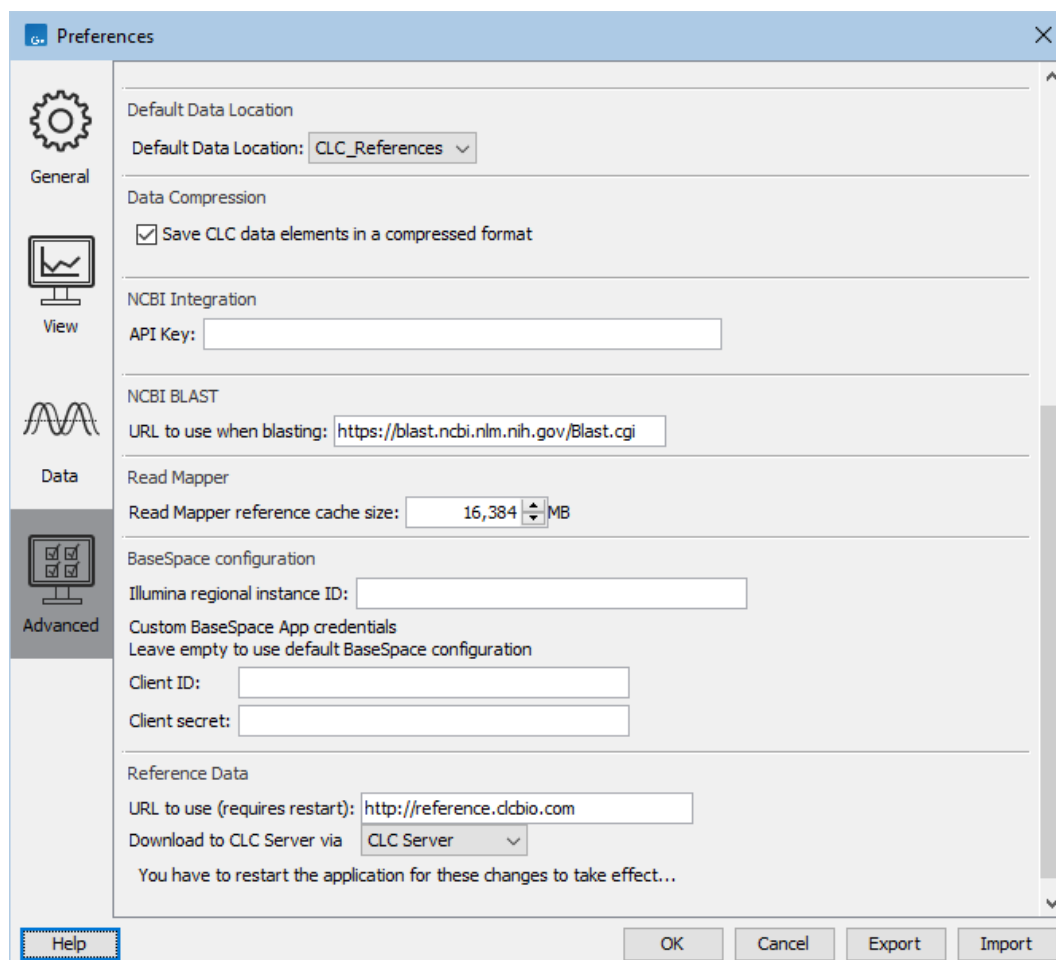


Figure 4.8: Some of the options in the Advanced area of the Workbench Preferences.

4.5 Export/import of preferences

Your *CLC Genomics Workbench* preferences can be exported or other sets of preferences imported. This is a way to share the setup of a *CLC Genomics Workbench* with others, or to back up your own preferences.

To export, open the **Preferences** dialog and click on the Export button at the bottom of the Preferences dialog. Select the relevant preference types and then proceed with the export (figure 4.9). A *.cpf file will be exported, which can be imported by other *CLC Genomics Workbench* users, if desired.

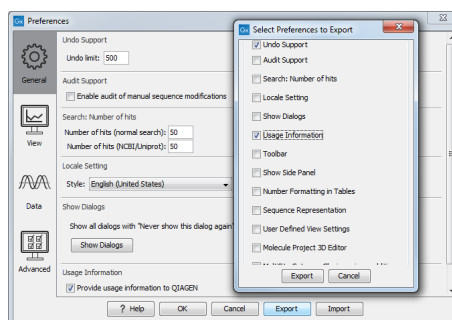


Figure 4.9: Select the preference types to export.

Note: The "User Defined View Settings" option here refers only to information on which view settings to set as the default for each view type. To export the view settings themselves, export a .vsf file from the User Defined View Settings section under the View tab of Preferences, as described in section 4.2.

4.6 Side Panel view settings

The **View Settings...** menu at the bottom, right of the **Side Panel** provides options for saving, applying and managing view settings (figure 4.10).

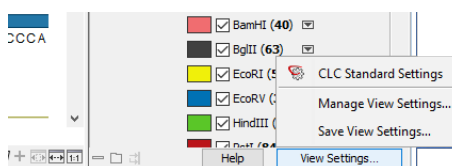


Figure 4.10: Click on the View Settings button at the bottom of a Side Panel to apply a new view settings or to open dialogs for saving and managing view settings.

This section focuses on the functionality provided under the **View Settings...** menu for applying and managing view settings. For general information about Side Panel settings, see section 2.1.6. For view settings specific to tables, including column ordering, see section 9.

Saving view settings

To save view settings for later use, select the option **Save View Settings...** under the **View Settings...** menu. In the resulting dialog (figure 4.11), provide a name for this group of settings, and specify whether it should available for use only by this particular data element, or whether it

should be made available for other elements. In the latter case, you can specify if this group of settings should be used as the default for this view, thereby affecting all elements with that view.

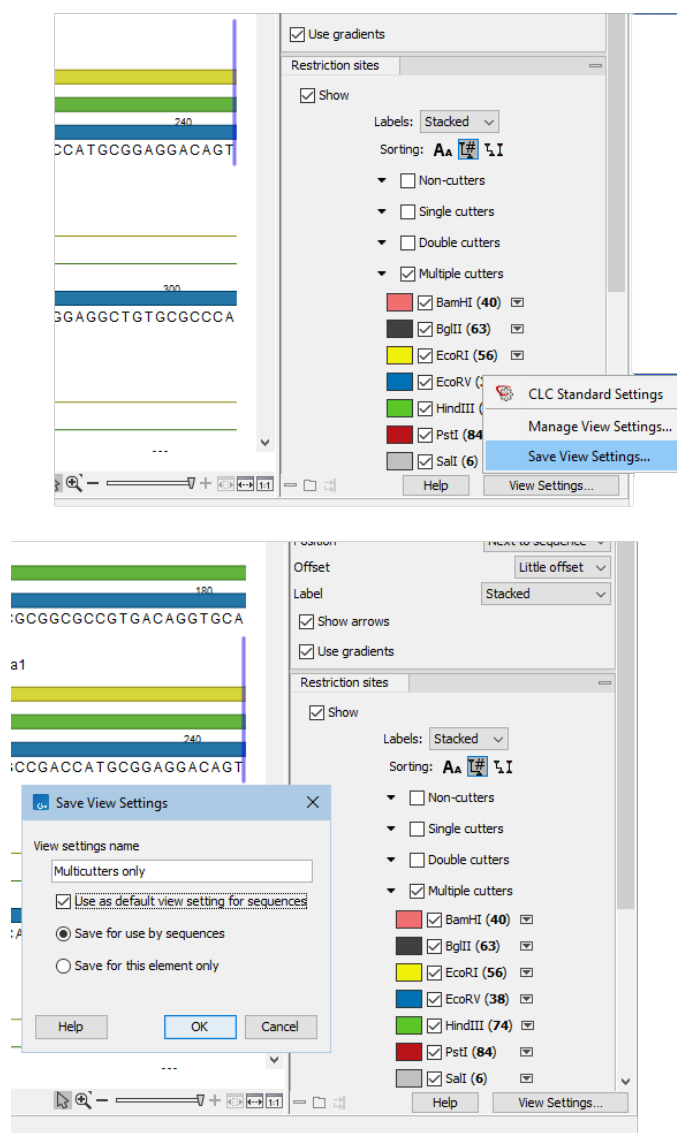


Figure 4.11: Click on the **Save View Settings** menu item (top) to open a dialog for saving the settings. A name needs to be supplied for these settings. The settings can be made available only for the data element being used or for all data elements of that type. Here, these settings have been set as the default for all elements of this type (bottom).

View settings are user-specific. If your *CLC Workbench* is shared by multiple people, you will need to export any custom view settings you wish them to have access to and they will need to import them, as described in the **Sharing view settings** section below.

Applying saved view settings

To apply a new set of saved view settings to an open element, click on the **View Settings...** button and then select from the settings listed (figure 4.12).

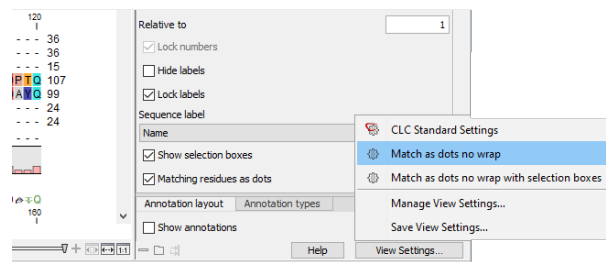


Figure 4.12: Select from saved view settings for the type of element open by clicking on the View Settings button at the bottom of a Side Panel.

View settings named *CLC Standard Settings* are available for each data type. Until custom view settings are saved and set as the default for a given data type, the CLC Standard Settings are used.

Managing view settings

Manage view settings for the open view by selecting the option **Manage View Settings...** from the **View Settings...** menu. In this dialog, you can set a new default, delete view settings, as well as export and import view settings (figure 4.13).

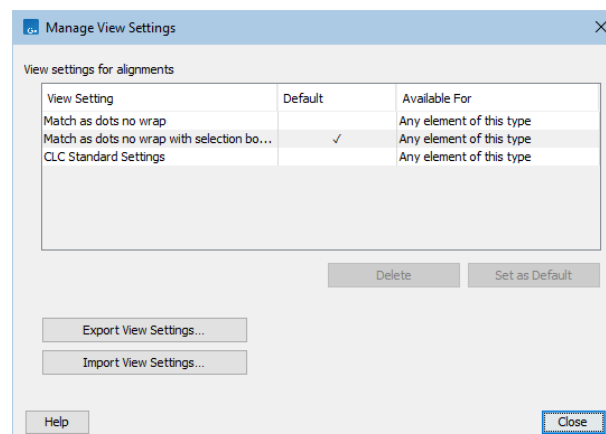



Figure 4.13: In the Manage View Settings dialog, you can specify the default for that view, delete saved settings, as well as export and import view settings.

To browse all custom view settings available in your *CLC Workbench*, open the View tab under **Preferences** (⚙️), as described in section 4.2.

Sharing view settings

To share saved view settings for the open view type with others, click on the **Export...** button in the **Manage View Settings...** dialog. Select the sets to export. When prompted, select a folder to save the exported settings to, and then provide a filename in the field at the top. A file of that name with the suffix `.vsf` will be saved.

To import view settings, open the **Manage View Settings...** dialog, click on the **Import...** button and select a (`.vsf`) file to import.

Note: To export and import view settings for multiple view types, use the functionality under **Preferences** () , described in section [4.2](#).

Chapter 5

Printing

Contents

5.1	Selecting which part of the view to print	106
5.2	Page setup	107
5.3	Print preview	109

CLC Genomics Workbench offers different choices of printing the result of your work.

This chapter deals with printing directly from *CLC Genomics Workbench*. Another option for using the graphical output of your work, is to export graphics (see chapter 8.2) in a graphic format, and then import it into a document or a presentation.

All the kinds of data that you can view in the **View Area** can be printed. The *CLC Genomics Workbench* uses a WYSIWYG principle: What You See Is What You Get. This means that you should use the options in the Side Panel to change how your data, e.g. a sequence, looks on the screen. When you print it, it will look exactly the same way on print as on the screen.

For some of the views, the layout will be slightly changed in order to be printer-friendly.

It is not possible to print elements directly from the **Navigation Area**. They must first be opened in a view in order to be printed. To print the contents of a view:

select relevant view | Print () in the toolbar

This will show a print dialog (see figure 5.1).

In this dialog, you can:

- Select which part of the view you want to print.
- Adjust **Page Setup**.
- See a print **Preview** window.

These three options are described in the three following sections.

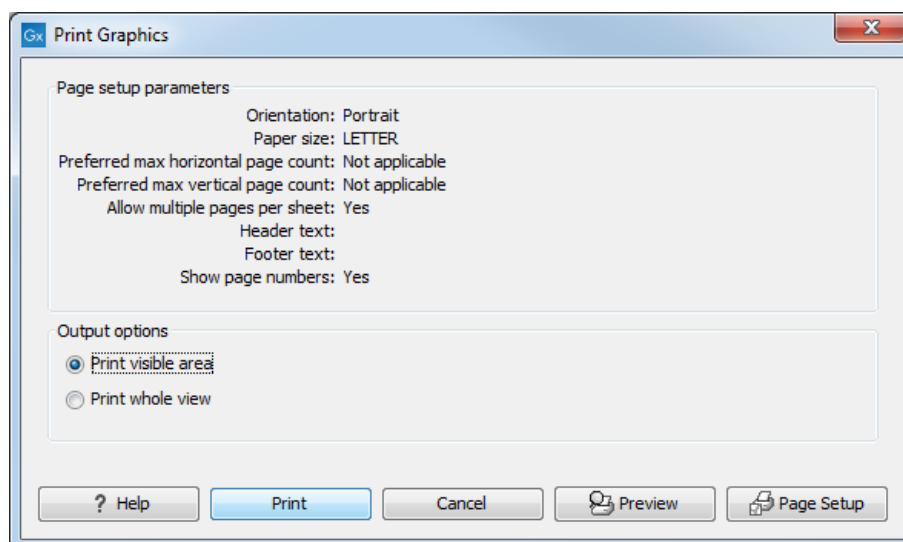


Figure 5.1: The Print dialog.

5.1 Selecting which part of the view to print

In the print dialog you can choose to:

- **Print visible area**, or
- **Print whole view**

These options are available for all views that can be zoomed in and out. In figure 5.2 is a view of a circular sequence which is zoomed in so that you can only see a part of it.

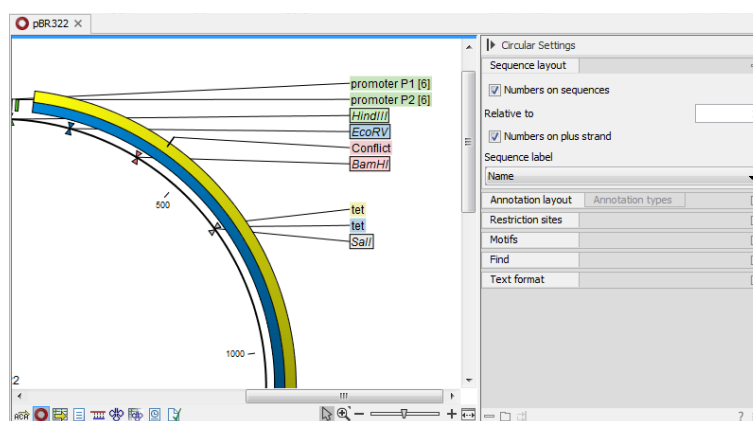


Figure 5.2: A circular sequence as it looks on the screen.

When selecting **Print visible area**, your print will reflect the part of the sequence that is *visible* in the view. The result from printing the view from figure 5.2 and choosing **Print visible area** can be seen in figure 5.3.

On the other hand, if you select **Print whole view**, you will get a result that looks like figure 5.4. This means that you also print the part of the sequence which is not visible when you have zoomed in.

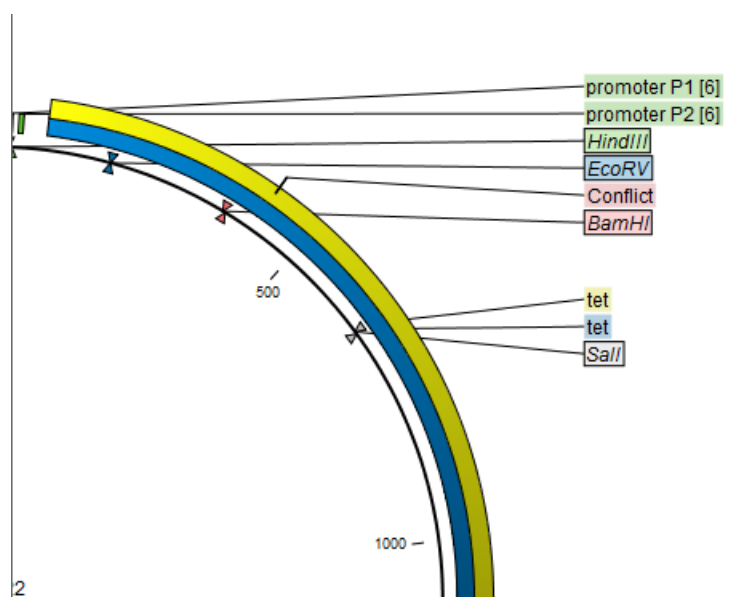


Figure 5.3: A print of the sequence selecting Print visible area.

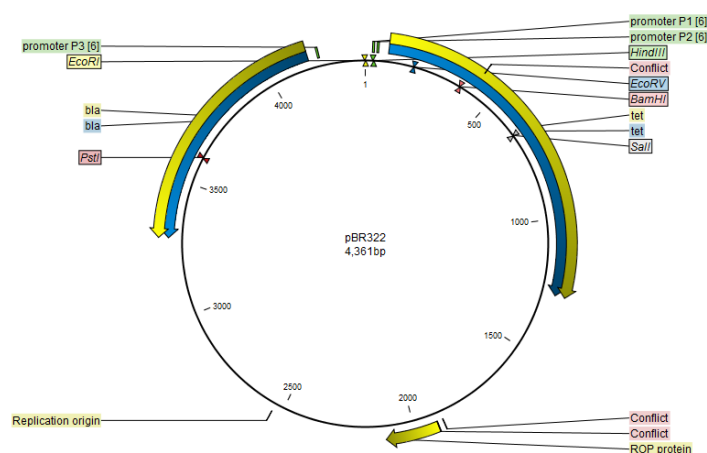


Figure 5.4: A print of the sequence selecting Print whole view. The whole sequence is shown, even though the view is zoomed in on a part of the sequence.

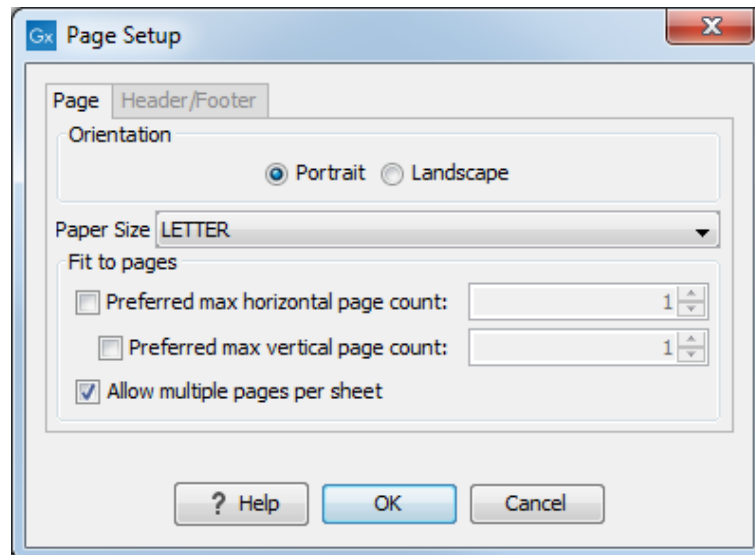
5.2 Page setup

No matter whether you have chosen to print the visible area or the whole view, you can adjust page setup of the print. An example of this can be seen in figure 5.5

In this dialog you can adjust both the setup of the pages and specify a header and a footer by clicking the tab at the top of the dialog.

You can modify the layout of the page using the following options:

- **Orientation.**
 - **Portrait.** Will print with the paper oriented vertically.
 - **Landscape.** Will print with the paper oriented horizontally.
- **Paper size.** Adjust the size to match the paper in your printer.

Figure 5.5: *Page Setup*.

- **Fit to pages.** Can be used to control how the graphics should be split across pages (see figure 5.6 for an example).
 - **Horizontal pages.** If you set the value to e.g. 2, the printed content will be broken up horizontally and split across 2 pages. This is useful for sequences that are not wrapped
 - **Vertical pages.** If you set the value to e.g. 2, the printed content will be broken up vertically and split across 2 pages.

Figure 5.6: An example where *Fit to pages horizontally* is set to 2, and *Fit to pages vertically* is set to 3.

Note! It is a good idea to consider adjusting view settings (e.g. **Wrap** for sequences), in the **Side Panel** before printing. As explained in the beginning of this chapter, the printed material will look like the view on the screen, and therefore these settings should also be considered when adjusting **Page Setup**.

Header and footer Click the **Header/Footer** tab to edit the header and footer text. By clicking in the text field for either **Custom header text** or **Custom footer text** you can access the auto

formats for header/footer text in **Insert a caret position**. Click either **Date**, **View name**, or **User name** to include the auto format in the header/footer text.

Click **OK** when you have adjusted the **Page Setup**. The settings are saved so that you do not have to adjust them again next time you print. You can also change the **Page Setup** from the **File** menu.

5.3 Print preview

The preview is shown in figure 5.7.

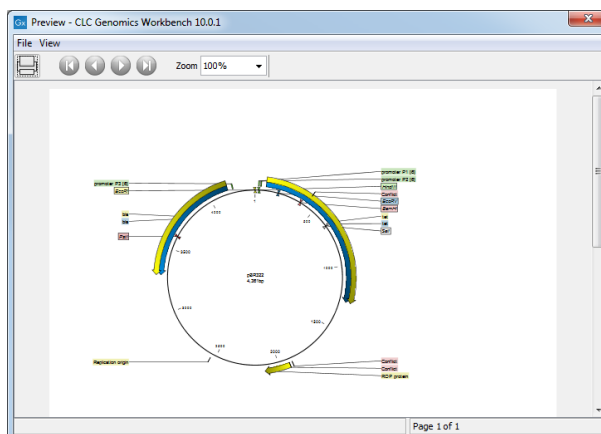



Figure 5.7: *Print preview.*

The **Print preview** window lets you see the layout of the pages that are printed. Use the arrows in the toolbar to navigate between the pages. Click Print () to show the print dialog, which lets you choose e.g. which pages to print.

The **Print preview** window is for preview only - the layout of the pages must be adjusted in the **Page setup**.

Chapter 6

Connections to other systems

Contents

6.1 CLC Server connection	110
6.1.1 CLC Server data import and export	113
6.2 AWS Connections	114

Under the Connections menu are tools for connecting the *CLC Genomics Workbench* to other systems.

6.1 CLC Server connection

Using a *CLC Server*, data can be stored centrally and analyses run on a central machine rather than on a personal computer. After logging into the *CLC Server* from a *CLC Workbench*:

- Data in *CLC Server* locations will be listed in the Workbench Navigation Area.
- When launching analyses that can be run on the *CLC Server*, you will be offered the choice of running them using the Workbench or the *CLC Server*.
- Workflows installed on the *CLC Server* will be available to launch.
- External applications configured and enabled on the *CLC Server* will be available to launch and to include in workflows.

Logging into a CLC Server from a CLC Workbench

To log in to a *CLC Server*, open the **CLC Server Connection** dialog by going to:

File | CLC Server Connection (S)

or click on the server icon ((S)) at the bottom left hand side of the Workbench frame. This icon is blue when there is an active server connection or grey when there is not.

Fill in the details requested in the **CLC Server Connection** dialog (figure 6.1) and click on the **Log In** button to connect to the *CLC Server*. Information about logging into an SSL-enabled *CLC Server* are provided below.

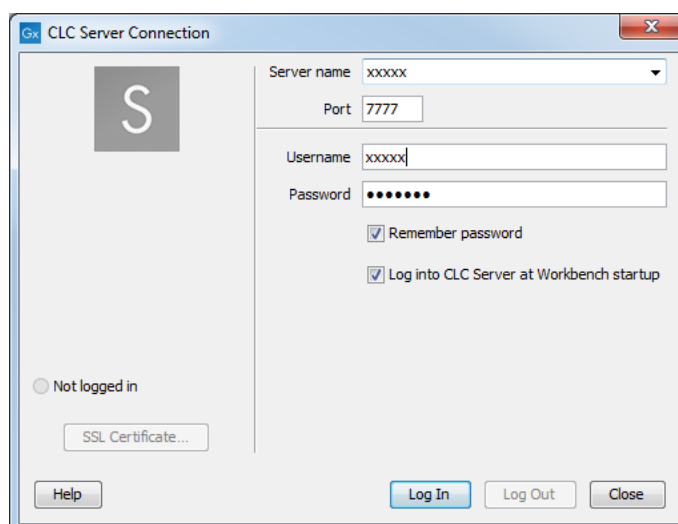



Figure 6.1: Logging into a CLC Server using the default port, 7777

Your username and the server details are saved between Workbench sessions. To save your password also, check the **Remember password** box.

To connect to the *CLC Server* automatically when the *CLC Workbench* starts up, check the **Log into CLC Server at Workbench startup** box. This option is only available when the **Remember password** option has been enabled.

Checking the status of a CLC Server connection

The status of an active connection can be seen by opening the **CLC Server Connection** dialog, as described above, or by hovering the mouse cursor over the server icon () at the bottom left hand side of the Workbench frame (figure 6.2).

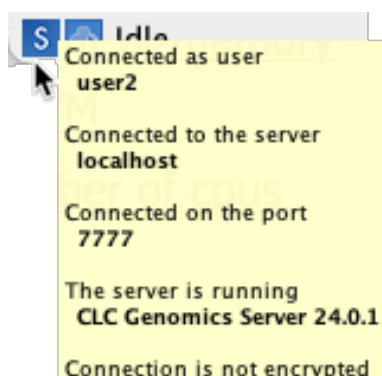


Figure 6.2: Hover the mouse cursor over the Server icon in the bottom left corner of the Workbench frame to quickly view the status of the server connection.

Key documentation about working with a CLC Server

Further information about working with a *CLC Server* from the *CLC Genomics Workbench*:

- Launching tasks on a *CLC Server*: section [12.1.1](#).

- Monitoring processes sent to the *CLC Server* from a *CLC Workbench*: section 2.4
- Viewing and working with data held on a *CLC Server*: section 3.1,
- Deleting data held on a *CLC Server*: section 3.1.8.
- Importing data to and exporting data from a *CLC Server* is described in section 6.1.1.

For those logging into the *CLC Server* as a user with administrative privileges, an option called *Manage Server Users and Groups...* will be available. This is described at https://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=User_authentication_using_Workbench.html.

Connecting to a *CLC Server* using SSL

To log into a *CLC Server* with SSL enabled, provide the secure port in the *Port* field of the **CLC Server Connection** dialog in the *Workbench*.

If SSL is detected on the port provided, the *CLC Server*'s certificate will be verified before the connection is established. A warning is displayed if the certificate is not signed by a recognized Certificate Authority (CA) (figure 6.3). When such a certificate has been accepted once, the warning will not appear again.

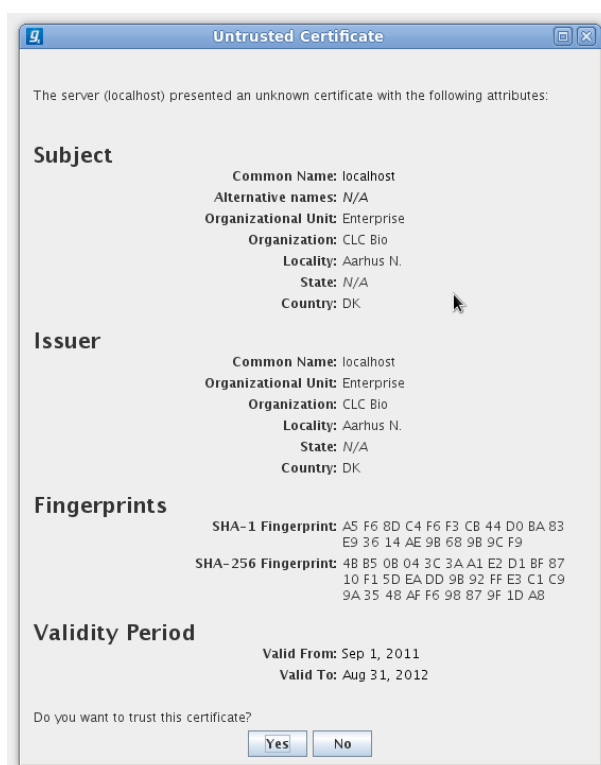


Figure 6.3: A warning is shown when the certificate is not signed by a recognized CA.

The certificate details can be viewed again later by clicking on the **SSL Certificate** button in the **CLC Server Connection** dialog.

The connection status information in the tooltip revealed when hovering over the (S) icon in the bottom left corner of the Workbench frame includes whether the connection is encrypted or not (figure 6.4).

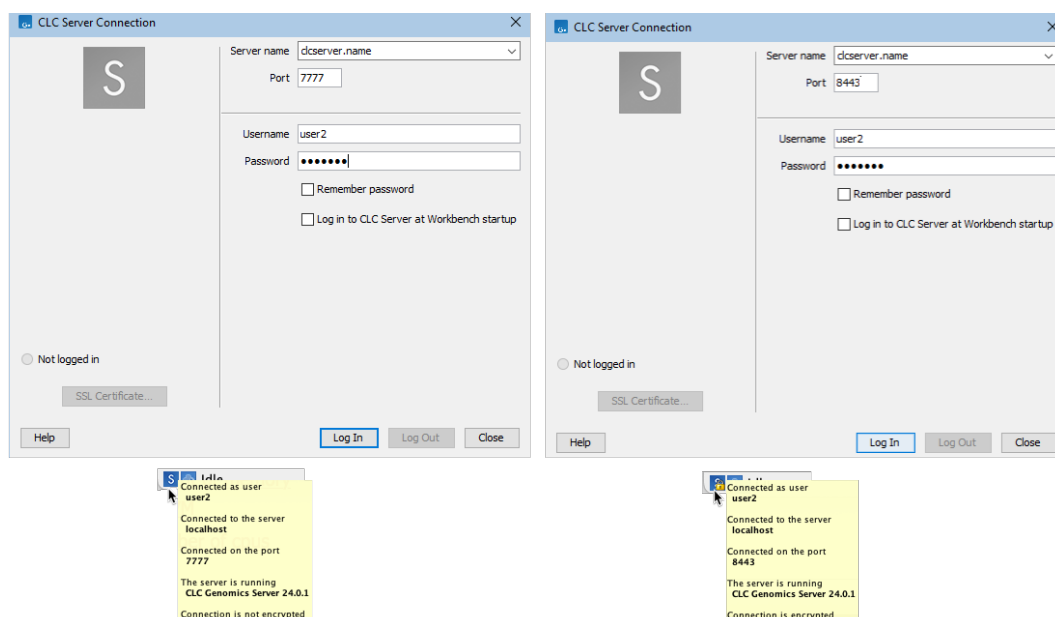


Figure 6.4: Login details and connection status information for an unencrypted connection to a CLC Server (left) and an encrypted connection (right). A padlock on the server icon in the bottom left corner of the Workbench frame also indicates the connection is encrypted.

6.1.1 CLC Server data import and export

This section covers general points related to importing data to and exporting data from a CLC Server when working with the CLC Genomics Workbench. General information about data import and data export is in chapter 7 and chapter 8, respectively.

Detailed information about CLC Server configuration is in the server administration manual at: <https://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=Introduction.html>

Data import to a CLC Server

The general location of the data to import is specified from a drop-down list (figure 6.5). If direct data transfer from client systems is enabled on the CLC Server, the option "File system" is included in that list, allowing files on the local system to be selected for import.

When the option represented in figure 6.5 as "Server Name (CLC Genomics Server)" is selected, the available import/export directories on that CLC Server will be listed in the "Server files" area. When an AWS location is selected, the available S3 buckets will be listed. When BaseSpace is selected, you will be prompted to log into BaseSpace before proceeding.

When "File system" is selected, the CLC Genomics Workbench must maintain its connection to the CLC Server during the first part of the import process, data upload. Further details about this are in section 2.4.

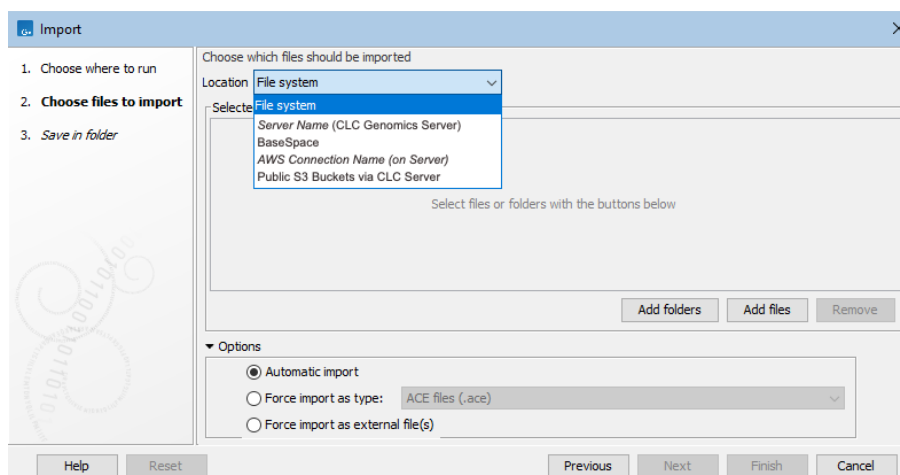


Figure 6.5: When an import is run on a CLC Server, the list of locations that data can be imported from reflects the server configuration.

Note that when importing data from an AWS S3 bucket, the data is first downloaded from AWS, which AWS charges for.

Data export from a CLC Server

When the *CLC Genomics Workbench* is connected to a *CLC Server*, data in CLC Server File System Locations can be exported. The choice of execution environment, as well as settings on the Workbench and Server, affect the locations data can be selected for export from and where the exported files can be saved to:

- **Running the export on the Workbench:**

- Data in Workbench or Server CLC File System Locations can be selected for export.
- Exported files can be saved to areas the *CLC Genomics Workbench* has access to, including AWS S3 buckets if an AWS S3 Connection has been configured in the *CLC Genomics Workbench*.

- **Running the export on the CLC Server or Grid via CLC Server:**

- Data from Server File System Locations can be selected for export.
- Exported files can be saved to Server import/export directories or to an AWS S3 bucket if an AWS Connection has been configured in the *CLC Server*.

6.2 AWS Connections

AWS connections are used when:

- Accessing AWS S3 buckets, to import data from or export data to.
- Submitting analyses to a *CLC Genomics Cloud* setup, if available on that AWS account.

Configuring access to your AWS accounts requires AWS IAM credentials. Configuring access to public S3 buckets requires only the name of the bucket.

Working with stored data in AWS S3 buckets via the Workbench is of particular relevance when submitting jobs to run on a *CLC Genomics Cloud* setup making use of functionality provided by the *CLC Cloud Module*.

When launching workflows to run locally using on-the-fly import and selecting files from AWS S3, the files selected are first downloaded to a temporary folder and are subsequently imported.

All traffic to and from AWS is encrypted using a minimum of TLS version 1.2.

Configuring access to AWS resources

To configure an AWS Connection or to configure access to public AWS S3 buckets, go to:

Connections | AWS Connections (🔗)

Already configured AWS connections and their status, and public S3 buckets are listed (figure 6.6). Editing or removal of these configurations is done from here.

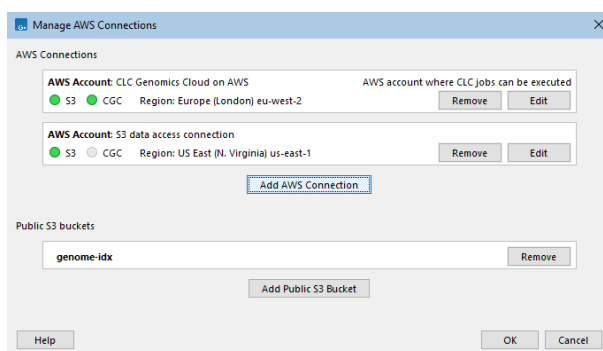


Figure 6.6: The configuration dialog for AWS connections. Here, two valid AWS connections, their status, and a public S3 bucket are listed.

Configuring an AWS Connection

To configure a new AWS Connection, click on the **Add AWS Connection** button and enter the following information in the dialog (figure 6.7):

- **Connection name:** A short name of your choice, identifying the AWS account. This name will be shown as the name of the data location when importing data to or exporting data from Amazon S3.
- **Description:** A description of the AWS account (optional).
- **AWS access key ID:** The access key ID for programmatic access for your AWS IAM user.
- **AWS secret access key:** The secret access key for programmatic access for your AWS IAM user.
- **AWS region:** An AWS region. Select from the drop-down list.

- **AWS partition:** The AWS partition for your account.

The dialog continuously validates the settings entered. When they are valid, the Status box will contain the text "Valid" and a green icon will be shown. Click on **OK** to save the settings.

Figure 6.7: Configuration of an AWS Connection in a CLC Workbench

AWS connection status is indicated using colors. Green indicates the connection is valid and ready for use. Connections to a *CLC Genomics Cloud* are indicated in the CGC column (figure 6.6). To submit analyses to the *CLC Genomics Cloud*, the *CLC Cloud Module* must be installed and a license for that module must be available.

AWS credentials entered are stored, obfuscated, in Workbench user configuration files.

Note: Multiple AWS Connections using credentials for the same AWS account cannot be configured.

Adding a public S3 bucket

To add a public bucket, click on the **Add Public S3 button** and provide the public bucket name (figure 6.8).

Figure 6.8: Provide a public AWS S3 bucket name to enable access to data in that public bucket.

Importing data from AWS S3

AWS S3 buckets for each AWS Connection and public S3 bucket configured are available in relevant import tool wizards and in workflow launch wizards when using on-the-fly import (see Launching workflows individually and in batches (section 14.3)).

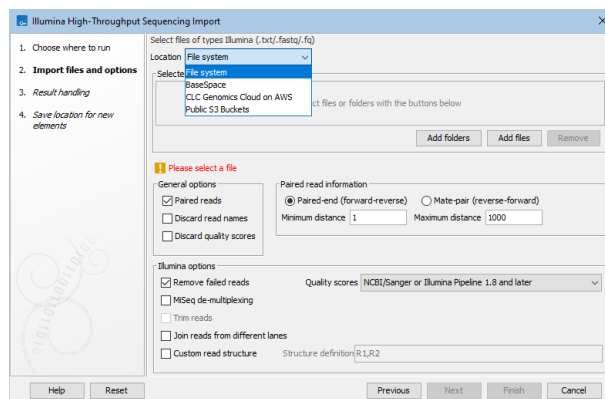


Figure 6.9: Files in local or remote locations can be selected for import by the Illumina importer of the CLC Genomics Workbench.

AWS S3 buckets can be browsed under the **Remote Files** tab next to the **Navigation Area** tab, in the top left side of the Workbench (see section 3.6).

Exporting data to AWS S3

To export data to an AWS S3 bucket, launch the exporter, and when prompted for an export location, select the relevant option from the drop-down menu (figure 6.10).

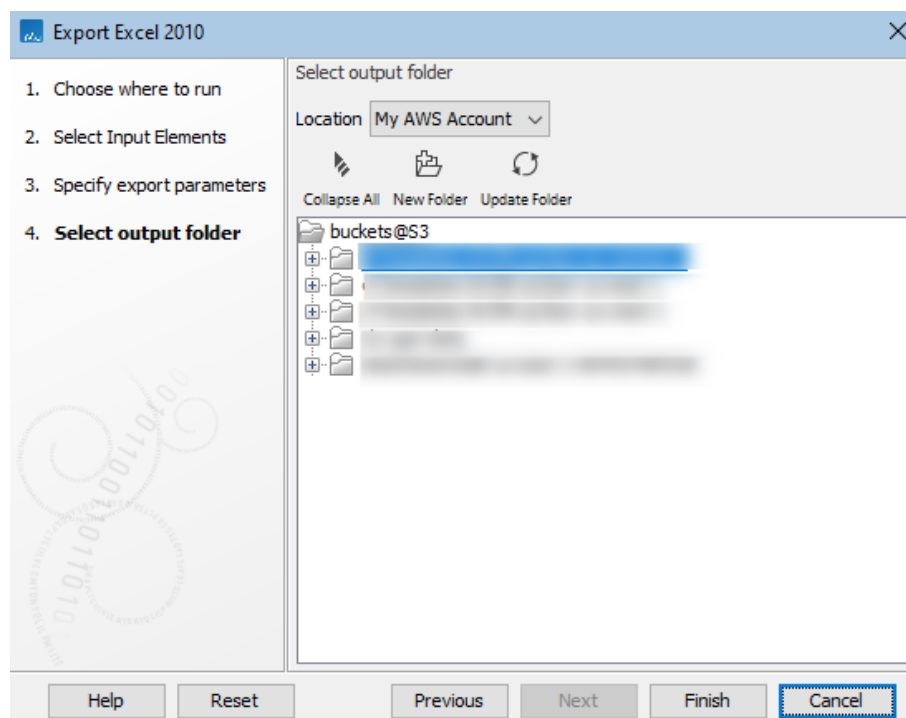


Figure 6.10: After an AWS connection is selected when exporting, you can select the S3 bucket and location within that bucket to export to.

Chapter 7

Importing data

Contents

7.1	Standard import	120
7.2	Import tracks	121
7.2.1	GFF3 format	124
7.2.2	VCF import	128
7.3	Import NGS Reads	131
7.3.1	Illumina	132
7.3.2	Oxford Nanopore	137
7.3.3	PacBio Long Reads	138
7.3.4	PacBio Onso	139
7.3.5	Element Biosciences	141
7.3.6	Ion Torrent	142
7.3.7	MGI/BGI	143
7.3.8	Singular Genomics	144
7.3.9	Ultima Genomics	145
7.3.10	General notes on handling paired data	147
7.3.11	General notes on UMIs	148
7.4	Import other high-throughput sequencing data	149
7.4.1	Fasta read files	149
7.4.2	Sanger sequencing data	150
7.4.3	SAM, BAM and CRAM mapping files	152
7.5	Import RNA spike-in controls	155
7.6	Import Primer Pairs	156

Many data formats are supported for import into the *CLC Genomics Workbench*. Data types that are not recognized are imported as "external files". Such files are opened in the default application for that file type on your computer (e.g. Word documents will open in Word). This chapter describes import of data, with a focus on import of common bioinformatics data formats.

For information about importing NGS sequencing reads, see section [7.3](#). Importing other sequencing data types is described in section [7.4](#). For import of tracks, please see section [7.2](#).

Where data can be imported from

Data can be imported from any location accessible via the system the *CLC Genomics Workbench* is installed on, including from AWS S3 if AWS Connections are configured (see section 6.2). Importing data from AWS S3 involves downloading files from AWS, which AWS charges for.


If connected to a *CLC Server*, data can be imported from areas accessible via the *CLC Server* (see section 6.1.1).

Using the Illumina importer, data can also be imported directly from BaseSpace (see section 7.3.1).

7.1 Standard import

CLC Genomics Workbench has support for a wide range of bioinformatic data formats. See section 1.1 for a list.

The Standard Import tool supports imports of many standard bioinformatics and tabular formats, as well as Excel format files. The full list of supported formats is provided in the tool, as described further below. Information about supported formats is also provided in the appendix section 1.1.

Standard Import can be launched by clicking on the **Import**  icon in the Toolbar and choosing **Standard Import**.

Alternatively, go to **File** | **Import**  | **Standard Import** .

Select files to import by clicking on the **Add files** button. Alternatively, specify folders containing files to import by clicking on the **Add folders** button (figure 7.1).

The default option is **Automatic import**. The file format is automatically detected based on a combination of the file extension (e.g. .fa for fasta) and detection of file contents specific to particular formats. Based on this, the relevant importer is run. The particular importer used is recorded in the element history. If the format is not supported, the file is imported as an external file, that is, it is imported to the CLC Location in its original format. See section 3.2 for details about working with such files.

You can specify the format explicitly using the option **Force import as type**. When choosing this option, the full list of supported data types is provided in a drop-down list.

The **Force import as external file** option can be useful if you are trying to import a standard format file, such as a text file, but it is being detected as bioinformatics format file, such as sequence data.

Standard Import is also used to import files that are dragged from a file browser and dropped into the Navigation Area. In this case, the file format is automatically detected. To force the file type, launch the tool explicitly, as described at the start of this section.

Importing by copy/pasting text

Data can be copied and pasted directly into the Navigation Area.

Copy text | **Select a folder in the Navigation Area** | **Paste** 

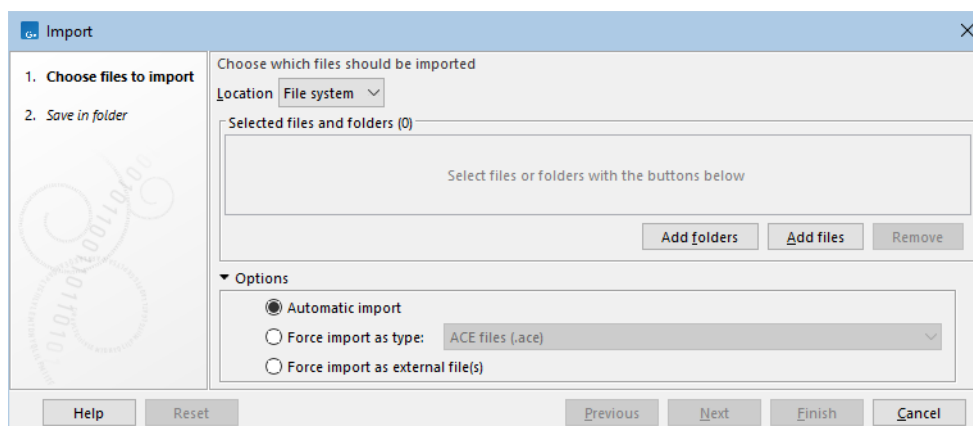


Figure 7.1: The Standard Import tool.

Standard import with automatic format detection is run using the pasted content as input.

This is a fast way to import data, but importing files as described above is less error prone, and thus generally recommended instead.

7.2 Import tracks

Tracks (see chapter 27) are imported in a special way, because extra information is needed in order to interpret the files correctly.

To import tracks, click on the **Import** (📁) icon in the Toolbar and choose **Tracks**. This will open a dialog as shown in figure 7.2.

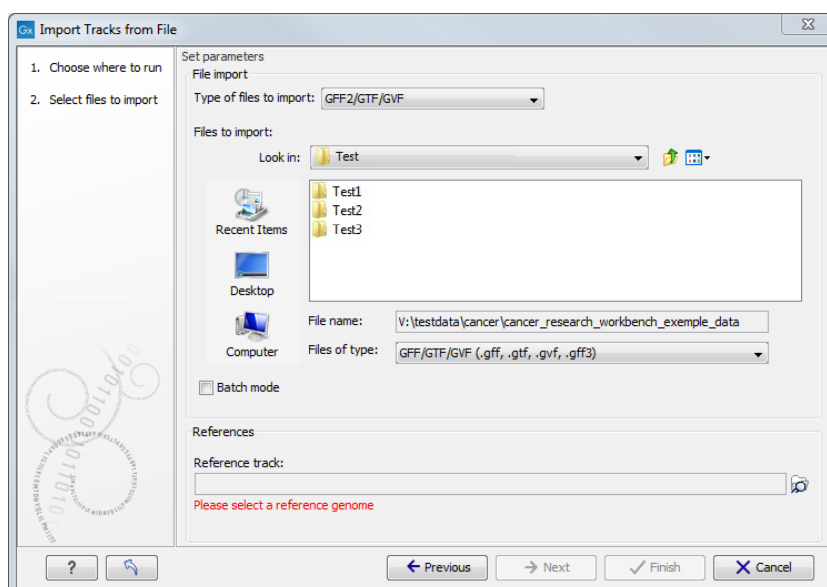


Figure 7.2: Select files to import.

At the top, you select the file type to import. Below, select the files to import. If import is performed with the batch option selected, then each file is processed independently and separate tracks are produced for each file. If the batch option is not selected, then variants for all files will

be added to the same track (or tracks in the case VCF files including genotype information). The formats currently accepted are:

FASTA This is the standard fasta importer that will produce a sequence track rather than a standard fasta sequence. Please note that this could also be achieved by importing using Standard Import (see section 7.1). and subsequently converting the sequence or sequence list to a track (see section 27.8).

GFF2/GTF/GVF A GFF2/GTF file does not contain any sequence information, it only contains a list of various types of annotations. A GVF file is similar to a GFF file but uses Sequence Ontology to describe genome variation data (see <https://github.com/The-Sequence-Ontology/Specifications/blob/master/gvf.md>). For these formats, the importer adds the annotation in each of the lines in the file to the chosen sequence, at the position or region in which the file specifies that it should go, and with the annotation type, name, description etc. as given in the file. However, special treatment is given to annotations of the types CDS, exon, mRNA, transcript and gene. For these, the following applies:

- A gene annotation is generated for each gene_id. The region annotated extends from the leftmost to the rightmost positions of all annotations that have the gene_id (gtf-style).
- CDS annotations that have the same transcriptID are joined to one CDS annotation (gtf-style). Similarly, CDS annotations that have the same parent are joined to one CDS annotation (gff-style).
- If there is more than one exon annotation with the same transcriptID these are joined to one mRNA annotation. If there is only one exon annotation with a particular transcriptID, and no CDS with this transcriptID, a transcript annotation is added instead of the exon annotation (gtf-style).
- Exon annotations that have the same parent mRNA are joined to one mRNA annotation. Similarly, exon annotations that have the same parent transcript, are joined to one transcript annotation (gff-style).

Note that genes and transcripts are linked by name only (not by position, ID etc).

For a comprehensive source of genomic annotation of genes and transcripts, we refer to the Ensembl web site at <https://www.ensembl.org/info/data/ftp/index.html>. On this page, you can download GTF files that can be used to annotate genomes for use in other analyses in the *CLC Genomics Workbench*. You can also read more about these formats at <https://gmod.org/wiki/GFF3>, <https://mblab.wustl.edu/GTF22.html> and <https://genomebiology.biomedcentral.com/articles/10.1186/gb-2010-11-8-r88>.

GFF3 A GFF3 file contains a list of various types of annotations that can be linked together with "Parent" and "ID" tags. Learn more about how the *CLC Genomics Workbench* handles GFF3 format in section 7.2.1.

VCF This is the file format used for variants by the 1000 Genomes Project and it has become a standard format. Read about VCF format here <https://samtools.github.io/hts-specs/VCFv4.2.pdf>. Learn how to access data at <https://www.internationalgenome.org/data/#DataAccess>. Learn more about how the *CLC Genomics Workbench* handles VCF format in section 7.2.2.

BED This format is typically used for simple annotations, such as target regions for sequence capture methods. The format is described at <https://genome.ucsc.edu/FAQ/FAQformat.html#format1>. The 3 required BED fields (chrom, chromStart and chromEnd) must be present as the first 3 columns in the file to be imported. Optional BED fields, present in the order stipulated in the UCSC format, are also imported, with the exceptions listed below. If there are additional columns, these are imported and assigned the header "Var" followed by a number, e.g. Var1, Var2, etc.

Exceptions:

- The itemRgb field is ignored.
- The thickStart and thickEnd fields are used to check if the annotation is valid, but are otherwise ignored.

Wiggle The Wiggle format as defined by UCSC (<https://genome.ucsc.edu/goldenPath/help/wiggle.html>) is used to hold continuous data like conservation scores, GC content etc. When imported into the *CLC Genomics Workbench*, a graph track is created. An example of a popular Wiggle file is the conservation scores from UCSC which can be download for human from <https://hgdownload.cse.ucsc.edu/goldenPath/hg19/phastCons46way/>.

UCSC variant database table dump Table dumps of **variant** annotations from the UCSC can be imported using this option. Mainly files ending with `.txt.gz` on this list can be used: <https://hgdownload.cse.ucsc.edu/goldenPath/hg19/database/>. Please note that importer is for variant data and is not a general importer for all annotation types. This is mainly intended to allow you to import the popular *Common SNPs* variant set from UCSC. The file can be downloaded from the UCSC web site here: <https://hgdownload.cse.ucsc.edu/goldenPath/hg19/database/snp138Common.txt.gz>. Other sets of variant annotation can also be downloaded in this format using the UCSC Table Browser.

COSMIC variation database This lets you import the COSMIC database, which is a well-known publicly available primary database on somatic mutations in human cancer. The file can be downloaded from the COSMIC web site here: <https://cancer.sanger.ac.uk/cancergenome/projects/cosmic/download>, You must first register to download the database. The following tsv format COSMIC files can be imported using the option **COSMIC variation database** under Import->Tracks:

- COSMIC Targeted Screens Mutants:
 - Cosmic_CompleteTargetedScreensMutant_vXXX_GRCh37.tsv
 - Cosmic_CompleteTargetedScreensMutant_vXXX_GRCh38.tsv
- COSMIC Genome Screen Mutants:
 - Cosmic_GenomeScreensMutant_vXXX_GRCh37.tsv
 - Cosmic_GenomeScreensMutant_vXXX_GRCh38.tsv
- COSMIC Census Genes Mutations
 - Cosmic_MutantCensus_vXXX_GRCh37.tsv
 - Cosmic_MutantCensus_vXXX_GRCh38.tsv
- COSMIC Resistance Mutations:

- Cosmic_ResistanceMutations_vXXX_GRCh37.tsv
- Cosmic_ResistanceMutations_vXXX_GRCh38.tsv

Variants in recent COSMIC tsv format files are 3'-shifted relative to the plus-strand of the reference. To compare variants detected using the *CLC Genomics Workbench* with COSMIC variants, it may be preferable to import COSMIC VCF files with variants 5'-shifted using the VCF importer. This is because variants detected using the *CLC Genomics Workbench*, in accordance with VCF recommendations. (See section 30.1.6.)

Please see chapter 1.1.6 for more information on how different formats (e.g. VCF and GVF) are interpreted during import in CLC format. For all of the above, zip files are also supported. Please note that for human data, there is a difference between the UCSC genome build and Ensembl/NCBI for the mitochondrial genome. This means that for the mitochondrial genome, data from UCSC should not be mixed with data from other sources (see <https://hgdownload.soe.ucsc.edu/goldenPath/hg19/bigZips/>). Most of the data above is annotation data and if the file includes information about allele variants (like VCF, Complete Genomics and GVF), it will be combined into one **variant** track that can be used for finding known variants in your experimental data.

For all types of files except fasta, you need to select a reference track as well. This is because most the annotation files do not contain enough information about chromosome names and lengths which are necessary to create the appropriate data structures.

7.2.1 GFF3 format

A GFF3 file contains a list of various types of annotations that can be linked together with "Parent" and "ID" tags.

Here are some example of a few common tags used by the format:

- **ID** IDs for each feature must be unique within the scope of the GFF file. In the case of discontinuous features (i.e., a single feature that exists over multiple genomic locations) the same ID may appear on multiple lines. All lines that share an ID collectively represent a single feature.
- **Parent** A parent ID can be used to group exons into transcripts, transcripts into genes, and so forth. A feature may have multiple parents. A parent ID can only be used to indicate a 'part of' relationship.
- **Name** The name that will be displayed as a label in the track view. Unlike IDs, there is no requirement that the Name be unique within the file.

Figure 7.3 exemplifies how tags are used to create annotations.

In the *CLC Genomics Workbench*, the GFF3 importer will create an output track for each feature type present in the file. In addition, the *CLC Genomics Workbench* will generate an (RNA) track that aggregates all the types that were "RNA" into one track (i.e., all the children of "mature_transcript", which is the parent of "mRNA", which is the parent of the "NSD_transcript"); and a (Gene) track that includes genes and Gene-like types annotations like ncRNA_gene, plastid_gene, and tRNA_gene. These "(RNA)" and "(Gene)" tracks are different

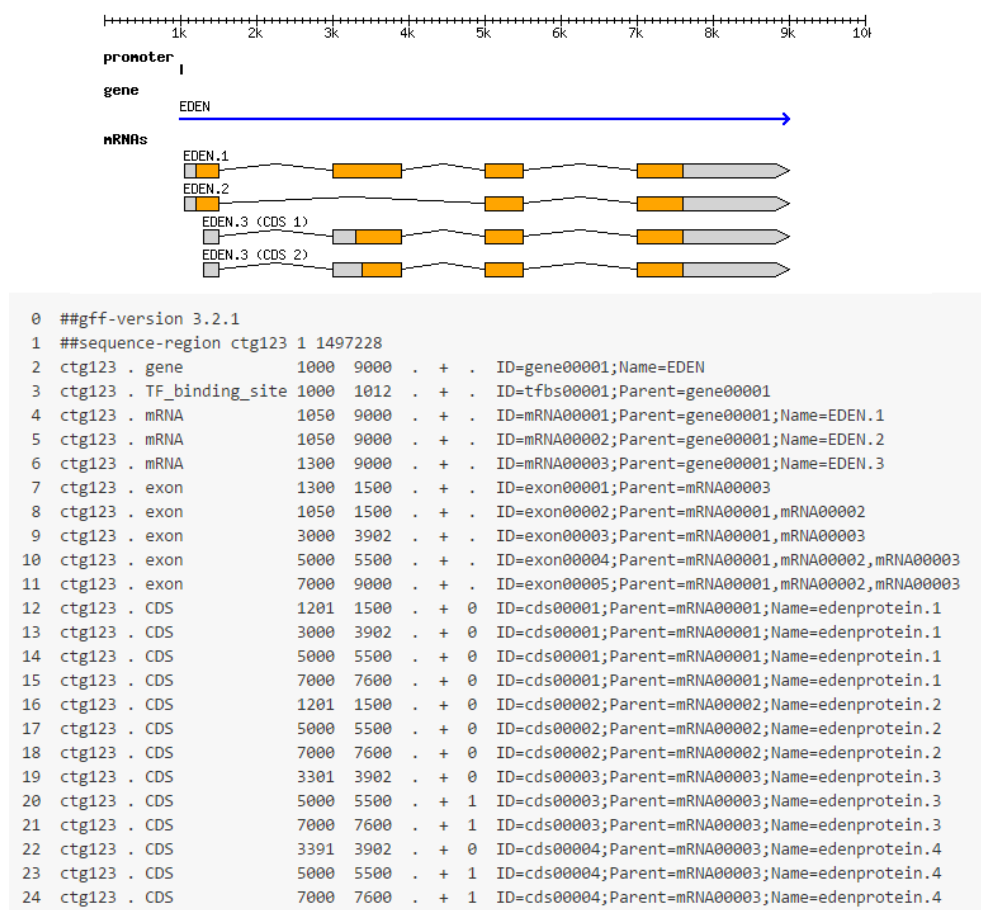


Figure 7.3: Example of a GFF3 file and the corresponding annotations from <https://github.com/The-Sequence-Ontology/Specifications/blob/master/gff3.md>.

from the ones ending with "_mRNA" and in "_Gene" in that they compile all relevant annotations in a single track, making them the track of choice for subsequent analysis (RNA-Seq for example).

- **Gene-like types.** These are types described in the Sequence Ontology as being subtypes of genes, e.g. ncRNA_gene, plastid_gene, tRNA_gene. Gene-like types are gathered together into an aggregated track with a name of the form "myFileName (Gene)". We recommend that users use this file in RNA-Seq.
- **Transcript-like types.** These are types described in the Sequence Ontology as being subtypes of transcripts that are neither primary transcripts (i.e., they do not require further processing to become functional), nor fusion transcripts. Again, there are several dozen, such as mRNA, lnc_RNA, threonyl_RNA. Transcript-like types are gathered together into an aggregated track with a name of the form "myFileName (RNA)". We recommend that users use this file in RNA-Seq.
- **Exons.** Where possible, exons are merged into their parent features. For example, the output of the lines shown in figure 7.4 will be a single mRNA feature with four exonic regions (from 1300 to 1500, 3000 to 3902, 5000 to 5500, and 7000 to 9000), and no exon features will be output on their own.

```

ctg123 . mRNA      1300 9000 . + . ID=mRNA00003;Parent=gene00001
ctg123 . exon      1300 1500 . + . Parent=mRNA00003
ctg123 . exon      3000 3902 . + . Parent=mRNA00003
ctg123 . exon      5000 5500 . + . Parent=mRNA00003
ctg123 . exon      7000 9000 . + . Parent=mRNA00003

```

Figure 7.4: Exons will be merged into their parent features when the parent is not a "gene-like" type.

In cases where the parent is of a "gene-like" type, exons are output as their own independent features in the exon track. Finding a lot of features in the exon track can suggest a problem with the file being imported. However, with large databases, this is more likely to be due to the database creators choosing to represent pseudogenes as exons with no transcript.

- **CDS** CDS regions with the same parent are joined together into a single spliced feature. If CDS features do not have a parent they are instead joined based on their ID, as for any other feature (described below)
- **Features with the same ID** Regardless of the feature type, features that have the same ID are merged into a single spliced feature. For example, the output of the following figure 7.5 will be a single cDNA_match feature with regions (1050..1500, 5000..5500, 7000..9000).

```

ctg123 . cDNA_match 1050 1500 5.8e-42 + . ID=match00001;Target=cdna0123 12 462
ctg123 . cDNA_match 5000 5500 8.1e-43 + . ID=match00001;Target=cdna0123 463 963
ctg123 . cDNA_match 7000 9000 1.4e-40 + . ID=match00001;Target=cdna0123 964 2964

```

Figure 7.5: Features that have the same ID are merged into a single spliced feature.

Naming of features

When one of the following qualifiers is present, it will be used for naming in the prioritized order:

1. the "Name" of the feature
2. the "Name" of the first named parent of the feature
3. the "ID" of the feature
4. the "ID" of the first parent
5. the type of the feature

Several examples of naming strategies are depicted in figure 7.6.

Merged CDS features have a slightly different naming scheme. First, if a CDS feature in the GFF3 file has more than one parent, we create one CDS feature in the *CLC Genomics Workbench* for each parent, and each is merged with all other CDS features from the GFF3 file that has the parent feature as parent as well. The naming is then done in the following prioritized order:

1. the "Name" of the feature, if all the constituent CDS features have the same "Name".
2. the "Name" of the first named parent of the feature, if it has a name.

1) The mRNA from the following will be called "EDEN-001"

```
ctg123 . gene      1000  9000  .  +  .  ID=gene00001;Name=EDEN
ctg123 . mRNA      1300  9000  .  +  .  ID=mRNA00003;Parent=gene00001,Name=EDEN-001
```

2) The mRNA from the following will be called "EDEN"

```
ctg123 . gene      1000  9000  .  +  .  ID=gene00001;Name=EDEN
ctg123 . mRNA      1300  9000  .  +  .  ID=mRNA00003;Parent=gene00001
```

3) The mRNA from the following will be called "mRNA00003"

```
ctg123 . gene      1000  9000  .  +  .  ID=gene00001
ctg123 . mRNA      1300  9000  .  +  .  ID=mRNA00003;Parent=gene00001
```

4) The mRNA from the following will be called "gene00001"

```
ctg123 . gene      1000  9000  .  +  .  ID=gene00001
ctg123 . mRNA      1300  9000  .  +  .  Parent=gene00001
```

5) The mRNA from the following mRNA without a parent gene will be called "mRNA"

```
ctg123 . mRNA      1300  9000  .  +  .
```

Figure 7.6: *Naming of features.*

3. the "Name" of the first of the merged CDS features with a name.
4. the "ID" of the first of the merged CDS features with an ID.
5. the "ID" of the parent.

For features with the same ID, the naming scheme is as follows:

1. the "Name" of the feature, if all have the same "Name".
2. If there is a set of common parents for the features and one of the common parents have a "Name", the name of the first common parent with a "Name" is used.
3. If at least one feature has a name, the name of the first feature with the name is used.
4. the "ID" of the first of the features

Limits of the GFF3 importer

Features are imported only if their SeqID (i.e., the value in the first column of the gff3) can be matched to the name of a chromosome in the genome. Matching need not be exact (see section [I.1.6](#)). However, in some cases it may be necessary to manually edit either the names of the genomic sequences (for example in a fasta file), or the SeqIDs in the GFF3 file so that they match. Features without a match aren't imported. You can see the number of skipped features in the importer log.

The start and stop position of a feature cannot extend beyond the ends of a chromosome, unless the chromosome is explicitly marked as circular, which is indicated by << and >> at the beginning and the end of the sequence.

Trying to import such a file will fail. One option is to delete the feature that extends beyond the end of the chromosome and to start the import again. Another option is to make the track circular. To do so, convert the linear track into a sequence using the Convert from Tracks tool. Open the sequence and right-click on its name to be able to choose the option "Make sequence

circular" from the drop-down menu. Convert the now circular sequence back into a track using the Convert to Tracks tool. Importing the gff3 should now be working.

The following instances are not supported:

- Interpreting SOFA accession numbers. The type of the feature is constrained to be either: (a) a term from the "lite" sequence ontology, SOFA; or (b) a SOFA accession number, distinguished using the syntax SO:000000. The importer recognizes terms from SOFA as well as terms from the full Sequence Ontology, but will not translate accession numbers to types. So for example, features with type SO:0000316 will not be interpreted as "CDS" but will be handled like any other type.
- The fasta directive ##FASTA. This FASTA section situated at the end of a GFF3 file specifies sequences of ESTs as well as of contigs. The GFF3 importer will ignore these sequences.
- Alignments. An aligned feature is handled as a single region, with the Gap and Target attributes added as annotations. We do not use Gap and Target to show how the feature aligns.
- Comment lines. We do not interpret lines beginning with a #. Especially relevant are lines "##sequence-region seqid start end" which some parsers use to perform bounds checking of features. Our bounds checking is instead performed against the user-supplied genome.

7.2.2 VCF import

Handling of the genotype (GT) field

The import process for VCF files into *CLC Genomics Workbench* currently works as follows:

1. In cases where GT = ./., no variants are imported at all.
2. In cases where GT = X/. or GT = ./X , and where X is not zero, a single variant is imported depending on the actual value of X.
3. In cases where GT = X/Y and X and Y are different but either one may be zero, two independent variants are created.

Note: The GT field is mandatory for import of sample variants (i.e., when FORMAT and sample columns are present).

Import of counts

To add variant count values to the imported variants, one of the following tags must be present in your VCF file: CLCAD2, AD, AO, or RO. Where more than one of these is present, they are prioritized in the following order:

1. CLCAD2
2. AD
3. AO and/or RO

Count values will be taken from the tag type with the highest priority, with values for other tags imported as annotations.

For example, if a VCF file has CLCAD2:AD for three possible variants with values 2,3,4:5,6,7, then the CLCAD2 values would be imported as counts, with each variant having a single count value (2,3,4 respectively), while the AD value for each variant would be included as an annotation (5,6,7 respectively).

Import of multiple samples and multiple VCF files

When importing a single VCF file, you will get a track for each sample contained in the VCF file.

In cases where information about more than one sample is present in the VCF file, you can choose to import the samples together into a single variant track, or import each sample into an individual variant track by checking the batch mode button in the lower left side of the wizard, as shown in figure 7.2. The difference between the two import modes is that the batch mode will import the samples individually in separate track files, whereas the non-batch mode will keep variants for one sample in one track, thus merging samples from the different input files (in cases where the same sample is contained in different input files).

If you select multiple VCF files, each containing multiple samples, then the non-batch mode will generate one track file for each unique sample. The batch mode will generate a track file for each of the original VCF files with the entire content, as if importing each of the VCF files one by one. For example, VCF file 1 contains sample 1 and sample 2, and VCF file 2 contains sample 2 and sample 3. When VCF file 1 and VCF file 2 are imported in non-batch mode, you will get three individual track files; one for each of the three samples 1, 2, and 3. If VCF file 1 and VCF file 2 were instead imported using the batch function, the result of the import would be four track files: a track from sample 1 from file 1, a track from sample 2 from file 1, a track from sample 2 from file 2, and a track from sample 3 from file 2.

Import of complex variants with reference overlap

Allelic variants that overlap but do not cover exactly the same range are called complex variants.

It can be specified that variants are represented using reference overlap by adding the line `###refOverlap=true` in the VCF header. If no such line is found in the header, the default is `"false"`, i.e., that no reference overlap alleles are present that need to be replaced by overlapping alleles.

Detection of complex regions: When reading a reference overlap VCF file, a complex region is initiated when overlapping alleles are called on different VCF lines. Complex regions can contain hundreds of complex variants, for example if one allele has a long deletion. Alleles overlap if they share a reference nucleotide position. Insertions overlap non-insertion if they are positioned internally, not if they are positioned at either boundary.

Replacing reference overlap alleles in complex regions: For each position with a complex alternate allele, a number of placeholder reference overlap alleles (refoPloidy) are expected to be present, so that the total number of alleles in the genotype field is equal to the ploidy at that position in the sample genome. For each such position in the complex region, it is then determined how many reference overlap alleles are replaced by overlapping

alternate and reference alleles (numReplaced). If any reference overlap alleles remain, they are assigned the allele depth: $\text{newAD} = \text{origAD} * (\text{refoPloidy} - \text{numReplaced}) / \text{refoPloidy}$, where origAD is the original allele depth for all reference overlap alleles at the position. In the "Reference overlap and depth estimate" example above (Table 2), the allele depth of the re-imported reference variant will be: $\text{newAD} = 6 * (2 - 1) / 2 = 3$. In the "Reference overlap" example above (Table 2), no reference overlap alleles will remain (numReplaced=2).

Alternative import of "Reference overlap" representation: The method above can be used for both "Reference overlap" and "Reference overlap with depth estimate" representations. However, a VCF file generated with the "Reference overlap" representation can also be imported correctly by simply importing as if it has no reference overlap, and subsequently removing all reference alleles with zero CLCAD2 allele depth.

Read more about complex variants with reference overlap in section [8.1.7](#).

Import variants represented as symbolic alleles

VCF Import supports the following symbolic alleles:

- - Deletions
- <INS> - Insertions
- <INV> - Inversions
- <DUP:TANDEM> - Tandem duplications

If possible, variants are imported to standard variant tracks. However, variants longer than 100,000 base pairs and variants that do not contain sufficient sequence information are imported to annotation tracks. Read about track types in section [27.1](#).

The following variants are imported to annotation tracks if represented as symbolic alleles in the VCF:

- Deletions longer than 100,000 base pairs
- All insertions
- All inversions
- Tandem duplications longer than 100,000 base pairs

Note that tandem duplications can also be represented as insertions, as described for the InDels output from InDels and Structural Variants in section [31.10.2](#).

Naming compatibility

Chromosome names in the VCF file must be identical to the chromosome names in the reference sequence specified during import, with the following exceptions:

- Inclusion of "chr" in chromosome names is optional. For example, VCF import understands both "1" and "chr1" to be chromosome 1.
- VCF import understands all the names for the alternative contigs in hg38 as they are listed here: <https://genome.ucsc.edu/cgi-bin/hgTracks?db=hg38&chromInfoPage=>

7.3 Import NGS Reads

CLC Genomics Workbench has dedicated importers for next-generation sequencing (NGS) reads, both short and long¹, described in the following sections.

Click on the **Import** (📁) button in the top toolbar to bring up a list of available importers, as shown in figure 7.7. Click on the importer to launch it.

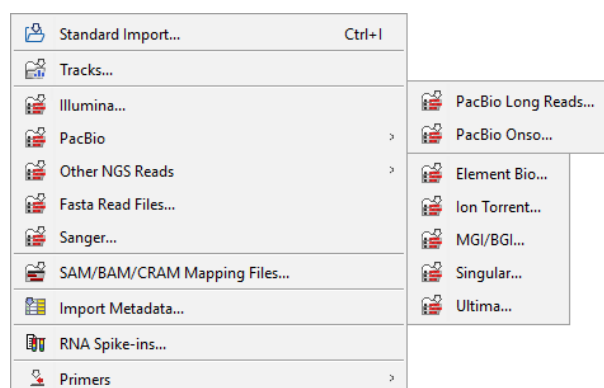


Figure 7.7: Importers for NGS short and long reads.

To specify the files to import, click:

- **Add folders** to choose one or several folders from which all the files should be imported.
- **Add files** to select individual files to import.

Files can be removed from the list by selecting them and clicking the **Remove** button.

The **Element Info** (📄) view of the imported element(s) shows and can be used to edit:

- The read group platform, which is determined by the importer, see figure 7.8.
- The paired status, see section 7.3.10.

Fastq importers can process UMI information from the fastq read headers, see section 7.3.11.

¹The *Long Read Support* plugin provides additional import functionality for long reads.

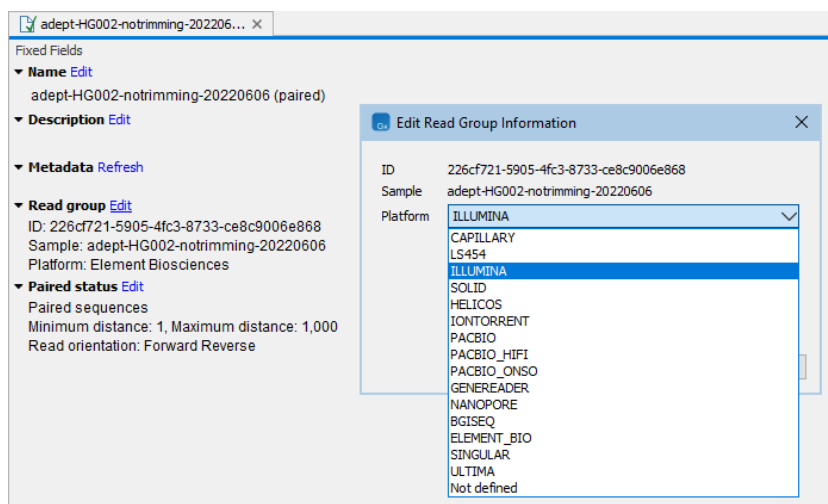


Figure 7.8: Editing the platform in the Element Info view.

7.3.1 Illumina

CLC Genomics Workbench supports data from Illumina's Genome Analyzer, HiSeq 2000, NextSeq and the MiSeq systems.

To launch the Illumina importer, go to:

Import (📁) | **Illumina** (🔍).

This opens a dialog where files can be selected and import options specified (figure 7.9).

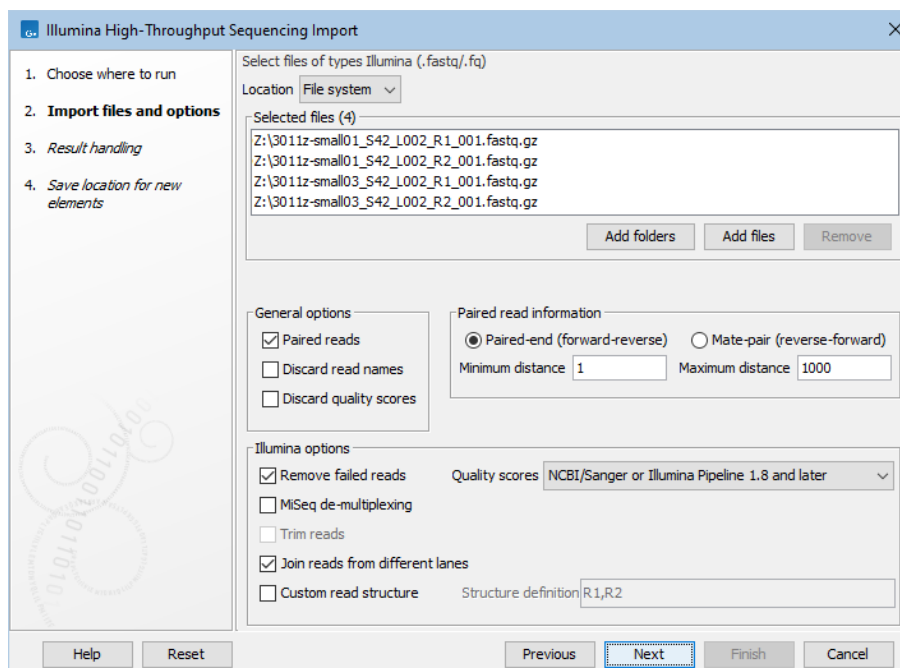


Figure 7.9: Importing data from Illumina systems.

Fastq (.fastq/.fq) files from Illumina can be imported. Uncompressed files as well as files compressed using gzip (.gz), zip (.zip) or bzip2 (.bz2) can be provided as input. The importer

processes UMI information from the fastq read headers, see section [7.3.11](#).

The drop down menu of input file locations includes the option **BaseSpace**. When selected, an **Access BaseSpace...** button is presented. Clicking this opens a browser window, where your Illumina BaseSpace credentials can be entered. After doing that, granting the *CLC Workbench* relevant access permissions and closing the browser window, you will be able to select files from BaseSpace in the Illumina High-Throughput Sequencing Import wizard. Your BaseSpace credentials remain valid for your current *CLC Workbench* session. BaseSpace configuration options are available in Preferences, see section [4.4](#).

The **General options** are:

- **Paired reads.** Files will be paired up based on their names, see **Default rules for determining pairs of files** below.

Under **Paired read information**:

- Choose the orientation of the paired reads, either **Forward-reverse** or **Reverse-forward**.
- Specify the insert sizes by setting **Minimum distance** and **Maximum distance**. Data sets with different insert sizes should be imported separately, with the correct minimum and maximum distance.

Read more about handling paired data in section [7.3.10](#).

- **Discard read names.** Read names can be discarded to save disk space without affecting analysis results. Keeping read names can be useful in some circumstances, such as when inspecting sequence list contents or when working downstream with subsets of sequences.
- **Discard quality scores.** Quality scores are visible in read mappings and are used by various tools, e.g. for variant detection. If quality scores are not relevant, use this option to discard them and reduce disk space and memory consumption.

Default rules for determining pairs of files First, the selected files are sorted based on the file names. Sorting is alphanumeric, except for files coming off the CASAVA1.8 pipeline, where pairs are organized according to their identifier and chunk number.

For example, for files from CASAVA1.8, files with base names like: ID_R1_001, ID_R1_002, ID_R2_001, ID_R2_002, the files would be sorted in the order below, where it is assumed that files with names containing "R1" contain the first sequences of the pairs, and those containing "R2" in the name contain the second sequence of the pairs.

1. ID_R1_001
2. ID_R2_001
3. ID_R1_002
4. ID_R2_002

In this example, the data in files ID_R1_001 and ID_R2_001 are treated as a pair, and ID_R1_002, ID_R2_002 are treated as a pair.

The file names are then used to check if each prospective file pair in this sorted list is valid. If the files in a pair seem to follow the following naming format:

`<sample name>_L<at least one digit>_[R1|R2]_<at least one digit>`,

then the files must contain the same sample name and lane information, in order to be valid.

If a prospective file pair does not follow this format, but the first file name does contain "_R1" and the second file name does contain "_R2", then the file pair is still considered valid. Note that if "_R1" or "_R2" occur more than once in a filename, the last occurrence in the name is used.

No data will be imported from file pairs that are not considered valid wrt. the above requirements. For such file pairs, a message will be printed in the log.

If the **Join reads from different lanes** option, in the Illumina options section of the dialog, is checked, then valid pairs of files with the same lane information in their file names will be imported into the same sequence list. If a valid pair of files do not contain the same lane information in their names, then no data is imported from those files and a message is printed in the log.

Within each file, the first read of a pair will have a 1 somewhere in the information line. In most cases, this will be a /1 at the end of the read name. In some cases though (e.g. CASAVA1.8), there will be a 1 elsewhere in the information line for each sequence. Similarly, the second read of a pair will have a 2 somewhere in the information line - either a /2 at the end of the read name, or a 2 elsewhere in the information line.

The organization of the files can be customized using the **Custom read structure** field, described in **Illumina options** below.

Illumina options

- **Remove failed reads.** Use this option to not import reads that did not pass a quality filter, as indicated within the fastq files.

Part of the header information for the quality score has a flag where Y means failed and N means passed. In this example, the read has not passed the quality filter:

```
@EAS139:136:FC706VJ:2:2104:15343:197393 1:Y:18:ATCACG
```

If you import paired data and one read in a pair is removed during import, the remaining mate will be saved in a separate sequence list with single reads.

- **MiSeq de-multiplexing.** Using this option on MiSeq multiplexed data will divide reads into different files based on the "IndexSequence" of the read header:

```
@Instrument:RunID:FlowCellID:Lane:Tile:X:Y:UMI ReadNum:FilterFlag:0:IndexSeq
```

- **Trim reads.** When checked, reads are trimmed when a B is encountered at either end of the reads in the input file. This option is only available when the "Quality score" option has been set to Illumina Pipeline 1.5 to 1.7 as a B in the quality score has a special meaning as a trim clipping in this pipeline. This trimming is carried out whether or not you choose to discard quality scores during import.

- **Join reads from different lanes.** When checked, fastq files from the same sequencing run but from different lanes are imported as a single sequence list.

Lane information is expected in the filenames as "_L<digits>", e.g. "L001" for lane 1. If this pattern occurs more than once in a filename, the last occurrence in the name is used. For example, for a filename "myFile_L001_L1.fastq" the lane information is assumed to be L1.

- **Quality scores.** There are four options for quality scores:

- NCBI/Sanger or Illumina 1.8 and later.
- Illumina Pipeline 1.2 and earlier.
- Illumina Pipeline 1.3 and 1.4.
- Illumina Pipeline 1.5 to 1.7.

See section 7.3.1 for details.

- **Custom read structure.** If the default organization of Illumina files for import does not match what is needed, you can check custom read structure and specify the desired organization in the structure definition field. Fastq files are specified by the read information in the name (e.g. R1, R2, I1, I2). When separated by a space, the specified reads for a given spot are concatenated on import. When comma separated, a paired sequence list is imported, with the first sequence in the pair made up of the read or reads listed before the comma, and the second sequence made up of the read or reads listed after the comma.

For example:

- If R2, R1 was entered, a paired sequence list would be imported. The first sequence of each pair would contain a read from the R2 fastq file, and its partner would contain the corresponding read from the R1 fastq file.
- If I1 R1 was entered, a sequence list containing single reads would be imported. Each read would contain sequence from the I1 fastq file prepended to sequence from the R1 fastq file.
- If R2 R1, R3 was entered, a paired sequence list would be imported. The first sequence of each pair would contain a read from the R2 fastq file prepended to the corresponding read from the R1 fastq file. The second sequence of each pair would contain the corresponding read from the R3 fastq file.

This could represent the situation where R1 contains forward reads, R3 has reverse reads, and R2 contains molecular indices.

In the next wizard step, options are presented for how to handle the results (see section 12.2). If you choose to **Save** the results, an option called "Create subfolders per batch unit" becomes available. When that option is checked, each sequence list is saved into a separate folder under the location selected to save results to. This can be useful for organizing subsequent analysis results and for running analyses in batch mode (see section 12.3).

Quality scores in the Illumina platform

When using the Illumina importer, you can select the quality score scheme applicable for your data. There are four options:

- **NCBI/Sanger or Illumina 1.8 and later.** Using a Phred scale encoded using ASCII 33 to 93. This is the standard for fastq formats except for the early Illumina data formats (this changed with version 1.8 of the Illumina Pipeline).
- **Illumina Pipeline 1.2 and earlier.** Using a Solexa/Illumina scale (-5 to 40) using ASCII 59 to 104. The Workbench automatically converts these quality scores to the Phred scale on import in order to ensure a common scale for analyses across data sets from different platforms (see details on the conversion next to the sample below).
- **Illumina Pipeline 1.3 and 1.4.** Using a Phred scale using ASCII 64 to 104.
- **Illumina Pipeline 1.5 to 1.7.** Using a Phred scale using ASCII 64 to 104. Values 0 (@) and 1 (A) are not used anymore. Value 2 (B) has special meaning and is used as a trim clipping. If this option is selected and the **Trim reads** option is checked, the reads are trimmed when a B is encountered at either end of the reads in the input file.

Further information about the fastq format, including quality score encoding, is available at https://en.wikipedia.org/wiki/FASTQ_format.

Small samples of three kinds of files are shown below. The names of the reads have no influence on the quality score format:

NCBI/Sanger Phred scores:

```
@SRR001926.1 FC00002:7:1:111:750 length=36
TTTTTGTAAAGAGGGGGGTCATCAAAATTTGCAAAA
+SRR001926.1 FC00002:7:1:111:750 length=36
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII'IB<IH
@SRR001926.7 FC00002:7:1:110:453 length=36
TTATATGGAGGCTTTAAGAGTCATAGGTTGTTCCCC
+SRR001926.7 FC00002:7:1:110:453 length=36
IIIIIIIIIIII:'III?=IIIIIII+&III/3I8F/&
```

Illumina Pipeline 1.2 and earlier (note the question mark at the end of line 4 - this is one of the values that are unique to the old Illumina pipeline format):

```
@SLXA-EAS1_89:1:1:672:654/1
GCTACGGAATAAAACCAGGAACAACAGACCCAGCA
+SLXA-EAS1_89:1:1:672:654/1
cccccccccccccccccccc]c``cVcZccbSYb?
@SLXA-EAS1_89:1:1:657:649/1
GCAGAAAATGGGAGTGAAAATCTCCGATGAGCAGC
+SLXA-EAS1_89:1:1:657:649/1
ccccccccccbccbccb``cccbcccZcc``^bR^`
```

The formulas used for converting the special Solexa-scale quality scores to Phred-scale:

$$Q_{phred} = -10 \log_{10} p$$

$$Q_{solexa} = -10 \log_{10} \frac{p}{1-p}$$

A sample of the quality scores of the Illumina Pipeline 1.3 and 1.4:

```
@HWI-E4_9_30WAF:1:1:8:178
GCCAGCGGCGCAAAATGNCGGCGGCGATGACCTTC
+HWI-E4_9_30WAF:1:1:8:178
babaaaa\ababaaaaREXabaaaaaaaaaaaaaa
@HWI-E4_9_30WAF:1:1:8:1689
GATGGAGATCTCGACCTNATAGGTGCCCTCATCGG
+HWI-E4_9_30WAF:1:1:8:1689
aab`']_aaaaaaaaaa[ER`abaaa\aaaaaaaaa[
```

Note that it is not possible to see from that data itself that it is actually not Illumina Pipeline 1.2 and earlier, since they use the same range of ASCII values.

To learn more about ASCII values, please see https://en.wikipedia.org/wiki/Ascii#ASCII_printable_characters.

7.3.2 Oxford Nanopore

The Oxford Nanopore importer supports import of reads generated by Oxford Nanopore sequencing technology.

To launch the Oxford Nanopore importer, go to:

Import (📁) | Oxford Nanopore (🏠).

This opens a dialog where files can be selected and import options specified (figure 7.10).

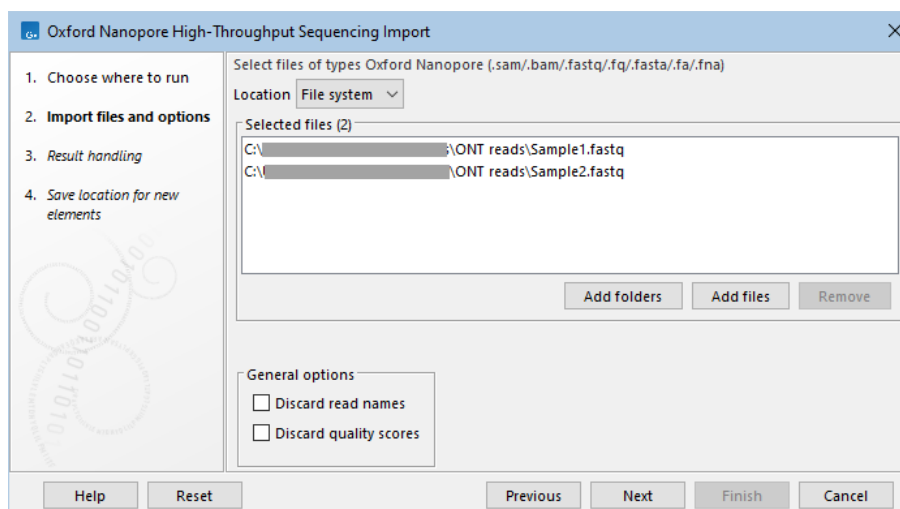


Figure 7.10: Importing data from Oxford Nanopore systems.

The following file formats are supported:

- Fastq (`.fastq/` `.fq`). Uncompressed files as well as files compressed using gzip (`.gz`), zip (`.zip`) or bzip2 (`.bz2`) can be provided as input. Quality scores are expected to be in the NCBI/Sanger format, see section 7.3.1. The importer processes UMI information from the fastq read headers, see section 7.3.11.
- Fasta files (`.fasta`) which contain sequence data. Compressed Fasta (`.fasta.gz`) files are also supported.

- SAM or BAM files (`.sam/.bam`) . Mapping information in the file is disregarded.

The **General options** are:

- **Discard read names.** Read names can be discarded to save disk space without affecting analysis results. Keeping read names can be useful in some circumstances, such as when inspecting sequence list contents or when working downstream with subsets of sequences.
- **Discard quality scores.** Quality scores are visible in read mappings and are used by various tools, e.g. for variant detection. If quality scores are not relevant, use this option to discard them and reduce disk space and memory consumption.

7.3.3 PacBio Long Reads

The PacBio Long Reads importer is designed to import long reads generated by PacBio sequencing technology.

To launch the PacBio Long Reads importer, go to:

Import (📁) | **PacBio** (🧬) | **PacBio Long Reads** (📄).

This opens a dialog where files can be selected and import options specified (figure 7.11).

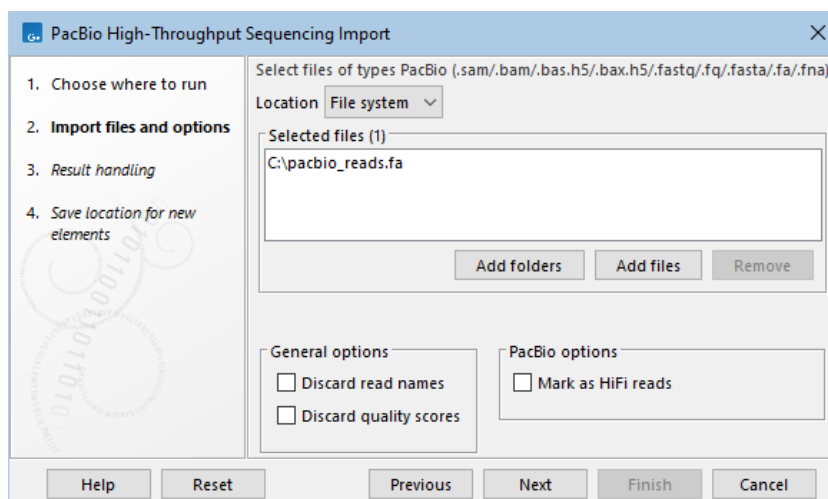


Figure 7.11: Importing data from PacBio.

The following file formats are supported:

- H5 files (`.bas.h5/.bax.h5`) which contain one of two things. `.bas.h5` files produced by instruments prior to PacBio RS II contain sequencing data such as reads and quality scores. `.bas.h5` files from more recent PacBio instruments contain a list of `.bax.h5` files where the actual sequencing data is stored. When importing H5 files, the user needs to select both the `.bas.h5` file and all the accompanying `.bax.h5` files belonging to a data set.
- Fastq (`.fastq/.fq`). Uncompressed files as well as files compressed using gzip (`.gz`), zip (`.zip`) or bzip2 (`.bz2`) can be provided as input. Quality scores are expected to be in the NCBI/Sanger format, see section 7.3.1. The importer processes UMI information from the fastq read headers, see section 7.3.11.

- Fasta files (`.fasta`) which contain sequence data. Compressed Fasta (`.fasta.gz`) files are also supported.
- SAM or BAM files (`.sam/.bam`). Mapping information in the file is disregarded.

Note that quality scores are capped at 64.

The **General options** are:


- **Discard read names.** Read names can be discarded to save disk space without affecting analysis results. Keeping read names can be useful in some circumstances, such as when inspecting sequence list contents or when working downstream with subsets of sequences.
- **Discard quality scores.** Quality scores are visible in read mappings and are used by various tools, e.g. for variant detection. If quality scores are not relevant, use this option to discard them and reduce disk space and memory consumption. As PacBio quality scores currently contain very little information, we recommend that you discard them. When importing Fasta files, this option is not available, since Fasta files do not contain quality scores.

The **PacBio options** are:

- **Mark as HiFi reads.** If checked, the reads will be recognized as PacBio HiFi sequencing reads instead of regular reads, enabling tools to apply HiFi specific settings when relevant. If importing SAM/BAM files where the Platform Model is set to HiFi (PM:HiFi), the reads will be imported as HiFi reads, regardless of whether **Mark as HiFi reads** is checked or not.

Mark imported PacBio reads as HiFi

To mark already imported PacBio reads as PacBio HiFi, change the **Read group** value:

1. Open the sequence list in the **View Area**.
2. Click the "Show Element Info" icon () at the bottom of the window.
3. Click **Edit** next to **Read group**.
4. In the **Platform** dropdown, change from PACBIO to PACBIO_HIFI.
5. Click **OK**.

7.3.4 PacBio Onso

The PacBio Onso importer is designed to import fastq (`.fastq/.fq`) files generated by PacBio Onso sequencing technology. Uncompressed files as well as files compressed using gzip (`.gz`), zip (`.zip`) or bzip2 (`.bz2`) can be provided as input. Quality scores are expected to be in the NCBI/Sanger format, see section 7.3.1. Note that quality scores are capped at 64. The importer processes UMI information from the fastq read headers, see section 7.3.11.

To launch the PacBio Onso importer, go to:

Import () | **PacBio** () | **PacBio Onso** ()

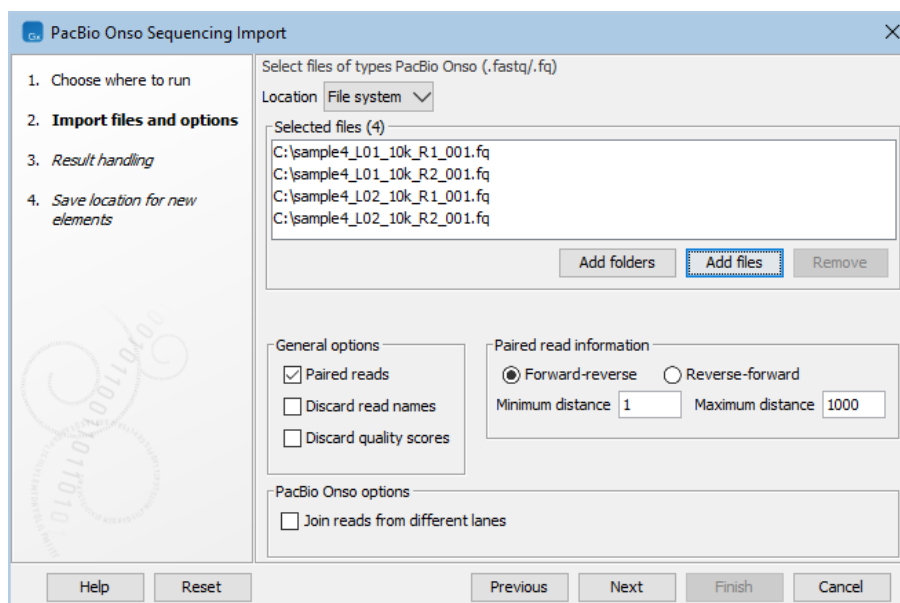


Figure 7.12: Importing data from PacBio Onso.

This opens a dialog where files can be selected and import options specified (figure 7.12).

The **General options** are:

- **Paired reads.** Files will be paired up based on their names, which are assumed to contain `_R1` and `_R2` (alternatively, `_1` and `_2`), respectively. Other than the `R1/R2` (or the `1/2`), the file names in a pair are expected to be identical.

Under **Paired read information**:

- Choose the orientation of the paired reads, either **Forward-reverse** or **Reverse-forward**.
- Specify the insert sizes by setting **Minimum distance** and **Maximum distance**. Data sets with different insert sizes should be imported separately, with the correct minimum and maximum distance.

Read more about handling paired data in section 7.3.10.

- **Discard read names.** Read names can be discarded to save disk space without affecting analysis results. Keeping read names can be useful in some circumstances, such as when inspecting sequence list contents or when working downstream with subsets of sequences.
- **Discard quality scores.** Quality scores are visible in read mappings and are used by various tools, e.g. for variant detection. If quality scores are not relevant, use this option to discard them and reduce disk space and memory consumption.

The **PacBio Onso options** are:

- **Join reads from different lanes.** When checked, fastq files from the same sequencing run but from different lanes are imported as a single sequence list.

Lane information is expected in the filenames as `"_L<digits>"`, e.g. `"L001"` for lane 1. If this pattern occurs more than once in a filename, the last occurrence in the name is used.

For example, for a filename "myFile_L001_L1.fastq" the lane information is assumed to be L1.

7.3.5 Element Biosciences

The Element Bio importer is designed to import fastq (.fastq/.fq) files generated by Element Biosciences sequencing technology. Uncompressed files as well as files compressed using gzip (.gz), zip (.zip) or bzip2 (.bz2) can be provided as input. Quality scores are expected to be in the NCBI/Sanger format, see section 7.3.1. The importer processes UMI information from the fastq read headers, see section 7.3.11.

To launch the Element Bio importer, go to:

Import (📁) | **Other NGS Reads** (📁) | **Element Bio** (📁).

This opens a dialog where files can be selected and import options specified (figure 7.13).

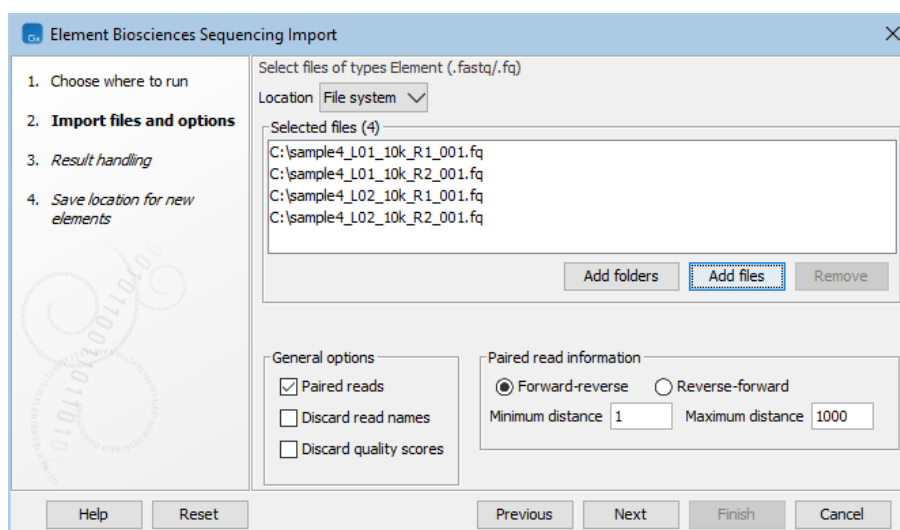


Figure 7.13: Importing data from Element Biosciences.

The **General options** are:

- **Paired reads.** Files will be paired up based on their names, which are assumed to contain _R1 and _R2 (alternatively, _1 and _2), respectively. Other than the R1/R2 (or the 1/2), the file names in a pair are expected to be identical.

Under **Paired read information**:

- Choose the orientation of the paired reads, either **Forward-reverse** or **Reverse-forward**.
- Specify the insert sizes by setting **Minimum distance** and **Maximum distance**. Data sets with different insert sizes should be imported separately, with the correct minimum and maximum distance.

Read more about handling paired data in section 7.3.10.

- **Discard read names.** Read names can be discarded to save disk space without affecting analysis results. Keeping read names can be useful in some circumstances, such as when inspecting sequence list contents or when working downstream with subsets of sequences.

- **Discard quality scores.** Quality scores are visible in read mappings and are used by various tools, e.g. for variant detection. If quality scores are not relevant, use this option to discard them and reduce disk space and memory consumption.

7.3.6 Ion Torrent

The Ion Torrent importer is designed to import files generated by Ion Torrent sequencing technology.

To launch the Ion Torrent importer, go to:

Import (📁) | **Other NGS Reads** (📁) | **Ion Torrent** (📁).

This opens a dialog where files can be selected and import options specified (figure 7.14).

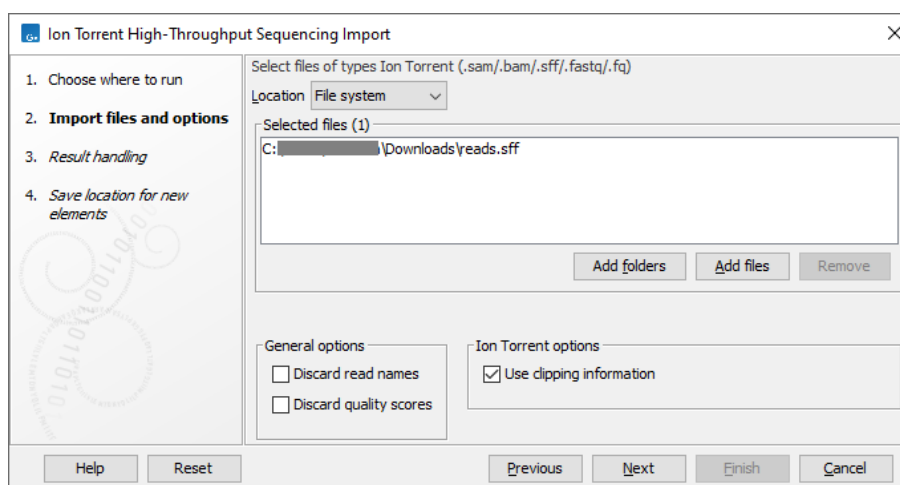


Figure 7.14: Importing data from Ion Torrent.

The following file formats are supported:

- SFF (.sff). Can provide extra information about adapter regions or regions of low quality.
- Fastq (.fastq/.fq). Uncompressed files as well as files compressed using gzip (.gz), zip (.zip) or bzip2 (.bz2) can be provided as input. Quality scores are expected to be in the NCBI/Sanger format, see section 7.3.1. The importer processes UMI information from the fastq read headers, see section 7.3.11.
- SAM or BAM (.sam/.bam). Mapping information in the file is disregarded.

The **General options** are:

- **Discard read names.** Read names can be discarded to save disk space without affecting analysis results. Keeping read names can be useful in some circumstances, such as when inspecting sequence list contents or when working downstream with subsets of sequences.
- **Discard quality scores.** Quality scores are visible in read mappings and are used by various tools, e.g. for variant detection. If quality scores are not relevant, use this option to discard them and reduce disk space and memory consumption.

The **Ion Torrent options** are:

- **Use clipping information.** Choose if clipping information in the sff format files should be used.

7.3.7 MGI/BGI

The MGI/BGI importer is designed to import fastq (.fastq/.fq) files generated by MGI/BGI sequencing technology. Uncompressed files as well as files compressed using gzip (.gz), zip (.zip) or bzip2 (.bz2) can be provided as input. Quality scores are expected to be in the NCBI/Sanger format, see section 7.3.1. The importer processes UMI information from the fastq read headers, see section 7.3.11.

To launch the MGI/BGI importer, go to:

Import (📁) | **Other NGS Reads** (📁) | **MGI/BGI** (📁).

This opens a dialog where files can be selected and import options specified (figure 7.15).

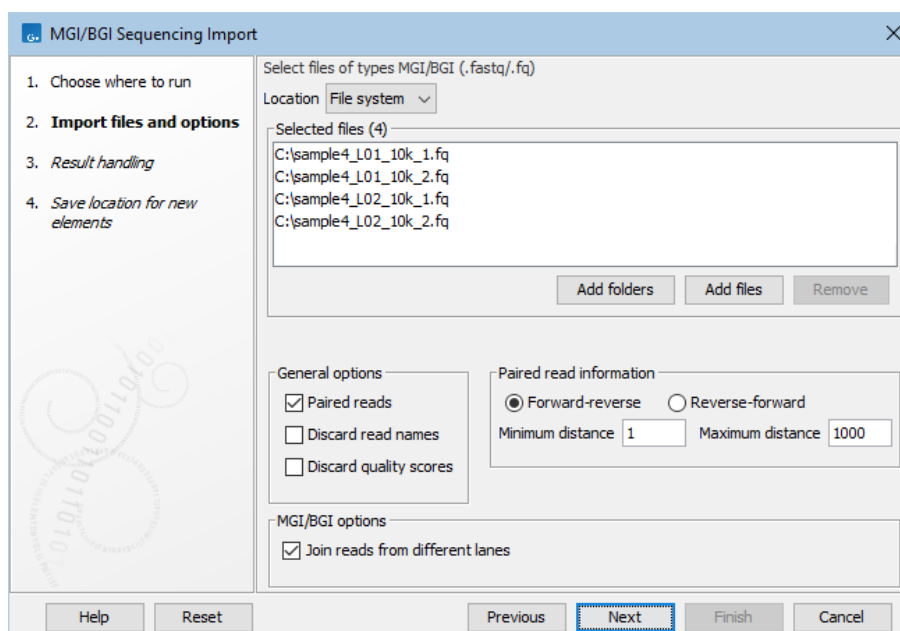


Figure 7.15: Importing data from MGI/BGI.

The **General options** are:

- **Paired reads.** Files will be paired up based on their names, which are assumed to contain _1 and _2 (alternatively, _R1 and _R2), respectively. Other than the 1/2 (or the R1/R2), the file names in a pair are expected to be identical. If such a file name format is not used, files will be paired up based on the names of their first read, using one of the following formats:
 - The read names end with /1 and /2, for example @sample1/1 and @sample1/2.
 - The read names contain a space followed by 1 or 2, for example @sample1 1:NNN and @sample1 2:NNN.

Under **Paired read information**:

- Choose the orientation of the paired reads, either **Forward-reverse** or **Reverse-forward**.
- Specify the insert sizes by setting **Minimum distance** and **Maximum distance**. Data sets with different insert sizes should be imported separately, with the correct minimum and maximum distance.

Read more about handling paired data in section 7.3.10.

- **Discard read names.** Read names can be discarded to save disk space without affecting analysis results. Keeping read names can be useful in some circumstances, such as when inspecting sequence list contents or when working downstream with subsets of sequences.
- **Discard quality scores.** Quality scores are visible in read mappings and are used by various tools, e.g. for variant detection. If quality scores are not relevant, use this option to discard them and reduce disk space and memory consumption.

The **MGI/BGI options** are:

- **Join reads from different lanes.** When checked, fastq files from the same sequencing run but from different lanes are imported as a single sequence list.

Lane information is expected in the filenames as "_L<digits>", e.g. "L001" for lane 1. If this pattern occurs more than once in a filename, the last occurrence in the name is used. For example, for a filename "myFile_L001_L1.fastq" the lane information is assumed to be L1.

7.3.8 Singular Genomics

The Singular importer is designed to import fastq (.fastq/.fq) files generated by Singular Genomics sequencing technology. Uncompressed files as well as files compressed using gzip (.gz), zip (.zip) or bzip2 (.bz2) can be provided as input. Quality scores are expected to be in the NCBI/Sanger format, see section 7.3.1. The importer processes UMI information from the fastq read headers, see section 7.3.11.

To launch the Singular importer, go to:

Import (📁) | **Other NGS Reads** (📁) | **Singular** (📁).

This opens a dialog where files can be selected and import options specified (figure 7.16).

The **General options** are:

- **Paired reads.** Files will be paired up based on their names, which are assumed to contain _R1 and _R2 (alternatively, _1 and _2), respectively. Other than the R1/R2 (or the 1/2), the file names in a pair are expected to be identical.

Under **Paired read information**:

- Choose the orientation of the paired reads, either **Forward-reverse** or **Reverse-forward**.
- Specify the insert sizes by setting **Minimum distance** and **Maximum distance**. Data sets with different insert sizes should be imported separately, with the correct minimum and maximum distance.

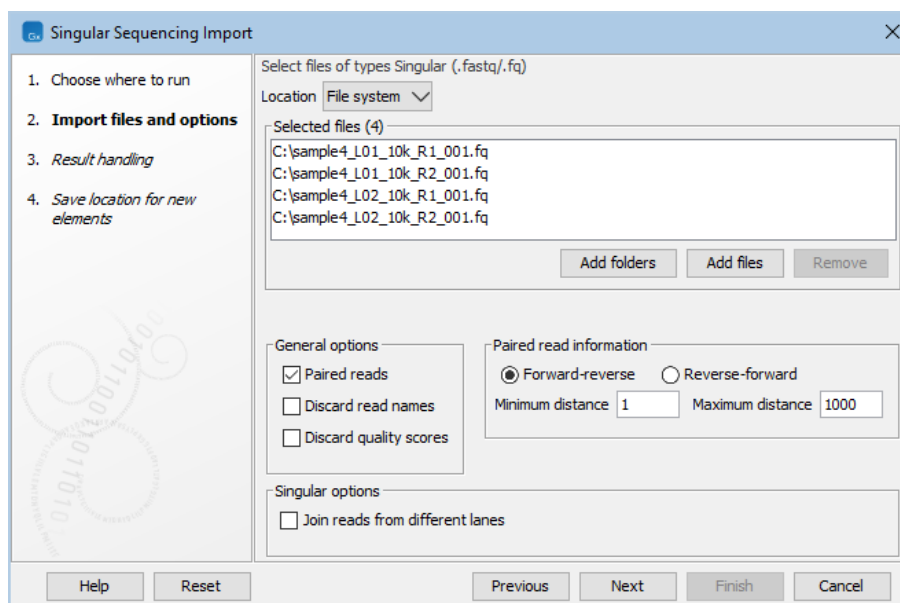


Figure 7.16: Importing data from Singular Genomics.

Read more about handling paired data in section 7.3.10.

- **Discard read names.** Read names can be discarded to save disk space without affecting analysis results. Keeping read names can be useful in some circumstances, such as when inspecting sequence list contents or when working downstream with subsets of sequences.
- **Discard quality scores.** Quality scores are visible in read mappings and are used by various tools, e.g. for variant detection. If quality scores are not relevant, use this option to discard them and reduce disk space and memory consumption.

The **Singular options** are:

- **Join reads from different lanes.** When checked, fastq files from the same sequencing run but from different lanes are imported as a single sequence list.
Lane information is expected in the filenames as "_L<digits>", e.g. "L001" for lane 1. If this pattern occurs more than once in a filename, the last occurrence in the name is used. For example, for a filename "myFile_L001_L1.fastq" the lane information is assumed to be L1.

7.3.9 Ultima Genomics

The Ultima importer can be used for importing CRAM files generated by Ultima sequencing technology.

To launch the Ultima importer, go to:

Import (📁) | **Other NGS Reads** (📁) | **Ultima** (📁).

This opens a dialog where files can be selected and import options specified (figure 7.17).

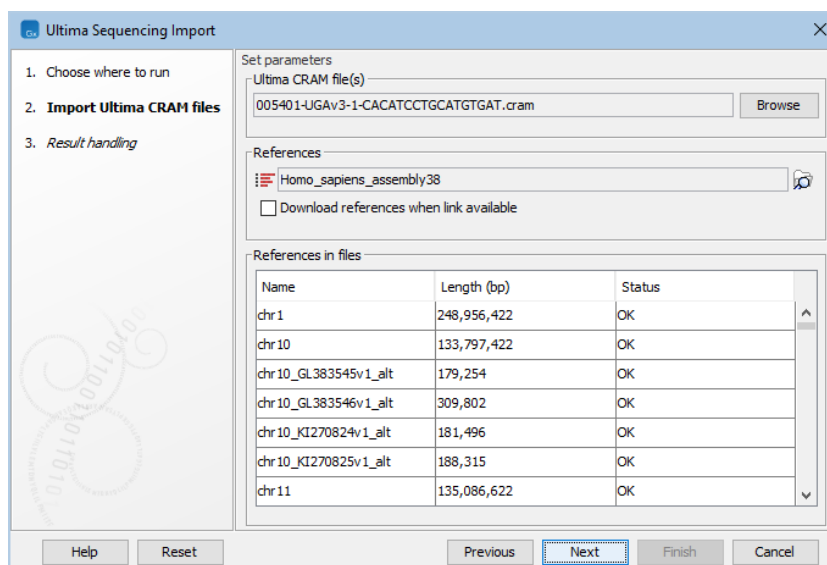


Figure 7.17: Choosing Ultima CRAM file and reference(s).

Providing references for the Ultima CRAM file

The reference sequence(s) that are referred to within the Ultima CRAM file must be specified in the 'Set parameters' wizard step (figure 7.17):

- If the Ultima CRAM file already contains information about where to find the reference(s), tick **Download references when link available**.
- If the input reference(s) are present in the *CLC Genomics Workbench*, click on the "Find in folder" icon (🔍) to select the reference(s).

Occurrences of disallowed characters according to the specification at <https://samtools.github.io/hts-specs/SAMv1.pdf> (whitespaces \, " ' @ () [] < >) in the input references are replaced by _ (underscore). Additionally, = and * are only disallowed at the beginning of the reference names. E.g., an input reference named *my=reference@sequence is considered the same as the reference _my=reference_sequence referred to within the Ultima CRAM file.

The table under 'References in files' contains the references that are referred to within the Ultima CRAM file, with their name, length, and a status. The status indicates whether a given reference referred to within the Ultima CRAM file is present in the input references. The status can be:

- **OK**. There is a reference in the input references with this name and length.
- **Length differs**. There is a reference in the input references with this name, but with a different length.
- **Download link available**. The Ultima CRAM file contains a URL for this reference. Tick **Download references when link available** to automatically download the reference.
- **Will download**. The Ultima CRAM file contains a URL for this reference and **Download references when link available** is already ticked. The reference is automatically downloaded.

- **Missing, download link not available.** There is no reference in the input references with this name, and there is no URL available in the Ultima CRAM file for downloading the reference.

A reference is 'matched' when the status is either **OK** or **Will download**. The import will fail if there are unmatched references.

For references located on a *CLC Genomics Server*, the table is empty. The importer can be launched, regardless of whether the correct references are selected, but it leads to an error in cases where they are not.

Output options

In the 'Result handling' wizard step, the downloaded reference sequences can be saved using the option **Save downloaded reference sequences** if the option **Download references when link available** was selected in the 'Set parameters' wizard step.

One sequence list per read group is created. Mapping information in the CRAM file is disregarded.

7.3.10 General notes on handling paired data

During import, information about paired data (orientation and distances) can be specified. This information is stored and can be taken into account during analyses.

The members of each read pair are placed sequentially in a sequence list. E.g. for paired-end data, the forward read in a pair is followed by the reverse read in that pair. When manipulating sequence lists, it is important not to interfere with this ordering. If read names were not discarded during import, the members of each read pair should have the same name, except for a 1 or a 2 somewhere in the name.

This information is used by tools such as Map Reads to Reference and RNA-Seq Analysis. Some tools have an "Auto-detect paired distances" option. When this option is checked, the distances set during import are disregarded, and are instead calculated from the data (see section 30.1.4).

The orientation and distance values for a sequence list can be edited by opening the **Element Info** (📄) view of the sequence list (see section 15.4). If the "Paired sequences" box is unchecked, the sequences will be handled as single (non-paired) reads.

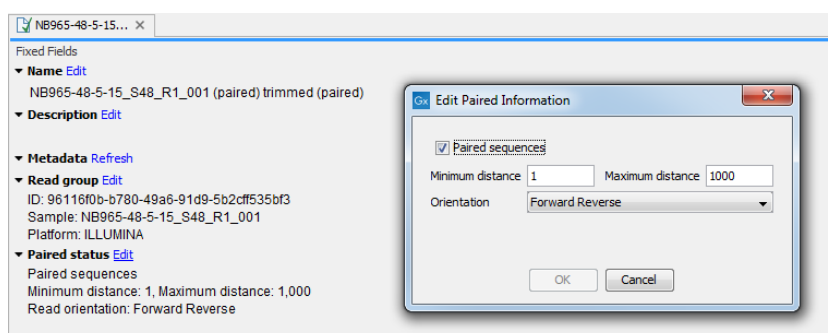


Figure 7.18: Editing paired orientation and distance in the Element Info view.

A paired read distance includes the full read sequence. E.g. for paired-end reads, the full

read sequence is from the beginning of the forward read to the beginning of the reverse read (figure 7.19).

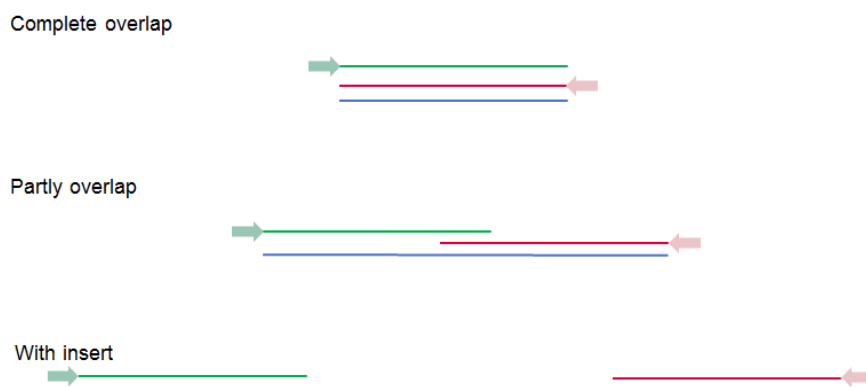


Figure 7.19: Green lines represent forward reads, red lines reverse reads, and in blue is shown the distance of the sequenced DNA fragment. Thus, if there is a complete overlap, the minimum distance will not be 0, but the length of the overlap.

7.3.11 General notes on UMIs

Some NGS library preparation protocols use Unique Molecular Indexes (UMIs) to improve performance by, for example

- identifying and correcting sequencing errors to allow higher sensitivity in variant calling
- eliminating library amplification and sequencing biases in RNA quantification

UMIs are usually located on the reads. The UMIs on the imported reads can be processed by tools delivered by the *Biomedical Genomics Analysis* plugin.


Various platforms offer the option to remove the UMIs and the information is instead added to the read headers in the fastq file. UMIs are extracted from read headers during import if the header of the first read in the file contains UMI information in one of the following formats:

- `<sample info>:<UMI> <read number>:<more sample info>`
- `<sample info>_<UMI> <read number>:<more sample info>`
- `<sample info>:<UMI>/<read number>`

The read header must contain exactly one space or slash, located between the `<UMI>` and `<read number>`. The imported sequences are annotated with the `<UMI>`. The allowed characters in the `<UMI>` are A, C, G, T and N. For paired reads, the `<UMI>` may contain one + (plus sign), separating the UMIs for each read in the pair, in which case the reads are annotated with the concatenated UMIs, i.e. the `<UMI>` without the +.

7.4 Import other high-throughput sequencing data

CLC Genomics Workbench has dedicated importers for high-throughput sequencing data formats, described in the following sections.

Click on the **Import** () button in the top toolbar to bring up a list of available importers, as shown in figure 7.20. Click on the importer to launch it.

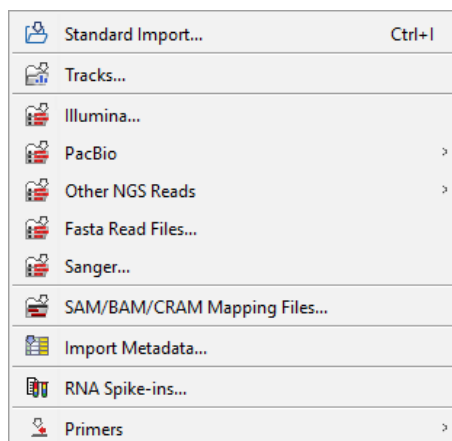


Figure 7.20: Importers for NGS short and long reads.

To specify the files to import, click:

- **Add folders** to choose one or several folders from which all the files should be imported.
- **Add files** to select individual files to import.

Files can be removed from the list by selecting them and clicking the **Remove** button.



7.4.1 Fasta read files

The Fasta Read Files importer is designed to import large volumes of read data such as high-throughput sequencing data (NGS reads) in fasta format. Uncompressed files as well as files compressed using gzip (.gz), zip (.zip) or bzip2 (.bz2) can be provided as input. When using the importer, read names can be included but descriptions from the fasta files are ignored.

To launch the Fasta Read Files importer, go to:

Import () | **Fasta Read Files** (.

This opens a dialog where files can be selected and import options specified (figure 7.21).

For import of other fasta format data, such as reference sequences, please use the Standard Import () instead (see section 7.1), as this import format also includes the descriptions. To have a reference in track format, please use **Tracks** () and set the "Type of files to import" to FASTA.

This data type can also be imported using the on-the-fly import functionality available in workflows, described in section 14.3.

The **General options** to the left are:

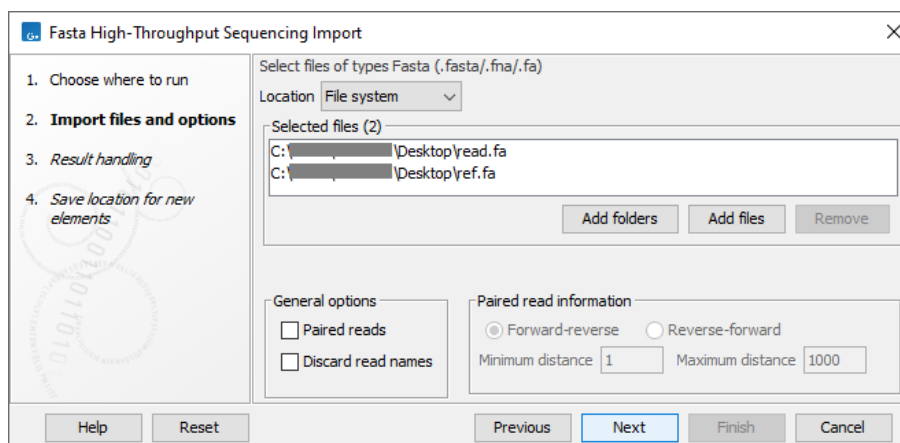


Figure 7.21: Importing data in fasta format.

- **Paired reads.** For paired import, the *CLC Genomics Workbench* expects the forward reads to be in one file and the reverse reads in another. The *CLC Genomics Workbench* sorts the files before import and then assumes that the first and second file belong together, and that the third and fourth file belong together etc. At the bottom of the dialog, you can choose whether the ordering of the files is **Forward-reverse** or **Reverse-forward**. As an example, you could have a data set with two files: `sample1_fwd` containing all the forward reads and `sample1_rev` containing all the reverse reads. In each file, the reads have to match each other, so that the first read in the `fwd` list should be paired with the first read in the `rev` list. Note that you can specify the insert sizes when importing paired read data. If you have data sets with different insert sizes, you should import each data set individually in order to be able to specify different insert sizes. Read more about handling paired data in section 7.3.10.
- **Discard read names.** Selecting this option saves disk space. Names of individual sequences are often irrelevant in large datasets.

Click **Next** to adjust how to handle the results (see section 12.2). We recommend choosing **Save** in order to save the results directly to a folder, since you probably want to save anyway before proceeding with your analysis. There is an option to put the import data into a separate folder. This can be useful for better organizing subsequent analysis results and for batching (see section 12.3).

7.4.2 Sanger sequencing data

The Sanger importer is designed to import large volumes of Sanger high-throughput sequencing data. Formats supported are `ab`, `abi`, `ab1`, `scf` and `phd`. Uncompressed files as well as files compressed using `gzip` (`.gz`), `zip` (`.zip`) or `bzip2` (`.bz2`) can be provided as input.

To launch the Sanger importer, go to:

Import (📁) | **Sanger** (🇬🇧).

This opens a dialog where files can be selected and import options specified (figure 7.22).

Sanger sequencing data can also be imported using the Standard Import (📁) (see section 7.1).

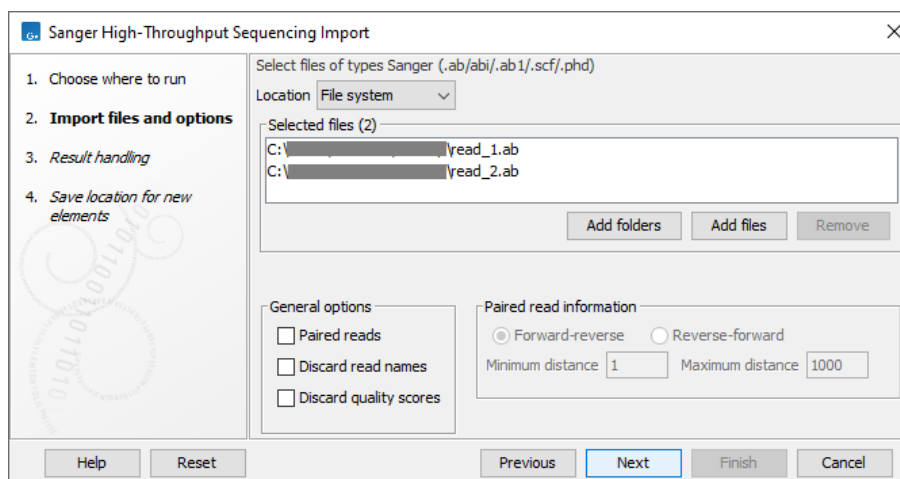


Figure 7.22: Importing Sanger sequencing data.

The following are key differences of the high-throughput importer when compared to the Standard Import:

- It is designed to handle large volumes of data efficiently.
- A given batch of sequences is imported to a single sequence list. The Standard Import creates a single sequence element for each imported sequence.
- The chromatogram traces are removed (quality scores remain). This improves performance; trace data takes up a lot of disk space, and this can impact speed and memory consumption of downstream analyses.
- Paired reads are supported.

Sanger data can also be imported using the on-the-fly import functionality available in workflows, described in section 14.3. Both the Sanger importer and the Standard Import ("Trace files") are available using the on-the-fly import.

The **General options** to the left are:

- **Paired reads** Import pairs of reads into a single sequence list. When enabled, the files selected for import are sorted, and then the first and second file are imported together as paired reads, the third and fourth file are imported together as paired reads, etc. The selection of "Forward-reverse" or "Reverse-forward" in the "Paired read information" area determines whether the first file is treated as containing forward reads and the second file reverse reads, or vice versa. As an example, with two files: `sample1_fwd` containing forward reads and `sample1_rev` containing reverse reads, and selecting the "Forward-reverse" option, you would get a single sequence list, marked as containing paired reads, with the pairs in the expected orientation. Insert sizes can also be specified, using the "Minimum distance" and "Maximum distance" settings. Data sets with different insert sizes should be imported separately. Read more about handling paired data in section 7.3.10.

- **Discard read names** Selecting this option saves disk space. Names of individual sequences are often irrelevant in large datasets.
- **Discard quality scores** Selecting this option can save substantial space, and can decrease memory consumption for downstream activities. Quality scores should be retained if they are relevant to your work. For example, quality scores are used for variant detection and can (optionally) be seen displayed in views of read mappings.

The next wizard step provides some options for handling the results, see section 12.2. When the option to "Create subfolders per batch unit" is enabled, each sequence list created is put into its own subfolder. This can be helpful for running analyses in batches (see section 12.3) and for organizing the results of subsequent analyses.

7.4.3 SAM, BAM and CRAM mapping files

The *CLC Genomics Workbench* supports import of files in the following alignment map formats for storing large nucleotide sequence alignments:

- SAM (Sequence Alignment/Map)
- BAM (Binary Alignment/Map)
- CRAM (Compressed Reference-oriented Alignment/Map)

See <https://samtools.github.io/hts-specs/> for specifications for these formats.

Alignments from a SAM/BAM/CRAM file can be imported as a reads track or as stand-alone read mappings. To import the reads in a SAM/BAM file as a sequence list, disregarding any alignment information, please use the Standard Import instead (see section 7.1).

Note that importing SAM/BAM/CRAM files can be very memory-consuming for large data sets, and particularly those with many reads marked as members of broken pairs in the mapping.


To launch the SAM/BAM/CRAM importer, go to:

Import  | **SAM/BAM/CRAM Mapping Files** .

This opens a dialog where files can be selected and import options specified (figure 7.23).

Providing references for the SAM/BAM/CRAM file

The reference sequence(s) that are referred to within the SAM/BAM/CRAM file must be specified in the 'Set parameters' wizard step (figure 7.23):

- If the SAM/BAM/CRAM file already contains information about where to find the reference(s), tick **Download references when link available**.
- If the input reference(s) are present in the *CLC Genomics Workbench*, click on the "Find in folder" icon  to select the reference(s).

Occurrences of disallowed characters according to the specification at <https://samtools.github.io/hts-specs/SAMv1.pdf> (whitespaces \, " ' @ () [] < >) in the input references are replaced by _ (underscore). Additionally, = and * are only disallowed at the beginning of the reference names. E.g., an input reference named *my=reference@sequence

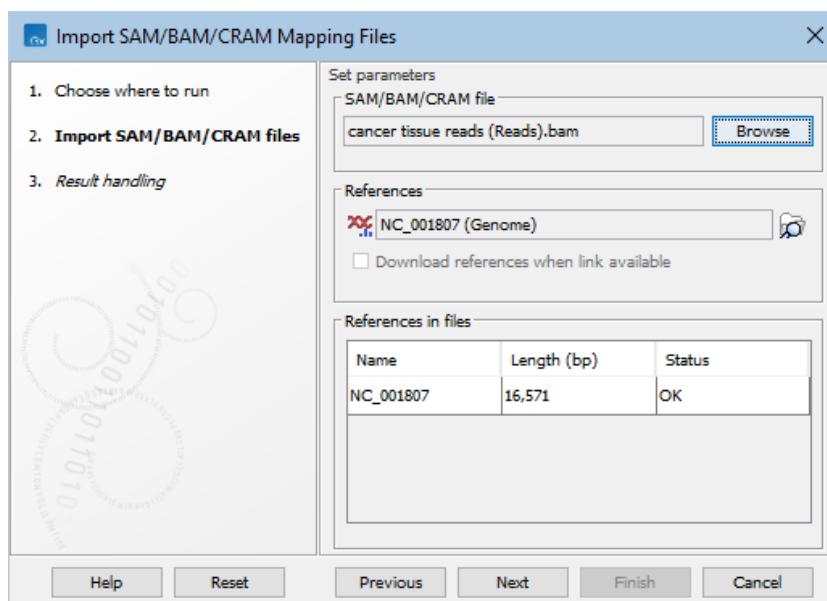


Figure 7.23: Choosing SAM/BAM/CRAM file and reference(s).

is considered the same as the reference `_my=reference_sequence` referred to within the SAM/BAM/CRAM file.

When importing SAM or BAM files, a range of different case-insensitive synonyms are used when matching input reference names with those specified in the file, e.g.:

- Chromosome 1: 1, chr1, chromosome_1, nc_000001
- Chromosome M: m, mt, chrm, chrmt, chromosome_m, chromosome_mt, nc_001807
- Chromosome Y: y, chry, chromosome_y, nc_000024

Note that synonyms are not used when importing CRAM files.

The table under 'References in files' contains the references that are referred to within the SAM/BAM/CRAM file, with their name, length, and a status. The status indicates whether a given reference referred to within the SAM/BAM/CRAM file is present in the input references. The status can be:

- **OK.** There is a reference in the input references with this name and length.
- **Length differs.** There is a reference in the input references with this name, but with a different length.
- **Download link available.** The SAM/BAM/CRAM file contains a URL for this reference. Tick **Download references when link available** to automatically download the reference.
- **Will download.** The SAM/BAM/CRAM file contains a URL for this reference and **Download references when link available** is already ticked. The reference is automatically downloaded.
- **Missing, download link not available.** There is no reference in the input references with this name, and there is no URL available in the SAM/BAM/CRAM file for downloading the reference.

A reference is 'matched' when the status is either **OK** or **Will download**. Only reads mapping to a matched reference are imported from SAM and BAM files. Import of CRAM files fails when there are unmatched references.

For references located on a *CLC Genomics Server*, the table is empty. The importer can be launched, regardless of whether the correct references are selected, but it leads to an error in cases where they are not.

Output options

In the 'Result handling' wizard step, the output options for the importer can be configured (figure 7.24):

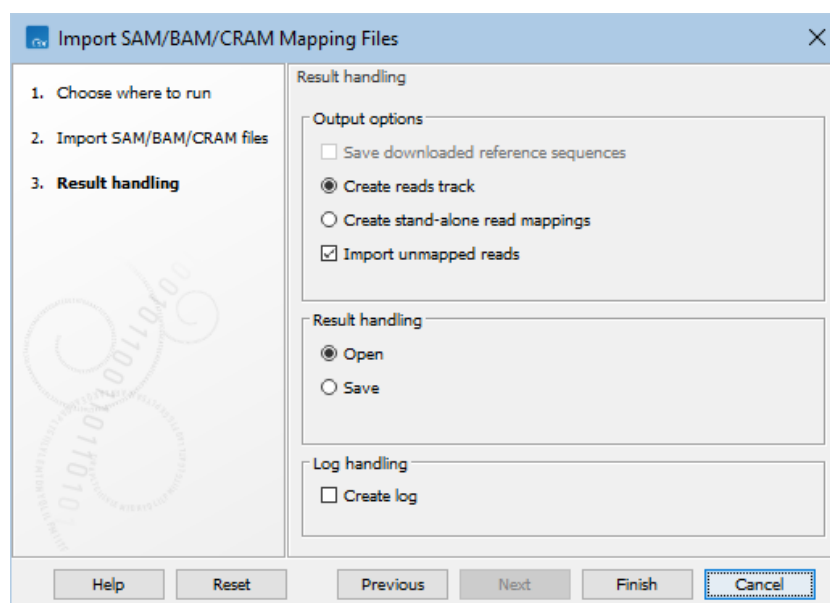


Figure 7.24: Result handling.

- **Save downloaded reference sequences.** This option is enabled if the option **Download references when link available** is selected in the 'Set parameters' wizard step.
- **Create reads track.** Mapped reads are imported as a reads track.
- **Create stand-alone read mappings.** Mapped reads are imported as stand-alone read mappings. When there is only one reference, the result is a single read mapping (📄), otherwise the result is a multi-mapping element (📄📄).
- **Import unmapped reads.** Unmapped reads are imported into sequence lists. One sequence list is created per read group.
 - If unmapped reads are part of an intact pair, they are imported into a sequence list of paired data.
 - If unmapped reads are single reads or a member of a pair that did not map while its mate did, they are imported into a sequence list containing single reads.

Only reads mapping to a matched reference are imported from SAM and BAM files. Import of CRAM files fails when there are unmatched references.

For files containing multiple alignment records for a single read, only the primary alignment (see <https://samtools.github.io/hts-specs/SAMv1.pdf>) is imported.

7.5 Import RNA spike-in controls

The *CLC Genomics Workbench* has a dedicated tool for importing RNA spike-in control data: **Import | RNA Spike-ins**

The wizard offers the option to import a standard ERCC file as provided by Thermo Fisher Scientific, or a custom made one (figure 7.25).

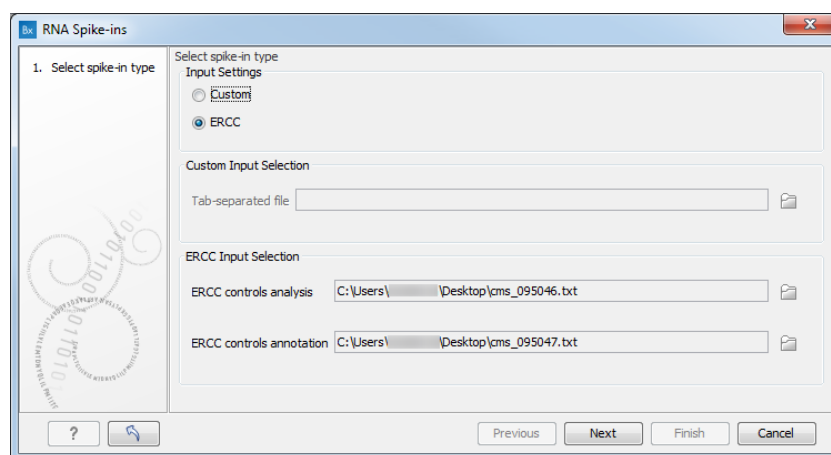


Figure 7.25: The RNA spike-in controls importer.

To import a standard ERCC file, look for **ERCC Controls Analysis** and **ERCC Control Annotation** files on the Thermo Fisher Scientific website, download both *.txt files on your computer, and start the importer. Select the option "ERCC" and specify the location of the analysis and annotation files in the relevant "ERCC Input Selection" fields at the bottom of the wizard.

For **custom-made spike-in controls**, choose the "Custom" option and specify in the "Custom Input Selection" field a tab-separated file (*.tsv or *.txt) containing the spike-in data organized as such: sequence name in the first column, nucleotide sequence in the second column, followed by as many columns as necessary to contain the concentrations of the spike-in measures in attomoles/microliters. Concentrations must not contain commas: write 15000 instead of 15,000. Remove any white space and save the table as a tab-separated TSV or TXT file on your computer.

It is also possible to import Lexogen Spike-in RNA Variant Control Mixes by modifying the SIRV files to fit the custom file requirements. Download the **SIRV sequence design overview (XLSX)** from the Lexogen website and open it in Excel. In the annotation column, "c" designate the data that should be imported ("i" is under-annotated while "O" is over-annotated). Filter the table to only keep the rows having a 1 in the "c" column, then keep only - and in that order - the sequence name, nucleotide sequence and concentration columns of the remaining rows. Reformat the values to numerical values in attomoles/microliters before saving the table as a *.tsv file. Import the file in the workbench using the "Custom" option.

Once a spike-in file is specified, click **Next** and choose to **Save** the file in the Navigation Area for

later use in RNA-Seq Analysis.

7.6 Import Primer Pairs

The **Import Primer Pairs** importer can import descriptions of primer locations from a generic text format file or from a QIAGEN gene panel primer file.

To run the tool, go to:

Import (📁) | **Primers** (🔍) | **Import Primer Pairs** (🔍)

This will open the wizard shown in figure 7.26. The first step is to select the data to import.

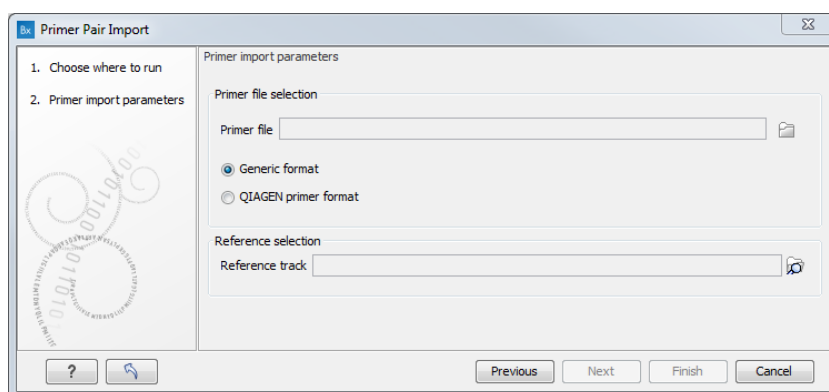


Figure 7.26: Select files to import.

- **Primer File** Click on the folder icon in the right side to select your primer pair location file. There are two primer pair formats that can be imported by the Workbench.

- **Generic Format** Select this option for primer location files with the exception of QIAGEN gene panel primers. Provide your primer location information in a tab delimited text file with the following columns:

- * Column 1: reference name
- * Column 2: primer1 first position (5'end) on reference
- * Column 3: primer1 last position (3'end) on reference
- * Column 4: primer2 first position (5'end) on reference
- * Column 5: primer2 last position (3'end) on reference
- * Column 6: amplicon name

Note: Primer position intervals are left-open and right-closed, so the leftmost position of the primer on the reference (column 2 and 5) should have one subtracted.

An example of the format expected for each row is:

```
chr1    42    65    142    106    Amplicon1
```

Indicating forward and reverse primers covering the reference nucleotides [43, 65] and [107, 142].

- **QIAGEN Primer Format** Use this option for importing information about QIAGEN gene panel primers.
- **Reference Track** Use folder icon in the right side to select the relevant reference track.

Click **Next** to go to the wizard step choose to save the imported primer location file.

Chapter 8

Exporting data and graphics

Contents

8.1	Data export	159
8.1.1	Export formats	159
8.1.2	Export parameters	161
8.1.3	Specifying the exported file name(s)	162
8.1.4	Export of folders and data elements in CLC format	164
8.1.5	Export of dependent elements	165
8.1.6	Export of tables	165
8.1.7	Export in VCF format	167
8.1.8	GFF3 export	172
8.1.9	BED export	173
8.1.10	JSON export	173
8.1.11	Graphics export	176
8.1.12	Export history	178
8.2	Export graphics to files	180
8.2.1	File formats	182
8.3	Export graph data points to a file	184
8.4	Copy/paste view output	185

Data and graphics can be exported from the *CLC Genomics Workbench* using export tools and workflows that contain export elements. Some types of data can also be exported using options available in right-click menus (see section 8.3) and others can be copy/pasted from within the *CLC Genomics Workbench* to other applications (see section 8.4).

Where data can be exported to

Data can be exported to any location accessible via the system the *CLC Genomics Workbench* is installed on, or to AWS S3 if an AWS Connection is configured that allows this (see section 6.2).

If connected to a *CLC Server*, data can be exported to areas accessible via the *CLC Server* (see section 6.1.1).

8.1 Data export

Data can be exported to many standard formats. The full list of supported formats is presented when launching the Export tool. Information about supported formats is also provided in the appendix section [I.1](#).

Launch the standard export functionality by clicking on the Export button on the toolbar, or selecting the menu option:

File | Export 

An additional export tool is available from under the File menu:

File | Export with Dependent Elements

This tool is described further in section [8.1.5](#).

The general steps when configuring a standard export job are:

- (Optional) Select data elements or folders to export in the **Navigation Area**.
- Launch the Export tool by clicking on the Export button in the Workbench toolbar or by selecting **Export** under the File menu.
- Select the format to export the data to.
- Select the data elements to export, or confirm elements that had been pre-selected in the **Navigation Area**.
- Configure the export parameters, including whether to output to a single file, whether to compress the outputs and how the output files should be named. Other format-specific options may also be provided.
- Select where the data should be exported to.
- Click **Finish**.

8.1.1 Export formats

Finding and selecting a format to export to When the Export tool is launched, a list of the available data formats is presented (figure [8.1](#)).

You can quickly find a particular format by typing a relevant search term into the text box at the top of the Export window, as shown in figure [8.2](#). Any formats with that search term in their name or description will be listed in the window. The search term is remembered when the Export tool is next launched. Delete the text from the search box if you wish to have all export formats listed.

Support for choosing an appropriate export format is provided in 2 ways:

- If data elements are selected in the **Navigation Area** before launching the Export tool, then a "Yes" or a "No" in the **Supported formats** column specifies whether or not the selected data elements can be exported to that format. If you have selected multiple data elements of different types, then formats that some, but not all, selected data elements can be exported to are indicated by the text "For some elements".

By default, supported formats appear at the top of the list.

- If no data elements are selected in the **Navigation Area** when the Export tool is launched, then the list of export formats is provided, but each row will have a "Yes" in the **Supported format** column. After an export format has been selected, only the data elements that can be exported to that format will be listed for selection in the next step of the export process.

Only zip format is supported when a folder, rather than data elements, is selected for export. In this case, all the elements in the folder are exported in CLC format, and a zip file containing these is created. See section 8.1.4.

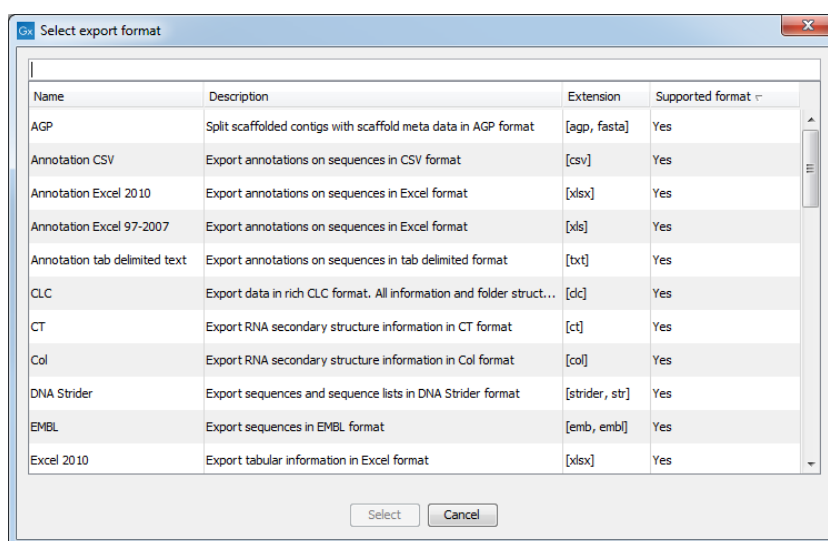


Figure 8.1: The Select export format dialog. Here, some sequence lists had been selected in the Navigation Area before the Export tool was launched. The formats that the selected data elements can be exported to contain a "Yes" in the Selected format column. Other export formats are listed below the supported ones, with "No" in the Supported format column.

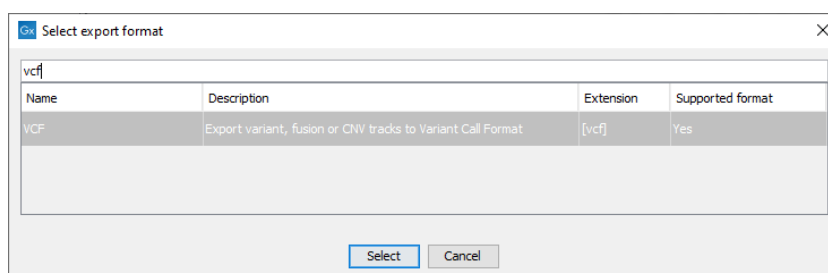


Figure 8.2: The text field has been used to search for the term "VCF" in the export format name or description field in the Select export dialog.

When the desired export format has been selected, click on the button labeled **Select**.

A dialog then appears, with a name reflecting the format you have chosen. For example if the VCF format was selected, the window is labeled "Export VCF".

If you are logged into a CLC Server, you will be asked whether to run the export job using the

Workbench or the Server. After this, you are provided with the opportunity to select or de-select data to be exported.

Selecting data for export In figure 8.3 we show the selection of a variant track for export to VCF format.

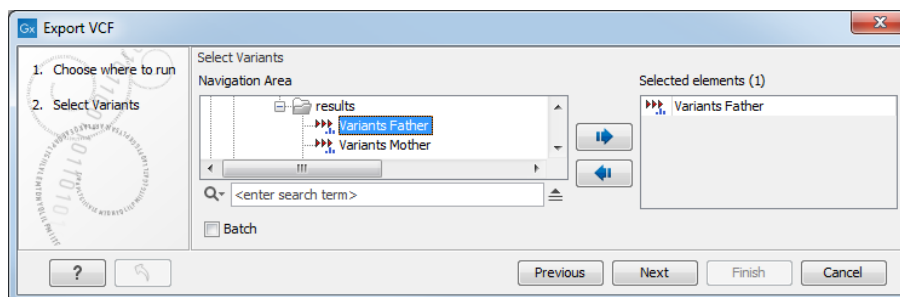


Figure 8.3: The Select export dialog. Select the data element(s) to export.

Further information is available about exporting the following types of information:

- Tables: section 8.1.6
- Variants to VCF format: section 8.1.7
- Reports to JSON format: section 8.1.10
- Graphics to a range of formats: section 8.1.11
- Data element history: section 8.1.12

8.1.2 Export parameters

The settings in the areas **Basic export parameters** and **File name** are offered when exporting to any format.

There may also be additional parameters for particular export formats. This is illustrated for the CLC exporter in figure 8.4.

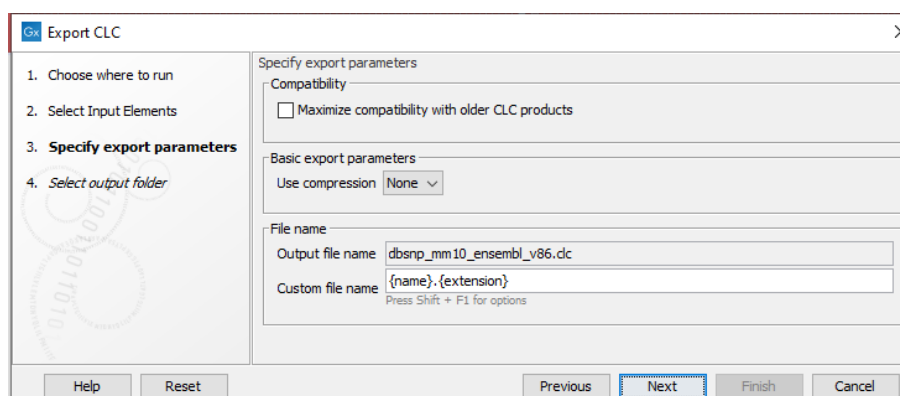


Figure 8.4: Configure the export parameters. When exporting to CLC format, you can choose to maximize compatibility with older CLC products.

Examples of configuration options:

- **Maximize compatibility with older CLC products** This is described in section 8.1.4.
- **Compression options** Within the **Basic export parameters** section, you can choose to compress the exported files. The options are no compression (None), gzip or zip format. Choosing zip format results in all data files being compressed into a single file. Choosing gzip compresses the exported file for each data element individually.
- **Paired reads settings** In the case of Fastq Export, the option "Export paired sequence lists to two files" is selected by default: it will export paired-end reads to two fastq files rather than a single interleaved file.
- **Exporting multiple files** If you have selected multiple files of the same type, you can choose to export them in one single file (only for certain file formats) by selecting "Output as single file" in the **Basic export parameters** section. If you wish to keep the files separate after export, make sure this box is not ticked. **Note:** Exporting in zip format will export only one zipped file, but the files will be separated again when unzipped.

The name to give exported files is also configured here. This is described in detail in section 8.1.3.

In the final wizard step, you select the location to save the exported files to.

8.1.3 Specifying the exported file name(s)

The names to give exported files can be configured in the export wizard. Names can be specified directly or placeholders can be used. Placeholders specify particular types of information, and thus are a convenient way to apply a consistent naming pattern to many exports.

The default is to use the placeholders `{name}` and `{extension}`, as shown in figure 8.5. Using these, the original data element name is used as the basename of the exported file, and the file format is used as the suffix. The actual filename that would result is shown in the **Output file name** field for the first element being exported.

When deciding on an output name, you can choose any combination of the different placeholders, standard text characters and punctuation, as in `{name}({day}-{month}-{year})`. As you add or remove text and terms in the **Custom file name** field, the text in the **Output file name** field will change so you can see what the result of your naming choice will be for your data.

An example where specific text instead of a placeholder might be preferred would be if the extension used for a particular format is not as desired. For example, the extension used for fasta files is `.fa`. To use `.fasta` instead, replace `{extension}` with `".fasta"` in the **Custom file name** field, as shown in figure 8.6.

When exporting a single file, the desired filename can just be typed in the **Custom file name** field. This should not be done when exporting to more than one file, as this would result in every exported file having an identical name.

The following placeholders are available:

- **{name}** or **{1}** - default name of the data element being exported
- **{extension}** - default extension for the chosen export format
- **{counter}** - a number that is incremented per file exported. i.e. If you export more than one file, counter is replaced with 1 for the first file, 2 for the next and so on.

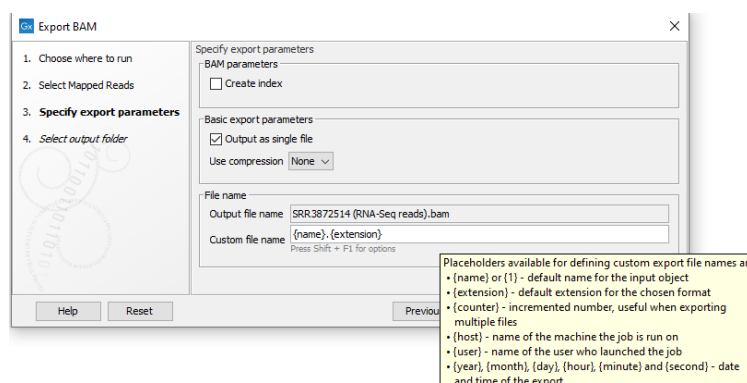


Figure 8.5: The default placeholders, separate by a "." are being used here. The tooltip for the Custom file name field provides information about these and other available placeholders.

- **{host}** - name of the machine the job is run on
- **{user}** - name of the user who launched the job
- **{year}**, **{month}**, **{day}**, **{hour}**, **{minute}**, and **{second}** - timestamp information based on the time an output is created. Using these placeholders, items generated by a tool at different times can have different filenames.

Note: Placeholders available for Workflow Export elements are different and are described in section 14.2.4.

Exported files can be saved into subfolders by using a forward slash character / at the start of the custom file name definition. When defining subfolders, all later forward slash characters in the configuration, except the last one, are interpreted as further levels of subfolders. For example, a name like /outputseqs/level2/myoutput.fa would put a file called myoutput.fa into a folder called level2 within a folder called outputseqs, which would be placed within the output folder selected in the final wizard step when launching the export tool. If the folders specified in the configuration do not already exist, they are created. Folder names can also be specified using placeholders.

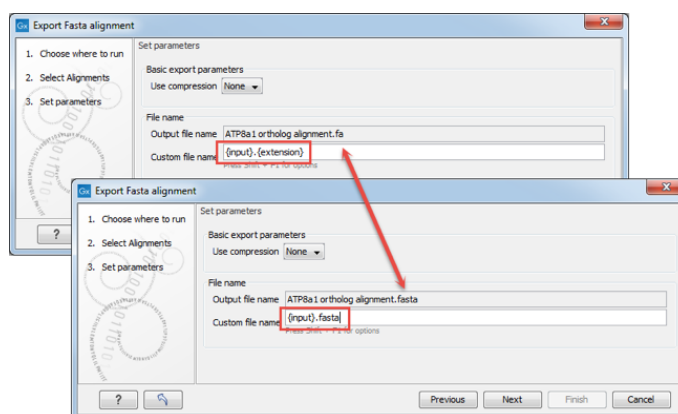


Figure 8.6: The file name extension can be changed by typing in the preferred file name format.

8.1.4 Export of folders and data elements in CLC format

The *CLC Genomics Workbench* stores data in CLC format. A CLC format file holds all the information for a given data element. This means the data itself, as well as information about that data, like history information.

Data can be exported in CLC format by selecting the CLC format, or the zip format, from the list of available formats.

If CLC format is chosen, each selected data element can be exported to an individual file. An option is offered later in the export process to apply gzip or zip compression. Choosing gzip compression at this stage will compress each data element individually. Choosing zip produces a single file containing the individual CLC format files. If a single zip file containing one or more CLC format files is the desired outcome, choosing the zip format in the first step of the export process specifies this directly.

If a folder is selected for export, only the zip format is supported. In this case, each data element in that folder will be exported to CLC format, and all these files will be compressed in a single zip file.

CLC format files, or zip files containing CLC format data, can be imported directly into a workbench using the Standard Import tool and selecting "Automatic import" in the Options area.

Backing up and sharing data

If you are backing up data, or plan to share data with colleagues who have a CLC Workbench, exporting to CLC format is usually the best choice. All information associated with that data element will then be available when the data is imported again. CLC format is also recommended when sharing data with the QIAGEN Bioinformatics Support team.

If you are planning to share your data with someone who does not have access to a licensed *CLC Genomics Workbench* but just wishes to view the data, then you can export to CLC format, and they can run the *CLC Genomics Workbench* in Viewing Mode, which does not require a license, see Viewing mode [1.4.8](#). In Viewing Mode, CLC format data can be imported and viewed in the same way it would be using a licensed Workbench.

Compatibility of the CLC data format between Workbench versions

When exporting to CLC or zip format, the option "Maximize compatibility with older CLC products" is presented at the **Specify export parameters** step. With that option checked, data is exported without internal compression applied. This is only relevant when sharing data with someone using a version of the CLC software where the compression applied by default is not supported.

Compatibility information:

- A new compression method was introduced with version 22.0 of the CLC Genomics Workbench, CLC Main Workbench and CLC Genomics Server. Compressed data created using those versions can be read by version 21.0.5 and above, but not earlier versions.
- Internal compression of CLC data was introduced in CLC Genomics Workbench 12.0, CLC Main Workbench 8.1 and CLC Genomics Server 11.0. Compressed data created using these versions is not compatible with older versions of the software. Data created using

these versions can be opened by later versions of the software, including versions 22.0 and above.

Information on how to turn off internal data compression entirely is provided in section 4.4. We generally recommend, however, that data compression remains enabled.

8.1.5 Export of dependent elements

Sometimes it can be useful to export the results of an analysis and its dependent elements. That is, the results along with the data that was used in the analysis. For example, one might wish to export an alignment along with all the sequences that were used in generating that alignment.

To export a data element with its dependent elements:

- Select the parent data element (like an alignment) in the **Navigation Area**.
- Start up the exporter tool by going to **File | Export with Dependent Elements**.
- Edit the output name if desired and select where the resulting zip format file should be exported to.

The file you export contains compressed CLC format files containing the data element you chose and all its dependent data elements.

A zip file created this way can be imported directly into a CLC workbench by going to

File | Import (📎) | Standard Import

and selecting "Automatic import" in the Options area.

Compatibility of the CLC data format between Workbench versions Internal compression of CLC data was introduced in CLC Genomics Workbench 12.0, CLC Main Workbench 8.1 and CLC Genomics Server 11.0. If you are sharing data for use in software versions older than these, then please use the standard Export functionality, selecting all the data elements, or folders of elements, to export and choosing either CLC or zip format as the export format. Further information about this is provided in section 8.1.4.

8.1.6 Export of tables

Data in tables can be exported to CSV, tab-delimited, Excel, or HTML format files.

The "Export all columns" option is selected by default. When it is deselected, options for selecting the columns to export are presented in the next wizard step.

When selecting specific columns for export, the option "Export the table as currently shown" is particularly useful if you have filtered, sorted, or selected particular columns in a table that is open in a View. In this case, the effects of these manipulations are preserved in the exported file. This option is not available for all data types.

When the "Export the table as currently shown" is unchecked or disabled, checkboxes for each column to be exported are available to select or deselect. The buttons below that section can help speed up the process of column selection:

- **All** Select all possible columns.
- **None** Clear the existing selection.
- **Default** Select a standard set of columns, as defined by the software for this data type.
- **Last export** Select the columns that were selected during the most recent previous export.
- **Active View** Select the same set of columns as those selected in the Side Panel of the open data element. This button is only visible if the element being exported is in an open View.

In the final wizard step, select the location where the exported elements should be saved.

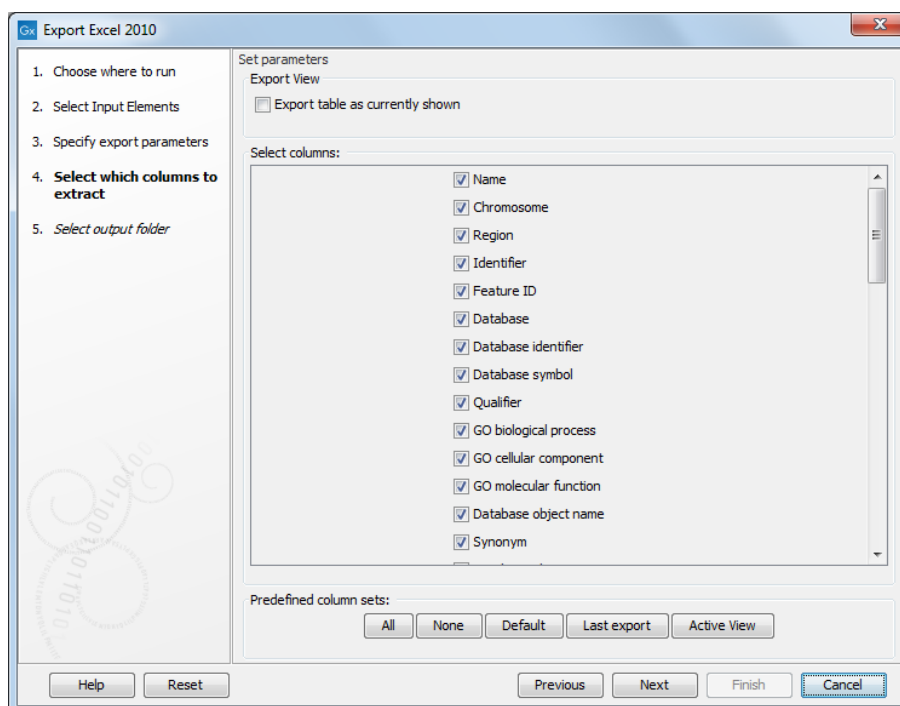


Figure 8.7: Selecting table columns to be exported.

The data exported will reflect any filtering and sorting applied.

Considerations when exporting tables

- **Row limits** Excel limits the number of hyperlinks in a worksheet to 66,530. When exporting a table of more than 66,530 rows, Excel will "repair" the file by removing all hyperlinks. If you want to keep the hyperlinks valid, you will need to subset your data and then export it to several worksheets, where each would have fewer than 66,530 rows.
- **Decimal places** When exporting to CSV, tab-separated, or Excel formats, numbers with many decimals are exported with 10 decimal places, or in scientific notation (e.g. 1.123E-5) when the number is close to zero.

When exporting a table in HTML format, data are exported with the number of decimals that have been defined in the *CLC Genomics Workbench* preference settings. When tables are exported in HTML format from a *CLC Server* the default number of decimal places is 3.

- **Decimal notation** When exporting to CSV and tab delimited files, decimal numbers are formatted according to the Locale setting of the *CLC Genomics Workbench* (see General preferences 4.1. If you open the CSV or tab delimited file with software like Excel, that software and the *CLC Workbench* should be configured with the same Locale.

8.1.7 Export in VCF format

Using this tool, variants, CNV and fusion data are exported to a VCF 4.2 format file.

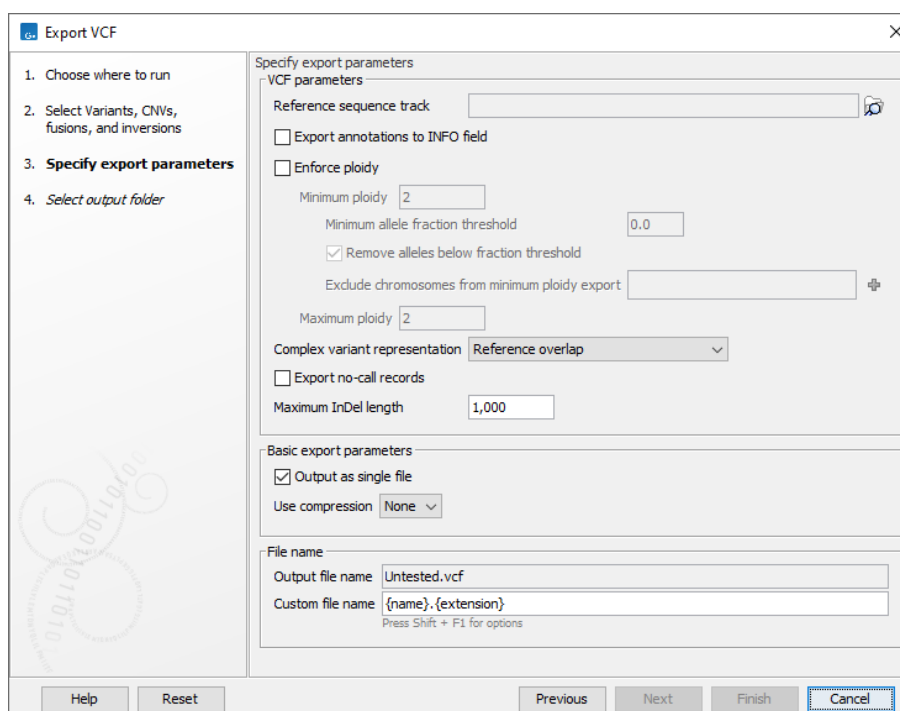


Figure 8.8: Several options are available when exporting to a VCF format file.

A number of configuration options are available (figure 8.8). Those specific to exporting to a VCF format file are:

Reference sequence track Since the VCF format specifies that reference and allele sequences cannot be empty, deletions and insertions have to be padded with bases from the reference sequence. The export needs access to the reference sequence track in order to find the neighboring bases.

Export annotations to INFO field Checking this option will export annotations on variant alleles as individual entries in the INFO field. Each annotation gets its own INFO ID. Various annotation tools can be found under Resequencing Analysis | Variant Annotation. Undesired annotations can be removed prior to export using Remove Information from Track, see section 27.5.5. Some variant annotations corresponding to database identifiers, such as dbSNP and db_xref, will also be exported in the ID field of the VCF data line.

Enforce ploidy Enforce minimum and maximum ploidy by modifying the number of alleles in the exported VCF genotype (GT) field. The two steps "Enforce minimum ploidy" and "Enforce maximum ploidy" are carried out separately during export in the mentioned order. Note that "Enforce minimum ploidy" can be disabled by setting both Minimum ploidy and Minimum allele fraction threshold to zero. "Enforce maximum ploidy" can be disabled by setting Maximum ploidy to 1000 or more.

- **Minimum and Maximum ploidy.** Minimum and maximum number of alleles to be written in the genotype field of the VCF. Enforcing minimum and maximum ploidy only affects the VCF genotype field. Both are set by default to 2, resulting in a VCF file in which the allele values in the Genotype (GT) field for haploid variants are reported following the format for diploid variants (i.e., the GT allele values reported could be 1/1). This is to allow compatibility of the exported VCF file with programs for downstream variant analysis that expect strictly diploid genomes. Note that it is proper to enforce diploid if the sample is diploid, and two alleles are expected to be present at all positions in the variant track (except excluded chromosomes). But if the variants have been filtered in a way that positions are no longer expected to have two alleles (e.g. all reference alleles have been removed), then it becomes wrong to enforce diploid.
- **Minimum allele fraction threshold and Remove alleles below fraction threshold.** Only alleles with an allele fraction above this threshold are considered as contributing to the minimum ploidy alleles. Alleles with a fraction below the threshold may still be reported in the VCF genotype field if the "Remove alleles below fraction threshold" option is disabled and the maximum ploidy allows it. The effect of this threshold depends on the minimum and maximum ploidy values set: For a minimum ploidy set at 2, a maximum ploidy set at 4 and the "Remove alleles below fraction threshold" option disabled, a case of 3 alleles where one (A) is above the threshold and two (C and T) are below will lead to the VCF genotype A/A/C/T. If the "Remove alleles below fraction threshold" option is enabled, or the maximum ploidy is set to 2, the VCF genotype field becomes A/A.
- **Exclude chromosomes from minimum ploidy export.** The user can specify that the Enforce minimum ploidy option is only applied to certain chromosomes, while others will be reported without enforcing a minimum ploidy.
Some chromosomes can be excepted from the enforced diploid export. For a human genome, that would be relevant for the mitochondrion and for male X and Y chromosomes. For this option, you can select which chromosomes should be excepted. They will be exported in the standard way without assuming there should be two genotypes, and homozygous calls will just have one value in the GT field.

Complex variant representation Complex variants are allelic variants that overlap but do not cover the same range. In exporting, a VCF line will be written for each complex variant. Choose from the drop down menu:

- Reference overlap: Accurate representation where reference alleles are added to the genotype field to specify complex overlapping alleles.
- Reference overlap and depth estimate: More widely compatible and less accurate representation where a reference allele will be added, and the allele depth will be estimated from the alternate allele depth and coverage.

- Star alleles: Accurate representation where star alleles are used to specify complex overlapping alleles.
- Without overlap specification: this is how complex variants used to be handled in previous versions of the *CLC Genomics Workbench*, where complex overlap does not affect how variants are specified.

Read more about these options in Complex variant representations and VCF reference overlap [8.1.7](#)

Export no-call records Some export parameter settings can result in removal of all alleles at a given locus present in the exported variant track. Enable this option to export such loci where no alleles are called. In the generated no-call record, the genotype will be specified as missing, however the original variant annotations will be available. No-call records may occur when 'Remove alleles below fraction threshold' is enabled, when enforcing a maximum ploidy, or when using the 'Reference overlap and depth estimate' complex variant representation.

Maximum InDel length The maximum length at which insertions and deletions (InDels) are represented with full sequence in the VCF. Variants that are longer than the threshold, will instead be included in the VCF as symbolic alleles.

For full compatibility with QIAGEN Clinical Insight Interpret (QCI Interpret), the threshold should be set to 1000.

Output as single file When this option is checked, data from multiple input tracks, including CNV tracks and fusion tracks, are exported together to a single VCF file.

Important details about VCF export

- When working with fusion data, only fusions with "PASS" in the "Filter" column will be exported.
- Where the same variant is reported multiple times, which is especially relevant when providing multiple variant tracks as input, the VCF file will include only one of these, the copy with the highest QUAL value.
- Counts from the variant track are put in CLCAD2 or AD fields depending on the chosen complex variant representation, and coverage is placed in the DP field. The values of the CLCAD2 tag follow the order of REF and ALT, with one value for the REF and for each ALT. For example, if there has been a homozygote variant identified at a certain position, the value of the GT field is 1/1 and the corresponding CLCAD2 value for the reference allele will be 0, which is always the first number in the CLCAD2 field. Please note that this does not mean the original *mapping* did not have any reads with that sequence, but it means that the *variant track* being exported does not contain the reference allele.

For descriptions of general export settings, see section [8.1.2](#) and section [8.1.3](#).

Complex variant representations and VCF reference overlap

Allelic variants that overlap but do not cover the same range are called **complex variants**. Whenever two independently called variant sets are joined, there is a chance of getting complex variants. Complex variants comprise 1.4% of variants called by the CLC Fixed Ploidy Variant Detection tool. The popular GATK haplotype caller also encounters this phenomenon.

It is tricky to describe complex variants in VCF since they have to be written on different lines due to their position while they also need to be specified in the genotype field of each line without referring to the other lines. It may be possible to extend or split overlapping variants to match each other's position in order to comply with the VCF format, however that will lead to inaccuracies when assigning attributes, such as read count and coverage, to the altered variants.

GATK4 outputs complex variants with a genotype field that includes a reference allele for each overlapping allele, thereby also indicating that the variant is heterozygous. Since it means that two VCF lines will contradict each other, it can be argued that this representation is counter-intuitive. RTG tools' vcfeval provides an option called "-ref-overlap" to handle this representation. When interpreting complex variants represented this way, in case of conflict, non-reference alleles trump reference alleles.

To allow flexibility for communication with variant tracks, we provide the users with the following representation options for the **VCF Export** tool:

- The reference overlap representation as described above, where reference alleles are added to the genotype field of complex variants. We refer to this as the "Reference overlap" option. We also provide a version of the "Reference overlap" option with allele depth estimation.
- The legacy VCF export format (as available in previous versions of the software)
- The star allele format, based on the star allele introduced in VCF v4.2.

All of these complex variant representations can be handled by the **VCF Import** tool. A comparison of the options available is presented in figure 8.9:

	Without overlap specification	Reference overlap	Reference overlap with depth estimate	Star alleles
Advantage	no difference (legacy)	maximum precision	maximum compatibility	maximum precision
Allele depth (read count) format field¹	CLCAD2	CLCAD2	AD	AD
Zygosity of complex variants can be determined from VCF genotype field	no	yes	yes	yes
Length and read support can be specified for each individual complex reference allele	yes	yes	no	yes
Appropriate for variant database export²	yes	no	no	yes
Can be used in other applications without special consideration of complex reference alleles	yes	yes/no ³	no	yes
Can be used in other applications without special consideration of star allele interpretation	yes	yes	yes	no

¹ The AD field is used if allele fraction can be calculated based on allele depth alone. If the DP field is also required, then the CLCAD2 field is used.

² For databases where no sample allele depth is available, reference overlap may in some situations merge or switch the reference alleles in complex variants.

³ Removal of reference alleles with zero allele depth may be necessary. Many applications ignore reference alleles, so often the added reference overlap alleles will not require any special attention.

Figure 8.9: *Main characteristics of the complex variant representations.*

Without overlap specification This is the representation used previously, where only variants that are present at the exact same ref positions are specified in the VCF genotype field. Variants that partially overlap do not affect the genotype field. Using this complex variant representation, two types of information are not available in the genotype field that is available for non-complex variants: zygosity of the variant, and the ploidy of the sample at the position.

Suggested use cases: export of database variants without sample specific annotations (such as clinvar), where specification of sample haplotype structure is not necessary. Also use for applications tailored to handle this legacy format.

Reference overlap This representation both allow specification of zygosity, ploidy, and phasing in the genotype field, as well as exact read support and length for complex reference alleles. At positions with complex alternate variants, a reference allele is specified in the VCF genotype field for each reference and alternate allele overlapping the position, these are termed reference overlap alleles. The allele depth is left at zero for reference overlap alleles, indicating that they are merely placeholders for overlapping alleles. The length and allele depth of complex reference alleles are specified separately, so the properties they have in the variant track are retained.

Suggested use cases: this should be the general first choice, since it is an accurate representation of the variants, widely compatible with downstream applications

Reference overlap with depth estimate This is the most compliant representation, where both the genotype and allele depth fields consider all alleles that overlap the position. In VCF files using the AD field for read count, it is common to be able to calculate allele frequency using the formula: $\text{frequency} = \text{AD} / \text{sum}(\text{AD})$, and that is also possible using this complex variant representation. The reference allele depth represents the combined read depth of overlapping alleles and reference alleles at the position, and is estimated as total read coverage (DP field) minus the combined allele depth of the ALT alleles at the position. This representation only specifies reference alleles together with alternate alleles. The main disadvantage of this representation is that it is not possible to specify exactly what the read support is for a complex reference allele, due to the fact that the reference allele depth is mixed with the overlapping allele depth. Complex reference alleles will get an average allele depth of the overlapping and reference alleles that are present at a position.

Suggested use cases: export of variants for use in applications that cannot handle the more accurate "Reference overlap" representation

Star alleles According to the VCF specification, star alleles are reserved for overlapping deletions, however some applications treat these in a way that is applicable to all types of overlapping variants. Since the overlapping deletion is defined in another VCF line, and it is unclear if the star allele signifies that the whole position is covered by the deletion, it is sometimes not appropriate to treat the star allele as an actual variant. The star allele can be interpreted merely as providing genotype information for the position, such as zygosity, ploidy, phasing and allele frequencies, whereas the actual overlapping variant will be dealt with at its start position where it is described in detail. This is the way the star allele is interpreted during VCF import in the CLC workbench. When using the star allele complex variant representation it is important to check if the variants are used in an application that handles the star alleles in a way similar to how the CLC workbench does, or if the star alleles are interpreted as actual deletion

variants. In the latter case, another complex variant representation should be considered. This representation estimates the star allele depth, i.e. the number of reads supporting the overlapping alleles, to be the difference between the total read coverage and the combined allele depth of the variants at the position. Thus, the allele fraction can be calculated based on allele depth alone, and therefore the AD field is used for allele depth.

Suggested use cases: This representation is accurate and does not require any special reference allele handling (no reference overlap). It should be used for all applications that handle star alleles as described above.

An example of export and import using the different complex variant representations is shown in figure 8.10:

Reference
9bp

GGGGGGGGG

Read mapping
10 reads

Variants
Variants (3)

AA

TT

GGG

Variants exported

CHROM

POS

ID

REF

ALT

QUAL

FILTER

INFO

FORMAT

sampleName

Variants re-imported

AA

TT

GGG

Without overlap specification

1

4

.

GG

TT

200.00

.

.

GT:CLCAD2:DP

1:0,4:10

1

4

.

GGG

.

.

.

GT:CLCAD2:DP

0:2:10

1

5

.

GG

AA

200.00

.

.

GT:CLCAD2:DP

1:0,4:10

Reference overlap

1

4

.

GG

TT

200.00

.

.

GT:CLCAD2:DP

0/0/1:0,4:10

1

4

.

GGG

.

.

.

GT:CLCAD2:DP

0:2:10

1

5

.

GG

AA

200.00

.

.

GT:CLCAD2:DP

0/0/1:0,4:10

Reference overlap and depth estimate

1

4

.

GG

TT

200.00

.

.

GT:AD:DP

0/0/1:6,4:10

1

4

.

GGG

.

.

.

GT:AD:DP

.:2:10

1

5

.

GG

AA

200.00

.

.

GT:AD:DP

0/0/1:6,4:10

Star alleles

1

4

.

GG

TT,*

200.00

.

.

GT:AD:DP

1/2/2:0,4,6:10

1

4

.

GGG

*

.

.

GT:AD:DP

0/1/1:2,8:10

1

5

.

GG

AA,*

200.00

.

.

GT:AD:DP

1/2/2:0,4,6:10

Figure 8.10: Example of export and import using the different complex variant representations.

8.1.8 GFF3 export

Annotation tracks, as well as annotations on sequences (individual elements or in sequence lists), can be exported to a GFF3 format file using the dedicated GFF3 exporter.

A subset of annotations on a sequence or sequence list can be exported from the Annotation Table (📄) view: select the annotations to export in the table, right-click, and choose **Export Selection to GFF3 File** from the menu (figure 15.20). Further information about sequence annotations is in section 15.3.

Maintaining relationships when exporting annotation tracks

When relationships between annotations should be maintained, for example the relationship of transcripts to genes, all the tracks containing related annotations must be exported together, with the "Output as single file" option selected.

Relationships are represented in GFF3 using Parent and ID tags, as described in the GFF3 documentation (<http://gmod.org/wiki/GFF3>). See also section 7.2.1. Parent tags cannot be preserved if relevant tracks are not included or if different annotation types are output to separate files.

For example, annotations in transcript tracks typically contain a Parent linking the transcript to a gene. If the transcript and gene tracks are exported together, the Parent tags are preserved. If this GFF3 file is later imported, the relationships between transcripts and genes will be intact in the resulting tracks.

8.1.9 BED export

Annotation and expression tracks can be exported to a BED format file using the dedicated BED exporter. See <http://genome.ucsc.edu/FAQ/FAQformat.html#format1> for format details. Note that:

- White spaces in the annotation/gene/transcript name are replaced with _.
- The thickStart and thickEnd BED fields are populated with the same values as the chromStart and chromEnd BED fields, respectively. I.e. the start and end positions of the feature in the chromosome or scaffold.
- The itemRgb BED field is assigned the value 0.
- The score BED field is assigned values as follows:
 - For annotation tracks, if the annotation does not contain a score, the value is set to 0. Otherwise, the annotation score is used, where negative scores are set to 0 and scores larger than 1000 are set to 1000.
 - For expression tracks, the score reflects the expression value and it is set to

$$\left\lfloor \frac{\log(v - m + 1)}{\log(M - m + 1)} \times 1000 \right\rfloor$$

where v is the expression value, and m/M is the lowest/largest observed expression values in the entire expression track.

8.1.10 JSON export

Reports can be exported in JSON format. An exported JSON file contains 4 main elements:

- **header.** Contains information about the version of the JSON exporter and front page elements included in the report (the front page elements are visible in the PDF export of the report).
- **data.** Contains the actual data found in the report (sections, subsections, figures, tables, text).
- **metadata.** Contains information about metadata files the report referenced to.
- **history.** Contains information about the history of the report (as seen in the "Show history" view).

The data section contains nested elements following the structure of the report:

- The keys of sections (and subsections, etc) are formed from the section (and subsection, etc) title, with special characters replaced. For example, the section "Counted fragment by type (total)" is exported to an element with the key "counted_fragments_by_type_total".
- A section is made of the section title, the section number, and all other elements that are nested in it (e.g., other subsections, figures, tables, text).
- Figures, tables and text are exported to elements with keys "figure_n", "table_n" and "text_n", n being the number of the elements of that type in the report.
- Figures contain information about the titles of the figure, x axis, and y axis, as well as legend and data. This data is originally available in the Workbench by double clicking on a figure in a report and using the "Show Table" view.
- The names of table columns are transformed to keys in a similar way to section titles.

Once exported, the JSON file can be parsed and further processed. For example, using R and the package jsonlite, reports from different samples can be jointly analyzed. This enables easy comparison of any information present in the original reports across samples.

Note that the tool Combine Reports tool (see section 37.5) already provides a similar functionality, but the JSON export allows for more flexibility in what is compared across samples.

Example of R script to generate a combined RNA-Seq report

R scripts can be used to process reports in JSON format. The script below is an example of how RNA-Seq reports generated by the CLC Genomics Workbench could be processed to compare information across samples.

The script uses jsonlite to parse all the JSON reports. Plots are produced using the package ggplot2. This script is intended for inspiration only and is not supported.

```
library(jsonlite)
library(tools)
library(ggplot2)
```

The script relies on the following functions to extract the data from the parsed JSON files.

```
# Get the read count statistics from a parsed JSON report.
get_read_count_stats <- function(parsed_report) {
  mapping_statistics <- parsed_report$data$mapping_statistics
  total_reads <- 0
  stats <- c()
  if ("single_reads" %in% names(mapping_statistics)) {
    table <- mapping_statistics$single_reads$table_1
    # use the id column to give names to the rows
    row.names(table) <- table$id
    stats <- c(table["Reads mapped", "percent"],
              table["Reads not mapped", "percent"])
    total_reads <- total_reads + table["Total", "number_of_sequences"]
  }
```

```

    }
    else {
      stats <- c(stats, rep(NA, 2))
    }
    if ("paired_reads" %in% names(mapping_statistics)) {
      table <- mapping_statistics$paired_reads$table_1
      # use the id column to give names to the rows
      row.names(table) <- table$id
      stats <- c(stats,
        table["Reads mapped in pairs", "percent"],
        table["Reads mapped in broken pairs", "percent"],
        table["Reads not mapped", "percent"])
      total_reads <- total_reads + table["Total", "number_of_sequences"]
    }
    else {
      stats <- c(stats, rep(NA, 3))
    }
    stats <- c(total_reads, stats)
    names(stats) <- c("reads_count", "single_mapped", "single_not_mapped",
      "paired_mapped_pairs", "paired_broken_pairs",
      "paired_not_mapped")
    return(data.frame(sample = basename(file_path_sans_ext(report)),
      t(stats)))
  }
}

#' Get the paired distance from a parsed report. Returns null if the reads were
#' unpaired.
get_paired_distance <- function(parsed_report) {
  section <- parsed_report$data$read_quality_control
  if (!("paired_distance" %in% names(section))) {
    return(NULL)
  } else {
    figure <- section$paired_distance$figure_1
    return(data.frame(sample = basename(file_path_sans_ext(report)),
      figure$data))
  }
}

#' Get the figure, x axis, and y axis titles from the paired distance figure
#' from a parsed report. Returns null if the reads were unpaired.
get_paired_distance_titles <- function(parsed_report) {
  section <- parsed_report$data$read_quality_control
  if (!("paired_distance" %in% names(section))) {
    return(NULL)
  } else {
    figure <- section$paired_distance$figure_1
    return(c("title" = figure$figure_title,
      "x" = figure$x_axis_title,
      "y" = figure$y_axis_title))
  }
}

#' Re-order the intervals for the paired distances by using the starting value of the interval.
order_paired_distances <- function(paired_distance) {
  distances <- unique(paired_distance$distance)
  starting <- as.numeric(sapply(strsplit(distances, split = " - "), function(l) l[1]))
  distances <- distances[sort.int(starting, index.return = TRUE)$ix]
  paired_distance$distance <- factor(paired_distance$distance, levels = distances)
  # calculate the breaks used on the x axis for the paired distances
  breaks <- distances[round(seq(from = 1, to = length(distances), length.out = 15))]
  return(list(data = paired_distance, breaks = breaks))
}

```

```
}
```

Using the above functions, the script below parses all the JSON reports found in the "exported reports" folder, to build a read count statistics table (read_count_statistics), and a paired distance histogram.

```
reports <- list.files("exported reports/", full.names = TRUE)

read_count_statistics <- data.frame()
paired_distance <- data.frame()
titles <- c(NA, NA, NA)

for (report in reports) {
  parsed_report <- fromJSON(report)



  read_count_statistics <- rbind(read_count_statistics,
                                get_read_count_stats(parsed_report))
  paired_distance <- rbind(paired_distance,
                           get_paired_distance(parsed_report))
  titles <- get_paired_distance_titles(parsed_report)
}

paired_distance <- order_paired_distances(paired_distance)

ggplot(paired_distance$data, aes(x = distance, y = number_of_reads, fill = sample)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_x_discrete(breaks = paired_distance$breaks, labels = paired_distance$breaks) +
  labs(title = titles["title"], x = titles["x"], y = titles["y"]) +
  theme(legend.position = "bottom")
```

8.1.11 Graphics export

CLC Genomics Workbench supports two ways of exporting graphics:

- You can export the current view, either the visible area or the entire view, by clicking on the **Graphics** button () in the top Toolbar. This is the generally recommended route for exporting graphics for individual data elements, and is described in section 8.2.
- For some data types, graphics export tools are available from the main **Export** menu, which can be opened by clicking on the **Export** () button in the top Toolbar. These are useful if you wish to export different data using the same view in an automated fashion, for example by running the export tool in batch mode or in a workflow context. This functionality is described below.

Using export tools to export graphics

The following types of data can be exported using dedicated export tools:

- Alignments
- Heat maps
- Read mappings

- Sequences
- Tracks
- Track lists

The general actions taken are:

- Click on the **Export** (📄) button in the top Toolbar or choose the **Export** option under the File menu.
- Type "graphics" in the top field to see just a list of graphics exporters, and then select the one you wish to use. For example, if you wish to export an alignment as graphics, select "Alignment graphics" in the list.
- Select the data elements to be exported.
- Configure any relevant options. Detailed descriptions of these are provided below.
- Select where the data should be exported to.

Options available when exporting sequences, alignments and read mappings to graphics format files are shown in figure 8.11.

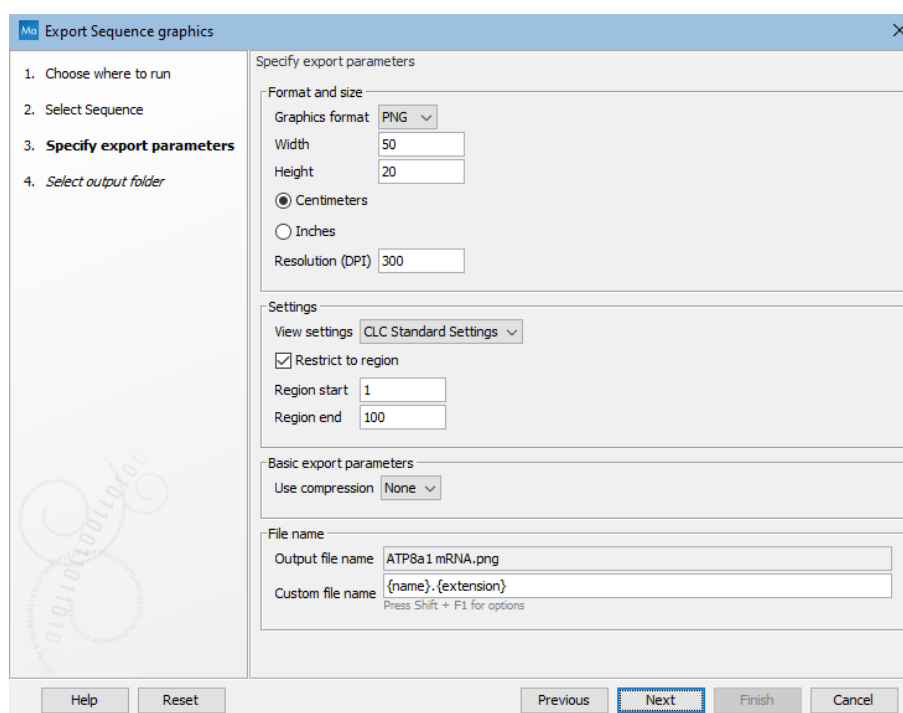


Figure 8.11: Options available when exporting sequences, alignments and read mappings to graphics format files.

The options available when exporting tracks and track lists to graphics format files are shown in figure 8.12.

The format and size of the exported graphics can be configured using:

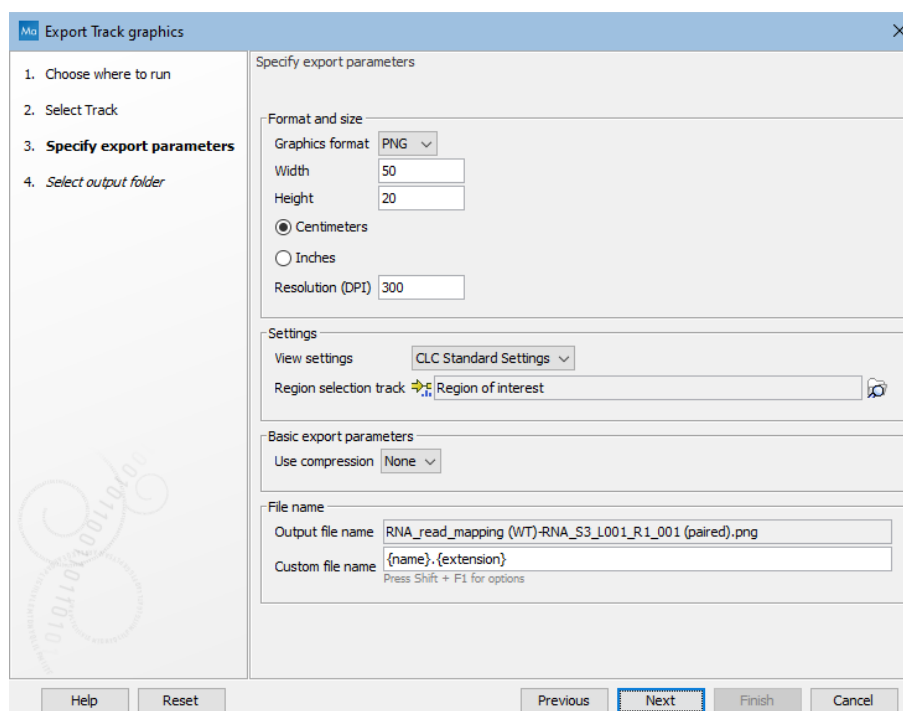


Figure 8.12: Options available when exporting tracks and track lists to graphics format files.

- **Graphics format:** Several export formats are available, including bitmap formats (such as .png, .jpg) and vector graphics (.svg, .ps, .eps).
- **Width and height:** The desired width and height of the exported image. This can be specified in centimeters or inches.
- **Resolution:** The resolution, specified in the units of "dpi" (dots per inch).


The appearance of the exported graphics can be configured using:


- **View settings:** The view settings available for the data type being exported. To determine how the data will look when a particular view is used, open a data element of the type you wish to export, click on the **Save View** button visible at the bottom of the Side Panel, and apply the view settings in the dialog that appears. View settings are described in section 4.6. Custom view settings will be available to choose from when exporting if the "Save for all <data type> views" option was checked when the view was saved.
- **Region restriction:** The region to be exported. For sequences, alignments and read mappings, the region is specified using start and end coordinates. For tracks and track lists, you provide an annotation track, where the region corresponding to the full span of the *first* annotation is exported. The rest of the annotations in the track have no effect.

8.1.12 Export history

The history for a data element can be exported to PDF or to CSV format files. This includes information like the date and time data was imported or an analysis was run, the parameters and values set, and where the data came from. For elements created by a workflow, the name and

version of that workflow is included in the PDF export. If created using an installed workflow, the workflow build-id is also included (figure 8.15).

The history information for an element can be seen in the *CLC Genomics Workbench* by clicking on the Show History view () at the bottom of the viewing area when a data element is open (see section 2.5).

To export the history of a data element, click on the **Export** () in the Toolbar and select **History PDF** or **History CSV** (figure 8.13).

After selecting the data to export the history for, you can configure standard export parameters for the type of format you are exporting to (figure 8.14).

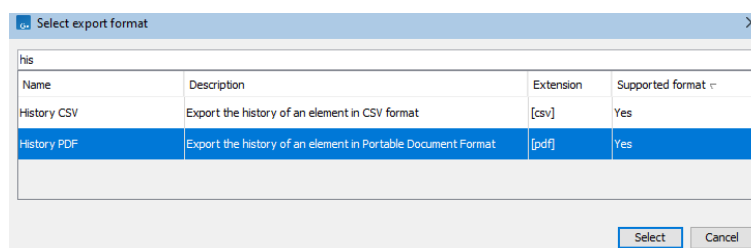


Figure 8.13: Select "History PDF" for exporting the history of an element as a PDF file.

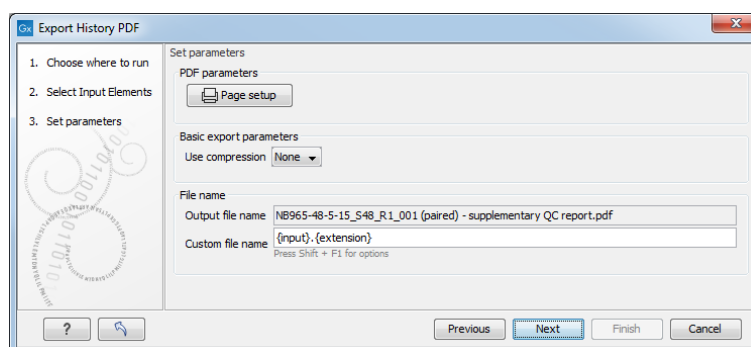



Figure 8.14: When exporting the history in PDF, it is possible to adjust the page setup.

History for:
 **Fwd1 Contig 1**
Workflow details: assemble-seqs-wf 2.1, Build: 290923-1316



Assemble Sequences	
29 Sep 2023 13:17:44	
Version:	CLC Genomics Workbench
Modified by:	
Minimum aligned read length	50
Alignment stringency	Medium
Conflicts	Vote (A, C, G, T)
Comments:	Number of reads: 4 Average coverage: 1.73 Fwd1 (trimmed) was reversed during the assembly process

Trim Sequences 1.1	
29 Sep 2023 13:17:43	
Version:	CLC Genomics Workbench
Modified by:	
Ignore existing trim information	Yes
Trim using quality scores	Yes
Quality limit	0.05
Trim using ambiguous nucleotides	Yes
Residues	2
Trim contamination from vectors in	No
UniVec database	
Trim contamination from	No


Figure 8.15: An example of the top of the exported PDF containing the history of an element generated using an installed workflow.

8.2 Export graphics to files

CLC Genomics Workbench supports two ways of exporting graphics:

- You can export the current view, either the visible area or the entire view, by clicking on the **Graphics** button  in the top Toolbar. This is the generally recommended route for exporting graphics for individual data elements, and is described below.
- For some data types, graphics export tools are available in the main **Export** menu, which can be opened by clicking on the **Export**  button in the top Toolbar. These are useful if you wish to export different data using the same view in an automated fashion, for example by running the export tool in batch mode or in a workflow context. That functionality is described in section 8.1.11.

Exporting a view of data element to a graphics format file

To export a view of an open data element to a graphics file, click on the **Graphics** button  in the top Toolbar, or choose **Export Graphics** from under the File menu.

How the data looks on the screen is how it will look in the exported file. Before exporting, the options in the Side Panel can be used to make adjustments as necessary.

For views that can be zoomed into or out of, you will be offered the choice of exporting the whole view or just the visible area (figure 8.16). For 3D structures, the section visible will always be exported, i.e. the equivalent to selecting to export just the visible area.

A view of a circular sequence, zoomed in so that you can only see a part of it, is shown in figure 8.17.

When the option **Export visible area** is selected, the exported file will only contain the part of the sequence that is *visible* in the view. The result of doing this for the view shown in figure 8.17 is shown in figure 8.18.

If you select **Export whole view**, the result would look like that shown in figure 8.19, where the

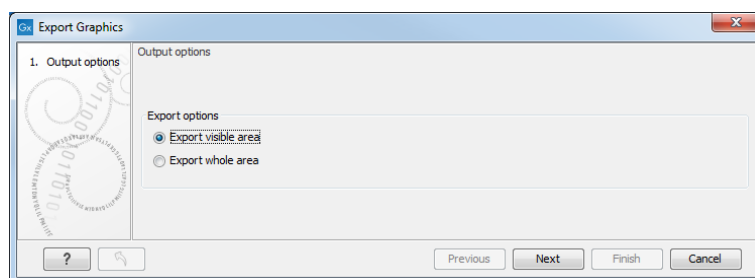


Figure 8.16: The whole view or just the visible area can be selected for export.

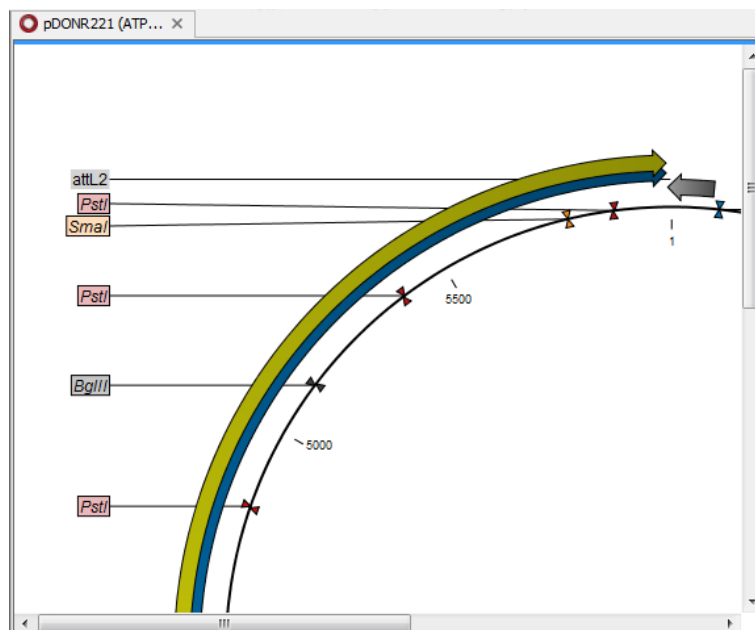


Figure 8.17: A circular sequence, as it looks on the screen when zoomed in.

part of the sequence not visible on screen is also exported.

You are prompted for the file format to use, and the name and location to save the output to.

For vector-based image formats, click on **Finish** to start the export. For bitmap-based image formats, click on **Next** to select a resolution level. You can click on **Finish** directly instead, in which case the image will be exported using the setting used the last time this functionality was run. If it has not been run before the default settings are used.

The available export formats are described in section [8.2.1](#).

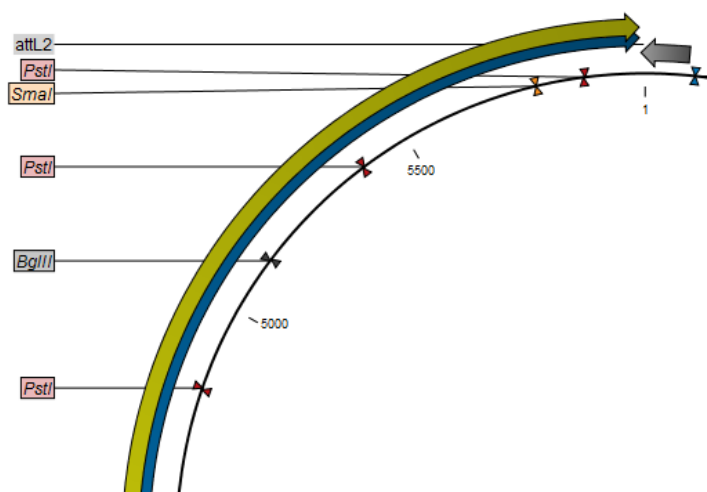


Figure 8.18: The exported graphics file when *Export visible area* was selected.

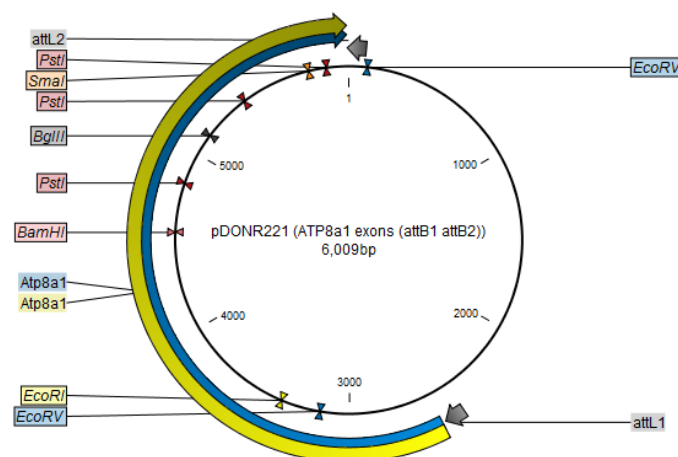


Figure 8.19: The exported graphics file when *Export whole view* was selected. The whole sequence is shown, not just the part visible on screen when the view was exported.

8.2.1 File formats

CLC Genomics Workbench supports the following file formats for graphics export:

Format	Suffix	Type
Portable Network Graphics	.png	bitmap
JPEG	.jpg	bitmap
Tagged Image File	.tif	bitmap
PostScript	.ps	vector graphics
Encapsulated PostScript	.eps	vector graphics
Portable Document Format	.pdf	vector graphics
Scalable Vector Graphics	.svg	vector graphics

These formats can be divided into bitmap and vector graphics. The difference between these two

categories is described below:

Bitmap images In a bitmap image, each dot in the image has a specified color. This implies, that if you zoom in on the image there will not be enough dots, and if you zoom out there will be too many. In these cases the image viewer has to interpolate the colors to fit what is actually looked at. A bitmap image needs to have a high resolution if you want to zoom in. This format is a good choice for storing images without large shapes (e.g. dot plots). It is also appropriate if you don't have the need for resizing and editing the image after export.

To produce a high resolution image with all the details of a large element visible, e.g. a large phylogenetic tree or a read mapping, we recommend exporting to a vector based format.

If Screen resolution and High resolution settings show the same pixel dimensions, this can be because the maximum supported number of pixels has been exceeded.

Parameters for bitmap formats For bitmap files, clicking **Next** will display the dialog shown in figure 8.20.

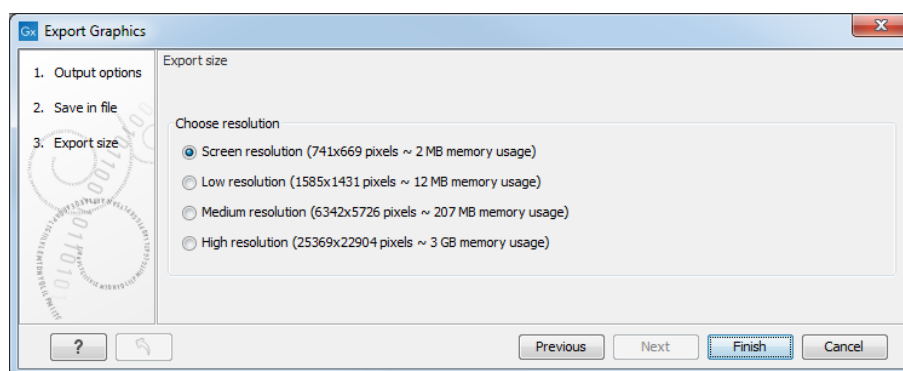


Figure 8.20: *Parameters for bitmap formats: size of the graphics file.*

You can adjust the size (the resolution) of the file to four standard sizes:

- Screen resolution
- Low resolution
- Medium resolution
- High resolution

The actual size in pixels is displayed in parentheses. An estimate of the memory usage for exporting the file is also shown. If the image is to be used on computer screens only, a low resolution is sufficient. If the image is going to be used on printed material, a higher resolution is necessary to produce a good result.

Vector graphics Vector graphic is a collection of shapes. Thus what is stored is information about where a line starts and ends, and the color of the line and its width. This enables a given viewer to decide how to draw the line, no matter what the zoom factor is, thereby always giving a correct image. This format is good for graphs and reports, but less usable for dot plots. If the

image is to be resized or edited, vector graphics are by far the best format to store graphics. If you open a vector graphics file in an application such as Adobe Illustrator, you will be able to manipulate the image in great detail.

Graphics files can also be imported into the **Navigation Area**. However, no kinds of graphics files can be displayed in *CLC Genomics Workbench*. See section 3.2 for more about importing external files into *CLC Genomics Workbench*.

Parameters for vector formats For PDF format, the dialog shown in figure 8.21 will sometimes appear after you have clicked finished (for example when the graphics use more than one page, or there is more than one PDF to export).

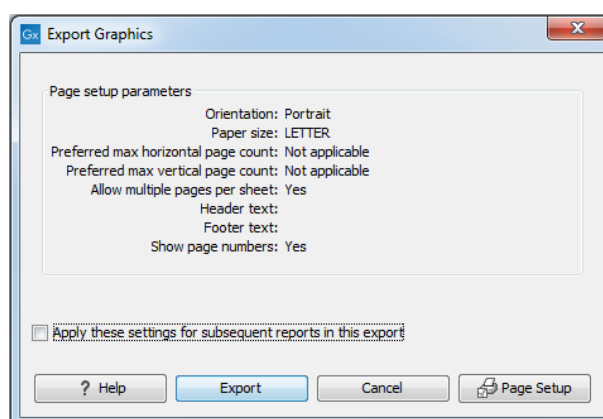


Figure 8.21: Page setup parameters for vector formats.

The settings for the page setup are shown. Clicking the **Page Setup** button will display a dialog where these settings can be adjusted. This dialog is described in section 5.2.

It is then possible to click the option "Apply these settings for subsequent reports in this export" to apply the chosen settings to all the PDFs included in the export for example.

The page setup is only available if you have selected to export the whole view - if you have chosen to export the visible area only, the graphics file will be on one page with no headers or footers.

Exporting protein reports It is possible to export a protein report using the normal **Export** function (📄) which will generate a pdf file with a table of contents:

Click the report in the Navigation Area | Export (📄) in the Toolbar | select pdf

You can also choose to export a protein report using the **Export graphics** function (🖨️), but in this way you will not get the table of contents.

8.3 Export graph data points to a file

Data points for graphs displayed along the sequence or along an alignment or mapping can be exported to a semicolon-separated text file (csv format). An example of such a graph is shown in figure 8.22, showing the conservation score of reads in a read mapping.

To export the data points for the graph, right-click the graph and choose **Export Graph to Comma-separated File**. Depending on what kind of graph you have selected, different options

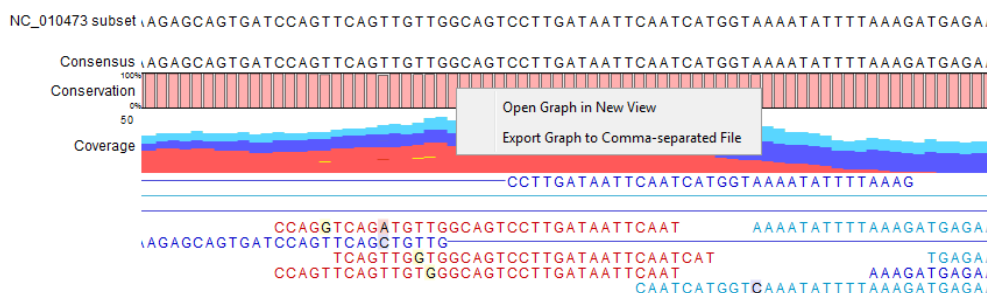


Figure 8.22: A conservation graph displayed along mapped reads. Right-click the graph to export the data points to a file.

will be shown: If the graph is covering a set of aligned sequences with a main sequence, such as read mappings and BLAST results, the dialog shown in figure 8.23 will be displayed. These kinds of graphs are located under **Alignment info** in the Side Panel. In all other cases, a normal file dialog will be shown letting you specify name and location for the file.

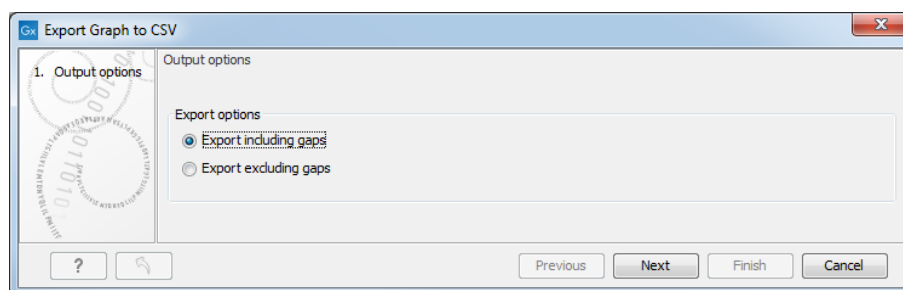


Figure 8.23: Choosing to include data points with gaps

In this dialog, select whether you wish to include positions where the main sequence (the reference sequence for read mappings and the query sequence for BLAST results) has gaps. If you are exporting e.g. coverage information from a read mapping, you would probably want to exclude gaps, if you want the positions in the exported file to match the reference (i.e. chromosome) coordinates. If you export including gaps, the data points in the file no longer corresponds to the reference coordinates, because each gap will shift the coordinates.

Clicking **Next** will present a file dialog letting you specify name and location for the file.

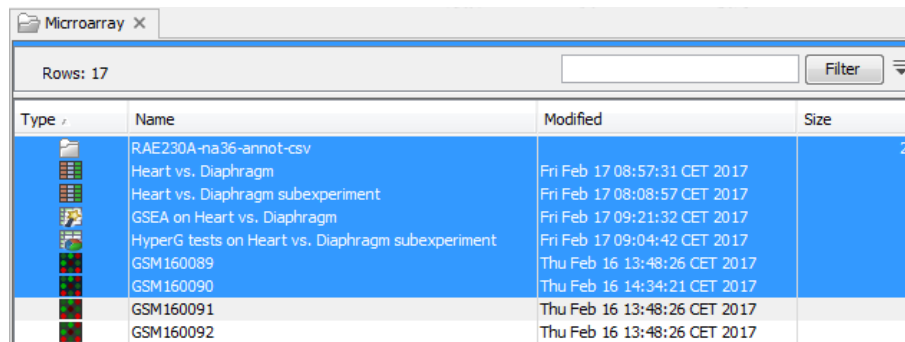
The output format of the file is like this:

```
"Position"; "Value";
"1"; "13";
"2"; "16";
"3"; "23";
"4"; "17";
...
```

8.4 Copy/paste view output

The contents of various element types can be copied and then pasted into programs outside of the Workbench. For example, tables from reports, table views of sequence lists, and tabular listings of folders, can be copied and then pasted into text editors or programs like Excel.

Example: Right click a folder in the Navigation Area and choose **Show | Content**. The folder contents are shown as a table in the viewing area. Select one or more of the rows and then copy them (Ctrl + C). That information can then be pasted into other programs (figure 8.24)



Type	Name	Modified	Size
File	RAE230A-na36-annot-csv		2
File	Heart vs. Diaphragm	Fri Feb 17 08:57:31 CET 2017	
File	Heart vs. Diaphragm subexperiment	Fri Feb 17 08:08:57 CET 2017	
File	GSEA on Heart vs. Diaphragm	Fri Feb 17 09:21:32 CET 2017	
File	HyperG tests on Heart vs. Diaphragm subexperiment	Fri Feb 17 09:04:42 CET 2017	
File	GSM160089	Thu Feb 16 13:48:26 CET 2017	
File	GSM160090	Thu Feb 16 14:34:21 CET 2017	
File	GSM160091	Thu Feb 16 13:48:26 CET 2017	
File	GSM160092	Thu Feb 16 13:48:26 CET 2017	

Figure 8.24: Selected elements in a Folder Content view.

Workflow designs can be copied as an image. To do this, select the elements in the workflow design (click in the workflow editor and then press keys Ctrl + A), then copy (Ctrl + C), and then paste where you wish the image to be placed, for example, in an email or presentation program. Workflows are described in detail in chapter 14.

Chapter 9

Working with tables

Contents

9.1	Table view settings and column ordering	188
9.2	Filtering tables	190

General features relevant to many table types are described in this section. For functionality associated with specific table types, please refer to the manual section describing that particular data type.

Key functionality available for tables includes:

- **Sorting** A table can be sorted according to the values of a particular column by clicking a column header. Clicking once will sort in ascending order. A second click will change the order to descending. A third click will set the order back its original order.
Pressing Ctrl - ⌘ on Mac - while you click other columns will refine the existing sorting with the values of the additional columns, in the order in which you clicked them.
- **Configuring the view** This includes specifying which columns should be visible and defining the column order (see section 9.1). View settings can be saved for later use, with a specific table or for any similar table. (see section or chapter 4.6).
- **Displaying only the selected rows** Click on the **Filter to Selection...** button above a table to update the view to show only the selected rows.
Rows can be selected manually, or by using the "Select in other views" option, which is available for some tables, generally those with an associated graphical view such as a Venn diagram, or a volcano plot.
To view the full table again, click on the **Filter to Selection...** button and choosing the option **Clear selection filter**.
- **Displaying only rows with content of interest** Tables can be interactively filtered using simple or complex search criteria such that only rows containing content of interest are shown. Sets of table filters can be saved for re-use. See section or chapter 9.2 for details.

Scroll bars appear at the bottom and at the right of a table when table contents exceed the size of the viewing area.

Table-specific right-click menu options

Right-clicking within a table reveals standard right-click menu options, as well as the following table-specific options:

- **File | Export Table** Export the table to CSV, TSV, HTML or Excel format. Filtering, sorting, column selection and column order are respected when exporting the table this way.
- **Edit | Copy Cell** Right-click on a cell and choose this option to copy the contents of that cell to the clipboard.

The option call **Table filters**, also available in the right-click menu, is explained in section or chapter 9.2.

9.1 Table view settings and column ordering

View settings for tables are configurable in the Side Panel (figure 9.1). These settings include how column widths should be determined, as well as which columns should be visible and the column order.

Saving view settings

View settings can be saved for continued use with this table, or later use with this table or other similar tables. Saving view settings is described in section or chapter 4.6. Table specific considerations when saving view settings include:

- If saved view settings are applied to a table that contains columns not defined in those view settings, those columns will be placed at the far right of the table.
- Saved view settings referring to columns not present in the table that they are being applied to are ignored.

Column width view settings

Column width options in the Side Panel are:

- **Automatic** Columns are sized to fit the width of the viewing area.
- **Manual** The width of each column can be adjusted manually.

Columns to show in the table

The columns available for that table are listed. Checking the box beside a column name makes that column visible in the viewing area. Unchecking that box hides the column.

Select All and **Deselect All** buttons allow you to make all columns visible in the viewing area, or hide all columns, in a single click.

Start codon	Start	End	Length	Found at strand
ACT	1	348	348	positive
TGC	433	789	357	positive
AGG	18424	18747	324	positive
TCC	19378	19719	342	positive
AAC	34264	34572	309	positive
TGG	41083	41403	321	positive
AGT	56899	57273	375	positive
GGC	92872	93231	360	positive
CCA	104482	104826	345	positive
AAT	111553	111885	333	positive
CCT	115306	115644	339	positive
AAA	136333	136773	441	positive
GCA	137026	137457	432	positive
AGT	139486	139953	468	positive
CTC	143110	143415	306	positive
CAC	145483	145815	333	positive
AGT	148756	149109	354	positive
ACT	149401	149856	456	positive
CTG	152509	152871	363	positive
GTT	156415	156765	351	positive
AAA	158331	158666	336	positive

Figure 9.1: A table with all but one available columns visible, and the "Start codon" column moved to the start of the table from its original location, which was at the end of the table.

Changing the column order

The column order in tables can be changed. This affects the order in the view as well as column order in files exported from the table.

Ways to change column order in a table:

1. **Drag the column to the desired location in the viewing area.**

Click on the column heading in the viewing area and, keeping the mouse button depressed, drag it to the desired location in the table.

The order of the columns in the Side Panel is updated automatically. Check the Side Panel listing to ensure the column is positioned as desired relative to any hidden (unchecked) columns.

2. **Move the column to the desired location in the Show columns palette in the Side Panel.**

Hover over the column name in the Side Panel, revealing the (⤴) icon, then depress the mouse button and drag the column to the position desired.

The order of the columns in the viewing area is updated automatically.

3. **Apply saved view settings where a relevant column order has been defined.** See section or chapter 4.6 for details about applying saved view settings.

Files exported from a table open for viewing, such as .csv files, can be exported using this custom column order. See section 8.1.6 for details.

9.2 Filtering tables

The rows shown in a table view can be limited to just those of interest by simple or advanced filtering, defined using functionality just above the table. Saving and reusing sets of table filters is described at the end of this section.

Filtering the rows of a table interactively does *not* change the underlying content of the table. Some table types have a button allowing the creating of a new table containing only the visible rows.

The **Filter on Custom Criteria** tool, described in section 37.3, can be used to filter tables non-interactively, generating a new table as a result.

Simple filtering

The default view of a table supports simple filtering, where rows containing a particular search term can be entered into a field to the left of the **Filter** button (figure 9.2). Simple filtering is enabled when there is an upwards pointing arrow at the top right of the table view. The keyboard shortcut Ctrl + F (mac: ⌘ + F) jumps the cursor into the simple filter field. (Clicking on the arrow beside that field reveals advanced filtering options, which are described later in this section.)

Simple filtering starts automatically, as you type, unless the table has more than 10,000 rows. In that case, click on the **Filter** button after typing the term to filter for.

The number of rows with a match to the term is reported in the top left of the table.

The following characters have special meanings when used in the simple filtering field:

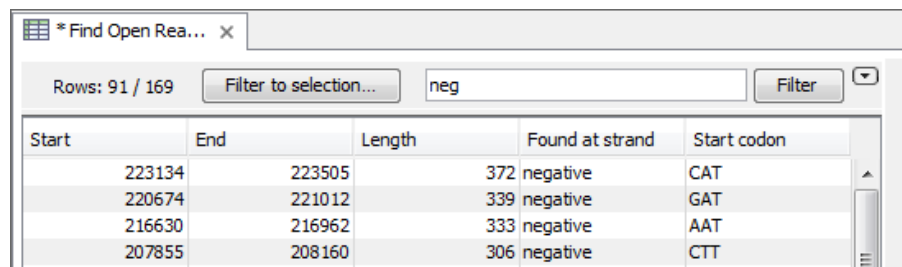
- **Space** Terms separated by spaces are treated as individual search terms unless the terms are placed within quotes. E.g. the term `cat dog` would return all rows with the term `cat` and/or the term `dog` in them, in any order.
- **Single and double quotes ' and "** Enclose a term containing spaces in quotes to search for exactly that term. E.g. `"cat dog"` would return rows containing the single term `cat dog`.
- **Backslash** Use this term to escape special characters. For example, to search for the term `"cat"` including the quotation marks, enter `\ "cat\"`.
- **Minus -** Place a minus symbol before a term to exclude rows containing that term. e.g. `-cat -dog` would exclude all rows containing either `cat` or `dog`.
- **Colon :** Specify the name of a column to be searched for the term. E.g. `Animal:cat` would search for the term `cat` only in a column called `Animal`. For this sort of filtering, please also refer to the advanced filtering information, below.

Advanced filtering

Functionality to define sets of filter criteria is revealed by clicking on the downwards-pointing arrow at the top right of the table view, (figure 9.3).

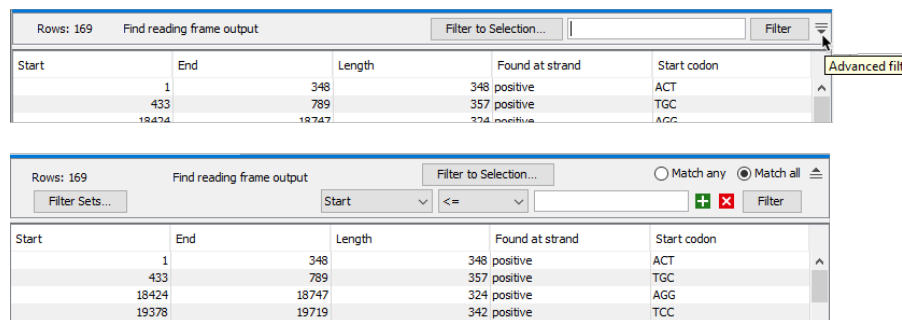
Each filter criterion consists of a column name, an operator and a value. Examples are described below.

Filter criteria can be added by:



Start	End	Length	Found at strand	Start codon
223134	223505	372	negative	CAT
220674	221012	339	negative	GAT
216630	216962	333	negative	AAT
207855	208160	306	negative	CTT

Figure 9.2: Filtering for rows that contain the term "neg" using the Filter button

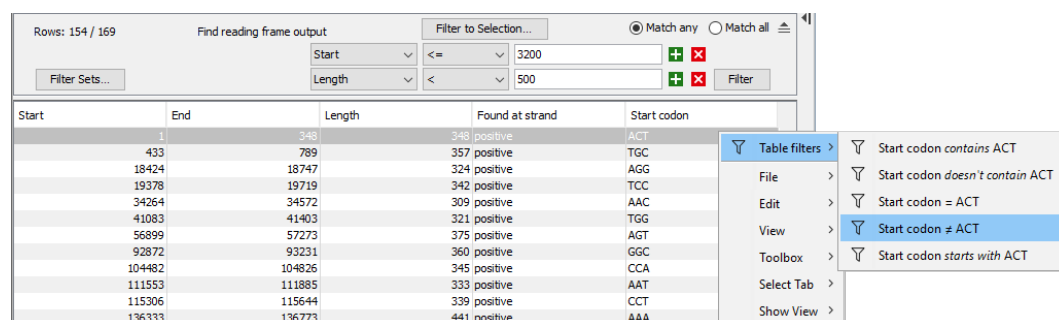


Start	End	Length	Found at strand	Start codon
1	348	348	positive	ACT
433	789	357	positive	TGC
18424	18747	324	positive	AGG
19378	19719	342	positive	TCC

Figure 9.3: When the Advanced filter icon is clicked on (top), Advanced filtering fields are revealed (bottom)

- Clicking the **Add** (+) icon.
- Right-clicking on a value in the table and selecting the **Table filters** option from the menu that appears. Predefined criteria for that column and value combination will be listed (figure 9.4). Selecting one of these adds it to the list of filters at the top of the table.

Filter criteria can be removed by clicking on the (X) icons.



Start	End	Length	Found at strand	Start codon
1	348	348	positive	ACT
433	789	357	positive	TGC
18424	18747	324	positive	AGG
19378	19719	342	positive	TCC
34264	34572	309	positive	AAC
41083	41403	321	positive	TGG
56899	57273	375	positive	ACT
92872	93231	360	positive	GGC
104482	104826	345	positive	CCA
111553	111885	333	positive	AAT
115306	115644	339	positive	CCT
136333	136773	441	positive	AAA

Figure 9.4: Right-click on a cell value and choose Table filters to reveal predefined criteria that can be added to the list of filters for this table.

Match all and **Match any** options allow you to specify, respectively, whether all criteria must be met for a row to be shown, or whether matching a single criteria is enough for a row to be shown (figure 9.5).

The number of rows with a match to the term is reported in the top left of the table.

Operators available for columns containing text are listed below. Tests for matches are not case specific.

Top Screenshot (Match all selected):

Rows: 6 / 169

Find reading frame output

Filter to Selection...

Start <= 3200

Length < 500

Filter

Start	End	Length	Found at strand	Start codon
1		348	348 positive	ACT
433		789	357 positive	TGC
462		767	306 positive	ACC
1998		2309	312 negative	AAT
405		800	396 negative	GTT
1378		1752	375 negative	TAT

Bottom Screenshot (Match any selected):

Rows: 154 / 169

Find reading frame output

Filter to Selection...

Start <= 3200

Length < 500

Filter

Start	End	Length	Found at strand	Start codon
1		348	348 positive	ACT
433		789	357 positive	TGC
18424		18747	324 positive	AGG
19378		19719	342 positive	TCC
34264		34572	309 positive	AAC
41083		41403	321 positive	TGG
56899		57273	375 positive	AGT

Figure 9.5: The same two criteria are defined, but with "Match all" selected in the top image, and "Match any" selected in the bottom image.. Six rows out of 169 match all the criteria, while 154 rows match one or both criteria.

- **contains**
 - **doesn't contain**
 - **=** Matches exactly
 - **≠** Does not match
 - **starts with** Start with the term provided.
 - **is in list** Matches at least one of the entries in the list.
 - **is not in list** Does not match any entry in the list.
- Terms in lists can be comma, semicolon, or space separated.

Operators available for columns containing numerical values are:

- **<=** Smaller than or equal to
- **<** Smaller than
- **>=** Greater than or equal to
- **>** Greater than
- **abs. value <** Absolute value smaller than.
- **abs. value >** Absolute value greater than.
- **=** Equal to
- **≠** Not equal to

- **is in list** Matches at least one of the entries in the list.
- **is not in list** Does not match any entry in the list.

Terms in lists can be comma, semicolon, or space separated.

Number formatting and filter criterion: The number of digits to display after the decimal separator (fractional digits) can be set in the *CLC Genomics Workbench* Preferences. Thus, there may be more digits in a number stored in a table than are shown in a view of that table. For this reason, we recommend using operators that do not require exact matches, such as \geq , when filtering on non-integer values.

Saving and reusing table filter sets

Sets of filter criteria defined in the advanced filtering area at the top of a table can be saved for re-use. Filter sets can also be exported and imported, supporting sharing of filter sets with others.

Options for saving and managing filter sets are provided in a menu revealed when you click on the **Filter Sets...** button (figures 9.6 and 9.8). Saved filter sets are also listed here (figure 9.7). Selecting a saved set from this menu will add those conditions to the top of the table, and apply the filtering.

Saved filter sets can also be applied from the Manage Filters dialog (figure 9.8).

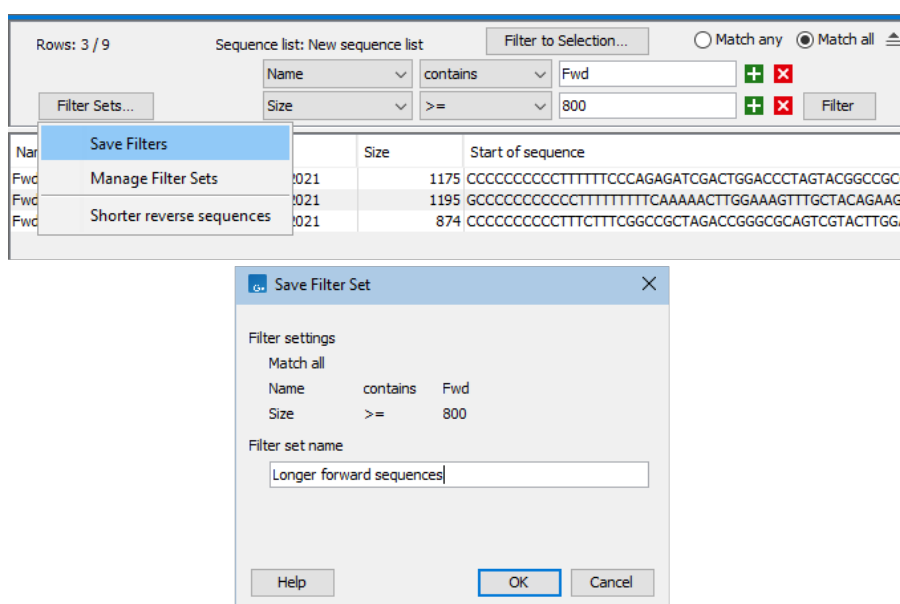


Figure 9.6: Selecting *Save Filters* from the menu under the *Filter Sets...* button (top) opens a dialog showing the filter criteria and prompting for a name for the filter set (bottom).

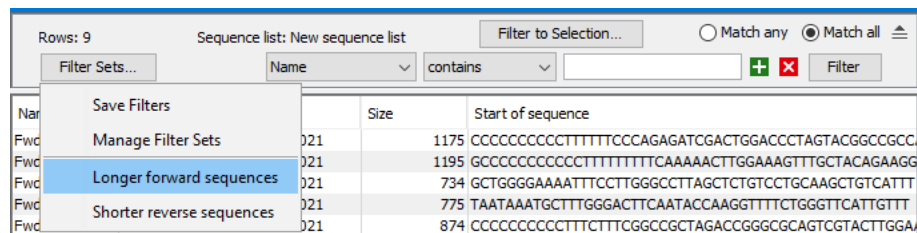


Figure 9.7: Saved filter sets are listed at the bottom of the drop-down menu revealed when you click on the Filter Sets... button.

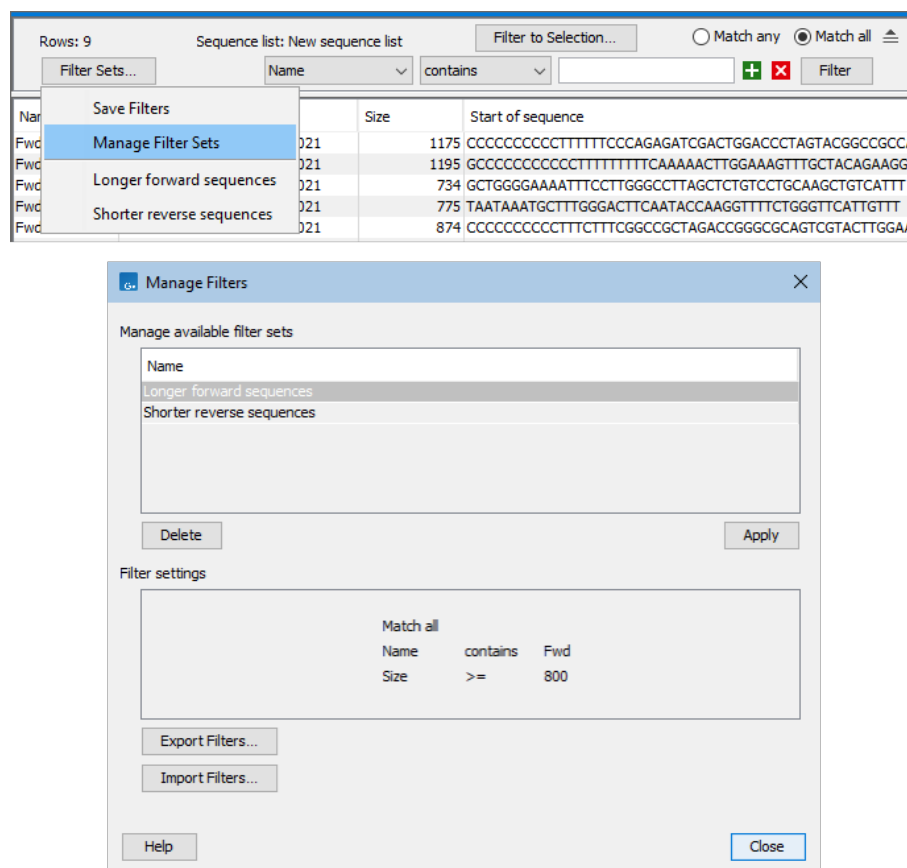


Figure 9.8: Selecting Manage Filters from the menu under the Filter Sets... button (top) opens the Manage Filters dialog, where saved filter sets can be applied to the open table, or deleted. Functionality to export and import filter sets is also provided here (bottom).

Chapter 10

Data download

Contents

10.1 Search for Sequences at NCBI	195
10.1.1 NCBI search options	196
10.1.2 Handling of NCBI search results	197
10.2 Search for PDB Structures at NCBI	198
10.2.1 Structure search options	199
10.2.2 Handling of NCBI structure search results	199
10.2.3 Save structure search parameters	201
10.3 Search for Sequences in UniProt (Swiss-Prot/TrEMBL)	201
10.3.1 UniProt search options	202
10.3.2 Handling of UniProt search results	203
10.4 Search for Reads in SRA	205
10.4.1 Searching SRA	205
10.4.2 Downloading reads and metadata from SRA	207
10.4.3 Troubleshooting SRA downloads	213
10.5 Sequence web info	214

CLC Genomics Workbench offers different ways of searching and downloading online data. You must be online when initiating and performing the following searches.

10.1 Search for Sequences at NCBI

Search for Sequences at NCBI supports searching the Entrez nucleotide, peptide and EST databases (<https://www.ncbi.nlm.nih.gov/books/NBK44864/>).

To start this tool, go to:

Download | Search for Sequences at NCBI () or **Ctrl + B** (⌘ + B on Mac)

Select the nucleotide, protein or EST database using options at the top of the view (figure 10.1).

The search you just set up can be saved by doing one of the following:

- Choose **Save As...** from under the File menu, or

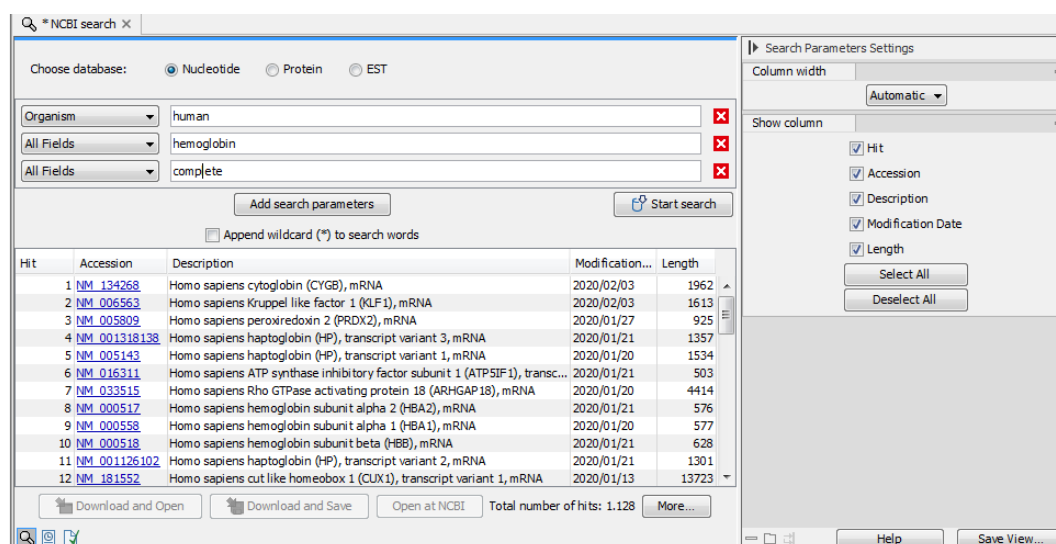


Figure 10.1: The GenBank search view.

- Click on the tab of the search view and drag and drop it into a folder in the Navigation Area.

These actions save the search query. (It does not save the search results.)

This can be useful when you run the same searches periodically.

10.1.1 NCBI search options

Conducting a search using **Search for Sequences at NCBI** corresponds to using the same query terms on the NCBI website. For example, searching the nucleotide database corresponds to a search at <https://www.ncbi.nlm.nih.gov/nucleotide/>. A list of the entries matching the search terms is returned. Those of interest can then be selected and then downloaded, ready for downstream use in the Workbench.

Click on **Add search parameters** to add parameters to your search. The following are available:

- **All Fields** Searches for the terms provided in all fields of the NCBI database.
- **Organism**
- **Definition/Title**
- **Modified** Search for entries modified within the period specified from a drop-down list.
- **Gene Location** Choose from Genomic DNA/RNA, Mitochondrion, or Chloroplast.
- **Molecule** Choose from Genomic DNA/RNA, mRNA or rRNA.
- **Sequence Length** Enter a number for a maximum or minimum length of the sequence.
- **Gene Name**
- **Accession**

Check the "Append wildcard (*) to search words" checkbox to indicate that the term entered should be interpreted as the first part of the term only. E.g. searching for "genom" with that box checked would find entries starting with that term, such as "genomic" and "genome".

When you are satisfied with the parameters you have entered, click on the **Start search** button.

Additional search tips

Feature keys can be added when searching for nucleotide sequences. Writing

gene[Feature key] AND mouse

searches for one or more genes and where 'mouse' appears somewhere in the entry. For more information about how to use this syntax, see <https://www.ncbi.nlm.nih.gov/books/NBK3837/>

Where multiple terms are entered into a single field, the terms are usually treated as if they have an AND between them, which means that entries returned must contain all the specified terms. A notable exception is if multiple accessions are listed in an "All Fields" search field, in which case a results for each valid accession listed is returned.

You can add the words "AND", "OR" or "NOT" to the terms in a given field to further customize your search. Further details about searching Entrez sequence databases can be found at <https://www.ncbi.nlm.nih.gov/books/NBK44864/>.

10.1.2 Handling of NCBI search results

A list of entries matching the search terms provided is returned. By default, the initial 50 results are returned, with 50 additional results loaded each time the **More...** button is clicked. This number can be configured in the Workbench Preferences (section 4).

Five pieces of information are presented for each entry:

- **Hit** The rank of the entry in the search result
- **Accession** The accession for that entry. Click on the link to open that entry's page at the NCBI in a web browser.
- **Description** The definition line of the entry
- **Modification date** The date the entry was last updated in the database searched
- **Length** The length of the sequence

The columns to display can be configured in "Show column" tab of right hand, side panel settings.

Select one or more rows of the table and use buttons at the bottom of the view to:

- **Download and Open** Sequences are opened in a new view after download is complete.
You can also download and open sequences by dragging selected rows to a new tab area or by double-clicking on a row.
- **Download and Save** Sequences are downloaded and saved to a location you specify.

You can also download and save sequences by selecting rows and copying them (e.g. using Ctrl + C), and then selecting a folder in the **Navigation Area** and pasting (e.g. using Ctrl + V).

- **Open at NCBI** The sequence entry page(s) at the NCBI are opened in a web browser.

The functions offered by these buttons are also available in the menu that appears if you right-click over selected rows.

Note: The modification date on sequences downloaded can be more recent than those reported in the results table. This depends on the database versions made available for searching at the NCBI.

Downloading and saving sequences can take some time. This process runs in the background, so you can continue working on other tasks. The download process can be seen in the **Status bar** and it can be stopped, if desired, as described in section 2.4.

10.2 Search for PDB Structures at NCBI

This section describes searches for three dimensional structures from the NCBI structure database <https://www.ncbi.nlm.nih.gov/Structure/MMDB/mmdb.shtml>. For manipulating and visualization of the downloaded structures see section 17.2.

The NCBI search view is opened in this way:

Download | Search for PDB Structures at NCBI (🌐)

or **Ctrl + B** (⌘ + B on Mac)

This opens the view shown in figure 10.2:

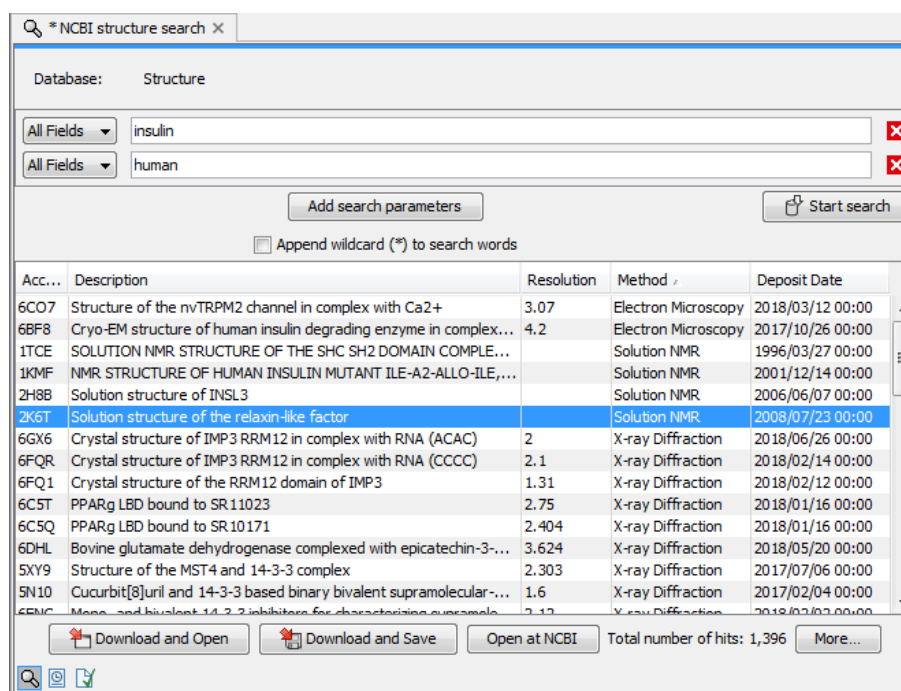


Figure 10.2: The structure search view.

10.2.1 Structure search options

Conducting a search in the **NCBI Database** from *CLC Genomics Workbench* corresponds to conducting search for structures on the NCBI's Entrez website. When conducting the search from *CLC Genomics Workbench*, the results are available and ready to work with straight away.

As default, *CLC Genomics Workbench* offers one text field where the search parameters can be entered. Click **Add search parameters** to add more parameters to your search.

Note! The search is a "AND" search, meaning that when adding search parameters to your search, you search for both (or all) text strings rather than "any" of the text strings.

You can append a wildcard character by clicking the checkbox at the bottom. This means that you only have to enter the first part of the search text, e.g. searching for "prot" will find both "protein" and "protease".

The following parameters can be added to the search:

- **All fields.** Text, searches in all parameters in the NCBI structure database at the same time.
- **Organism.** Text.
- **Author.** Text.
- **PdbAcc.** The accession number of the structure in the PDB database.

The search parameters are the most recently used. The **All fields** allows searches in all parameters in the database at the same time.

All fields also provide an opportunity to restrict a search to parameters which are not listed in the dialog. E.g. writing 'gene[Feature key] AND mouse' in **All fields** generates hits in the GenBank database which contains one or more genes and where 'mouse' appears somewhere in GenBank file. NB: the 'Feature Key' option is only available in GenBank when searching for nucleotide structures. For more information about how to use this syntax, see http://www.ncbi.nlm.nih.gov/entrez/query/static/help/Summary_Matrices.html#Search_Fields_and_Qualifiers

When you are satisfied with the parameters you have entered click **Start search**.

Note! When conducting a search, no files are downloaded. Instead, the program produces a list of links to the files in the NCBI database. This ensures a much faster search.

10.2.2 Handling of NCBI structure search results

The search result is presented as a list of links to the files in the NCBI database. The **View** displays 50 hits at a time (can be changed in the **Preferences** (see chapter 4)). More hits can be displayed by clicking the **More...** button at the bottom right of the **View**.

Each structure hit is represented by text in three columns:

- Accession.

- Description.
- Resolution.
- Method.
- Protein chains
- Release date.

It is possible to exclude one or more of these columns by adjust the View preferences for the database search view. Furthermore, your changes in the View preferences can be saved. See section 4.6.

Several structures can be selected, and by clicking the buttons in the bottom of the search view, you can do the following:

- **Download and open.** Download and open immediately.
- **Download and save.** Download and save lets you choose location for saving structure.
- **Open at NCBI.** Open additional information on the selected structure at NCBI's web page.

Double-clicking a hit will download and open the structure. The hits can also be copied into the **View Area** or the **Navigation Area** from the search results by drag and drop, copy/paste or by using the right-click menu as described below.

Drag and drop from structure search results

The structures from the search results can be opened by dragging them into a position in the **View Area**.

Note! A structure is not saved until the **View** displaying the structure is closed. When that happens, a dialog opens: Save changes of structure x? (Yes or No).

The structure can also be saved by dragging it into the **Navigation Area**. It is possible to select more structures and drag all of them into the **Navigation Area** at the same time.

Download structure search results using right-click menu

You may also select one or more structures from the list and download using the right-click menu (see figure 10.3). Choosing **Download and Save** lets you select a folder or location where the structures are saved when they are downloaded. Choosing **Download and Open** opens a new view for each of the selected structures.

The selected structures are not downloaded from the NCBI website but is downloaded from the RCSB Protein Data Bank <http://www.rcsb.org/pdb/home/home.do> in PDB format.

Copy/paste from structure search results

When using copy/paste to bring the search results into the **Navigation Area**, the actual files are downloaded.

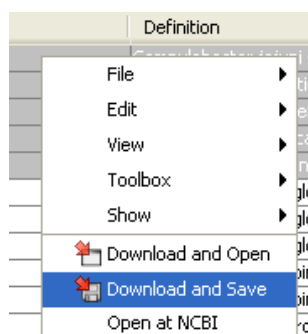


Figure 10.3: By right-clicking a search result, it is possible to choose how to handle the relevant structure.

To copy/paste files into the **Navigation Area**:

select one or more of the search results | **Ctrl + C** (⌘ + C on Mac) | **select location or folder in the Navigation Area** | **Ctrl + V**

Note! Search results are downloaded before they are saved. Downloading and saving several files may take some time. However, since the process runs in the background (displayed in the **Status bar**) it is possible to continue other tasks in the program. Like the search process, the download process can be stopped. This is done in the in the **Processes** tab, available in the Toolbox area in the bottom left side of the Workbench.

10.2.3 Save structure search parameters

The search you just set up can be saved by doing one of the following:

- Choose **Save As...** from under the File menu, or
- Click on the tab of the search view and drag and drop it into a folder in the Navigation Area.

These actions save the search query. (It does not save the search results.)

This can be useful when you run the same searches periodically.

10.3 Search for Sequences in UniProt (Swiss-Prot/TrEMBL)

Search for Sequences in UniProt searches the UniProt Knowledgebase (UniProtKB). UniProtKB contains UniProtKB/Swiss-Prot entries, which are manually curated, and UniProtKB/TrEMBL entries, which are annotated by automated systems.

To open the tool, go to:

Download | **Search for Sequences in UniProt** (🔍)

One or more search terms can be entered, and a table of entries matching the search terms is returned. Rows in that table can be selected and the corresponding UniProtKB entries downloaded and opened, or downloaded and saved (figure 10.4).

Searching is described in more detail in section 10.3.1.

Working with results returned by a search is described in section 10.3.2.

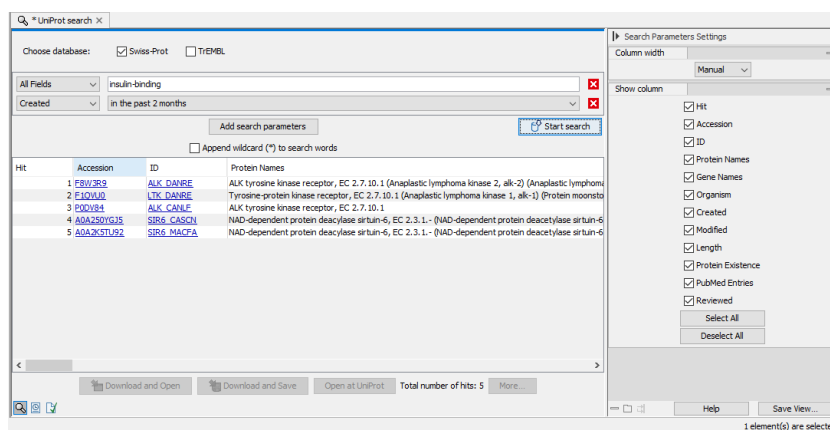


Figure 10.4: Search in UniProtKB by entering search terms and clicking on the "Start search" button. A table containing information about entries matching the query terms is returned.

10.3.1 UniProt search options

Database choice

Select one of the 2 subsections of UniProtKB to search in, or select both to search all of UniProtKB.

- **Swiss-Prot** Searches among manually curated entries. These are the entries marked as "reviewed" in UniProtKB.
- **TrEMBL** Searches among computationally analyzed entries that have been annotated using automated systems. These are the entries marked "unreviewed" in UniProtKB.

Search fields

A single search field is presented by default. Click on "Add search parameters" to add more.

The following options are available:

- **All fields** Search for the term provided in all fields available at the UniProtKB website <https://www.uniprot.org/>.
- **Accession** Search for entry accessions.
- **Protein Names** Search for terms in protein names.
- **Gene Names** Search for terms in gene names.
- **Organism** Search for terms in organism names.
- **Created** Search for entries created within the period specified from a drop-down list.
- **Modified** Search for entries modified within the period specified from a drop-down list.
- **Protein existence.** Search for entries with the evidence level specified from a drop-down list.

When the **Append wildcard (*) to search words** is checked, the search is broadened to include entries containing terms starting with text you provided.

Click on the **Start search** button to run the search.

Information about entries meeting all the conditions specified is returned in a table. No data is downloaded at this point. Working with these results, including downloading entries, is described in section 10.3.2.

Saving a configured UniProt search

The search you just set up can be saved by doing one of the following:

- Choose **Save As...** from under the File menu, or
- Click on the tab of the search view and drag and drop it into a folder in the Navigation Area.

These actions save the search query. (It does not save the search results.)

This can be useful when you run the same searches periodically.

10.3.2 Handling of UniProt search results

The results table

By default, the first 50 entries found in UniProtKB that met all the search conditions are returned in the results table. Clicking on the **More...** button at the bottom right loads further results.

The default value of 50 can be changed. See chapter 4 for details.

For each entry, the following information is provided:

- **Hit** The position of the entry in the results. E.g. 1 for the first entry in the list returned, 2 for the second, and so on.
- **Accession** The accession of the entry. Clicking on the link opens the entry's page at the UniprotKB website.
- **ID** The ID of the entry. Clicking on the link opens the entry's page at the UniprotKB website.
- **Protein Names** The protein names and synonyms.
- **Gene Names** The names of the gene(s) encoding the protein.
- **Organism** The scientific name and synonyms of the source organism.
- **Created** The date the entry was created.
- **Modified** The date of the latest annotation update.
- **Length** The length of the canonical sequence.
- **Protein Existence** The level of evidence supporting the existence of the protein.

- **Pubmed Entries** The list of Pubmed IDs mapped to the entry. Clicking on the link opens a page listing these Pubmed entries.
- **Reviewed** Either "reviewed" for entries in Swiss-Prot, or "unreviewed" for entries in TrEMBL.

The columns displayed can be customized using in the side panel settings. See section 4.6 for details.

If you wish to open webpages for several entries at once, highlight the rows of interest and click on the **Open at UniProt** button.

Downloading results from UniprotKB

Entries from UniprotKB can be downloaded and saved, or downloaded and opened, directly in the viewing area. When opened directly, the entries are not saved until you take explicit action to do so (see section 2.1.2).

Downloading many large files may take some time. The download process can be stopped from under the Processes tab (see section 2.4).

To download and save entries, select the rows of interest and then do one of the following:

- Click on the **Download and Save** button. You will be prompted for a location to save the entries to.
- Right-click over a selected area and choose the option **Download and Save** from the menu presented.
- Drag the row(s) to a folder in the Navigation Area.
- Copy (Ctrl-C) to copy the entry information. Click on a folder in the Navigation Area and then paste (Ctrl-V).

The selected entries are downloaded from UniprotKB. Multiple entries selected at the same time are saved to a single protein sequence list.

To download and open entries directly in the viewing area, select the rows of interest and then do one of the following:

- Click on the **Download and Open** button.
- Right-click over a selected area and choose the option **Download and Open** from the menu presented.
- Drag the row(s) until the mouse cursor is next to an existing tab in the view area. When the mouse button is released, a new tab is opened, and the selected entries are downloaded and opened in that tab.

Double-clicking on a single row will download and open that entry.


10.4 Search for Reads in SRA

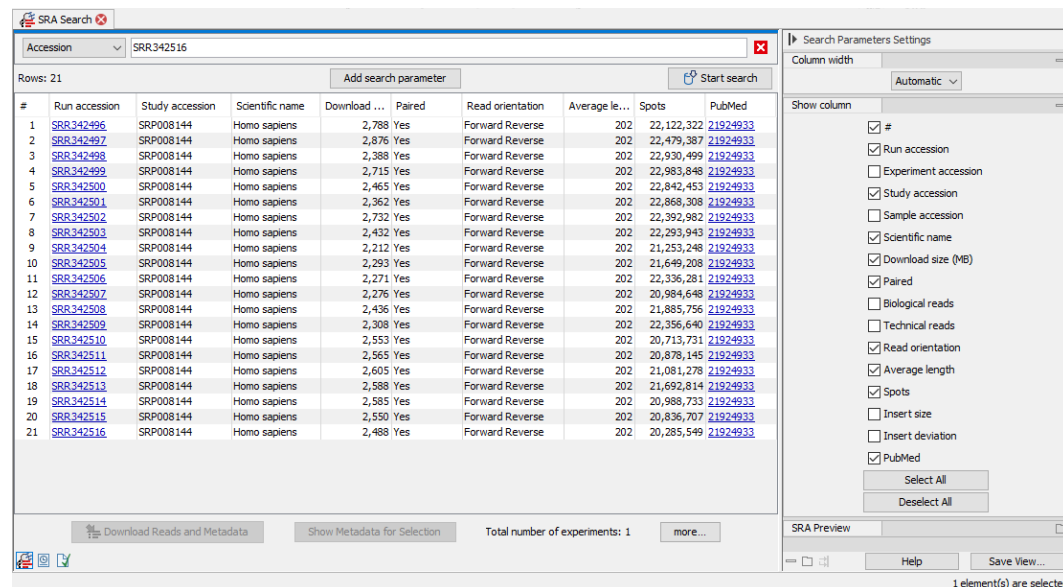
10.4.1 Searching SRA

Search for Reads in SRA searches the NCBI SRA database for entries with metadata matching the query terms you provide. The sequences and metadata for selected runs can be downloaded. Alternatively, just the metadata for the selected runs can be saved for use later.

Search for Reads in SRA uses NCBI's e-utilities, which occasionally experience downtime. If searches unexpectedly return no results, go to <https://www.ncbi.nlm.nih.gov/sra/> to check the status of the service.

Start **Search for Reads in SRA** by going to:

Download | Search for Reads in SRA 



#	Run accession	Study accession	Scientific name	Download ...	Paired	Read orientation	Average le...	Spots	PubMed
1	SRR342496	SRP008144	Homo sapiens	2,788	Yes	Forward Reverse	202	22,122,322	21924933
2	SRR342497	SRP008144	Homo sapiens	2,876	Yes	Forward Reverse	202	22,479,387	21924933
3	SRR342498	SRP008144	Homo sapiens	2,388	Yes	Forward Reverse	202	22,930,499	21924933
4	SRR342499	SRP008144	Homo sapiens	2,715	Yes	Forward Reverse	202	22,983,848	21924933
5	SRR342500	SRP008144	Homo sapiens	2,465	Yes	Forward Reverse	202	22,842,453	21924933
6	SRR342501	SRP008144	Homo sapiens	2,362	Yes	Forward Reverse	202	22,868,308	21924933
7	SRR342502	SRP008144	Homo sapiens	2,732	Yes	Forward Reverse	202	22,392,982	21924933
8	SRR342503	SRP008144	Homo sapiens	2,432	Yes	Forward Reverse	202	22,293,943	21924933
9	SRR342504	SRP008144	Homo sapiens	2,212	Yes	Forward Reverse	202	21,253,248	21924933
10	SRR342505	SRP008144	Homo sapiens	2,293	Yes	Forward Reverse	202	21,649,208	21924933
11	SRR342506	SRP008144	Homo sapiens	2,271	Yes	Forward Reverse	202	22,336,281	21924933
12	SRR342507	SRP008144	Homo sapiens	2,276	Yes	Forward Reverse	202	20,984,648	21924933
13	SRR342508	SRP008144	Homo sapiens	2,436	Yes	Forward Reverse	202	21,885,756	21924933
14	SRR342509	SRP008144	Homo sapiens	2,308	Yes	Forward Reverse	202	22,356,640	21924933
15	SRR342510	SRP008144	Homo sapiens	2,553	Yes	Forward Reverse	202	20,713,731	21924933
16	SRR342511	SRP008144	Homo sapiens	2,565	Yes	Forward Reverse	202	20,878,145	21924933
17	SRR342512	SRP008144	Homo sapiens	2,605	Yes	Forward Reverse	202	21,081,278	21924933
18	SRR342513	SRP008144	Homo sapiens	2,588	Yes	Forward Reverse	202	21,692,814	21924933
19	SRR342514	SRP008144	Homo sapiens	2,585	Yes	Forward Reverse	202	20,988,733	21924933
20	SRR342515	SRP008144	Homo sapiens	2,550	Yes	Forward Reverse	202	20,836,707	21924933
21	SRR342516	SRP008144	Homo sapiens	2,488	Yes	Forward Reverse	202	20,285,549	21924933

Download Reads and Metadata | Show Metadata for Selection | Total number of experiments: 1 | more...

1 element(s) are selected

Figure 10.5: "Accession" was selected from the drop-down menu of search fields, and a single accession was entered. The results of the search include the run with the accession provided, as well as runs submitted to SRA as part of the same experiment.

Search fields

A drop-down menu at the top left lists the fields that can be searched (figure 10.5). Generally, all search terms provided must be present in an SRA entry for it to be returned. Within a single search field, OR can be added between terms to indicate that just one of the terms needs to match. The exception to this is the Accession field, which is described further below.

Details about selected search fields:

- **All Fields** All fields are searched with the terms provided.

Example queries:

"Plasmodium falciparum" "Plasmodium vivax" would yield a list of runs where both terms were found within any of the fields searched.

"Plasmodium falciparum" OR "Plasmodium vivax" would yield a list of runs where either term was found within any of the fields searched.

SRR20016268 SRR20016241 SRR20016341 would return no results because an "AND" is assumed between each term. Contrast this with searching in the Accession field with the same terms, described below.

- **Accession** Provide a single accession, or multiple accessions separated by spaces, commas or semicolons. Each term is searched for individually. I.e. terms in a single Accession field are handled as if they were separated by OR. Unlike in other fields writing "OR" between terms is not supported as it is already assumed.

Example query:

SRR20016268 SRR20016241 SRR20016341 would return 3 entries.

The run with a specified accession is returned, as well as any other runs that were submitted as part of the same experiment. To retrieve all runs in a study, provide the study accession as the query term.

- **Modification/Publication date** Find entries modified/published within a date range, specified as [MM] [YYYY]. For example, a publication date given the following range 08 2016 to 08 2016 would return results published between the first and last days in August 2016.
- **Strategy** Select from a drop-down list of types of experiments e.g., RNA-Seq, ChIP-Seq, etc.
- **Library Selection** Select from a drop-down list of known library preparation methods, e.g. Poly(A), Size fractionation, etc.
- **Platform** Select from a drop-down list of NGS sequencing platforms e.g., Illumina, Ion Torrent, etc. Note: Download of data from some platforms, such as Complete Genomics, is not supported.
- **Instrument** Select from a drop-down list of individual NGS sequencing machines e.g., HiSeq X Ten, Ion Torrent PGM, etc.
- **Paired Status** The options are Paired and Single. When Paired is selected, SRA runs specified as paired are returned. Selecting Single returns runs where paired status has not been specified.
- **Availability** Select Public or dbGaP. The latter contains confidential data. Entries in dbGAP can be searched and metadata returned can be saved, but reads cannot be downloaded directly. Access to dbGAP involves an application to the NCBI.
- **PubMed** Select "has abstract" to find entries with a PubMed abstract or "has full-text article" for entries where the entire publication is available.

SRA search results

A table with one row per result is returned. The columns to display in the table can be configured in the side panel, on the right.

Details about selected column contents:

- **Run Accession** The accession for an SRA run, hyperlinked to the relevant NCBI webpage, where additional information can be found.

- **Download size** The size of the SRA format (.sra) file for that run. At least twice this amount of space should be available as temporary space during download and import. See section 10.4.2 for more on space requirements.
- **Biological reads** The number of biological reads per spot. If there is no read type information for that run in SRA, all reads are assumed to be biological.
- **Technical reads** The number of technical reads per spot. If there is no read type information for that run in SRA, the value will be 0.
- **Read orientation** Relevant for paired reads. **Unknown** means there is no orientation information for that run in SRA. This is always the case for single end reads, but is also frequently the case for paired reads. For such paired end runs, Forward-Reverse orientation is assumed by default when importing, but this is configurable.
- **Average length** The average length of all the reads in a spot combined. The read lengths in the imported sequence list may differ from these values if you choose not to download all the reads available for the run. E.g. downloading just biological reads when technical reads are also available.
- **PubMed** If a PubMed entry is associated with the run, it is listed and hyperlinked to the relevant Pubmed webpage.

When a run is selected in the table, the title and abstract for the SRA experiment it is part of is displayed in the SRA Preview tab, under the column configuration section of the side panel.

Please refer to the SRA documentation at the NCBI for full information on the data and metadata available <https://www.ncbi.nlm.nih.gov/sra/>.

Further details about download and import of data from SRA, including information on file sizes and paired read handling, is provided in section 10.4.2.

The total number of experiments found is reported at the bottom of the search table. An experiment may have more than one run associated with it.

By default, up to 50 results are retrieved at a time. Click on the **more...** button below the table to pull additional results, 50 at a time. This number can be configured in Preferences:

Edit | Preferences (⚙️) | General | Number of hits (NCBI/Uniprot)

Right-click on a row in the results table to get a list of possible additional searches, based on the selected run (figure 10.6).

10.4.2 Downloading reads and metadata from SRA

At the bottom of the SRA results table are two buttons: **Download Reads and Metadata** and **Show Metadata for Selection**. The functionality of each is described in this section.

Download Reads and Metadata

Select the rows of interest in the results table and then click on **Download Reads and Metadata** to download and import reads and metadata from the selected runs.

The screenshot displays the SRA Search interface. The main window shows a search result table for "Plasmodium vivax" from Vietnam. The table has 9 columns: #, Run accession, Study accession, Scientific name, Download..., Paired, Read orientation, Average l..., Spots, and PubMed. The first 32 rows are visible, with the 8th row highlighted. The right sidebar contains "Search Parameters Settings" with a "Show column" list and checkboxes for various filters. The "Download Reads and Metadata" button is at the bottom left.

#	Run accession	Study accession	Scientific name	Download...	Paired	Read orientation	Average l...	Spots	PubMed
1	SRR20306012	SRP387013	Plasmodium vivax	702	Yes	Unknown	302	6,685,718	
2	SRR20306001	SRP387013	Plasmodium vivax	504	Yes	Unknown	302	4,758,485	
3	SRR20305990	SRP387013	Plasmodium vivax	419	Yes	Unknown	302	4,020,451	
4	SRR20305985	SRP387013	Plasmodium vivax	817	Yes	Unknown	302	7,741,230	
5	SRR20305974	SRP387013	Plasmodium vivax	730	Yes	Unknown	302	6,959,407	
6	SRR20306013	SRP387013	Plasmodium vivax	628	Yes	Unknown	300	4,913,590	
7	SRR20306011	SRP387013	Plasmodium vivax	564	Yes	Unknown	300	4,396,480	
8	SRR20306010	SRP387013	Plasmodium vivax	361	Yes	Unknown	300	2,774,870	
9	SRR20306009	SRP387013	Plasmodium vivax	279	Yes	Unknown	300	2,129,599	
10	SRR20306008	SRP387013	Plasmodium vivax	257	Yes	Unknown	300	1,996,905	
11	SRR20306007	SRP387013	Plasmodium vivax	395	Yes	Unknown	300	3,095,658	
12	SRR20306006	SRP387013	Plasmodium vivax	204	Yes	Unknown	300	1,539,554	
13	SRR20306005	SRP387013	Plasmodium vivax	359	Yes	Unknown	300	2,719,350	
14	SRR20306004	SRP387013	Plasmodium vivax	566	Yes	Unknown	300	4,321,165	
15	SRR20306003	SRP387013	Plasmodium vivax	296	Yes	Unknown	300	2,235,416	
16	SRR20306002	SRP387013	Plasmodium vivax	455	Yes	Unknown	300	3,430,808	
17	SRR20306000	SRP387013	Plasmodium vivax	265	Yes	Unknown	300	2,015,514	
18	SRR20305999	SRP387013	Plasmodium vivax	583	Yes	Unknown	300	4,610,619	
19	SRR20305998	SRP387013	Plasmodium vivax	334	Yes	Unknown	300	2,509,850	
20	SRR20305997	SRP387013	Plasmodium vivax	746	Yes	Unknown	300	5,464,768	
21	SRR20305996	SRP387013	Plasmodium vivax	347	Yes	Unknown	300	2,677,877	
22	SRR20305995	SRP387013	Plasmodium vivax	604	Yes	Unknown	300	4,707,505	
23	SRR20305994	SRP387013	Plasmodium vivax	501	Yes	Unknown	300	3,937,575	
24	SRR20305993	SRP387013	Plasmodium vivax	479	Yes	Unknown	300	3,707,176	
25	SRR20305992	SRP387013	Plasmodium vivax	584	Yes	Unknown	300	4,578,022	
26	SRR20305991	SRP387013	Plasmodium vivax	653	Yes	Unknown	300	5,050,079	
27	SRR20305989	SRP387013	Plasmodium vivax	662	Yes	Unknown	300	5,285,946	
28	SRR20306045	SRP387013	Plasmodium vivax	657	Yes	Unknown	300	5,037,455	
29	SRR20305987	SRP387013	Plasmodium vivax	107	Yes	Unknown	300	786,162	
30	SRR20305986	SRP387013	Plasmodium vivax	444	Yes	Unknown	300	3,352,349	
31	SRR20305984	SRP387013	Plasmodium vivax	569	Yes	Unknown	300	4,289,379	
32	SRR20305983	SRP387013	Plasmodium vivax	641	Yes	Unknown	300	4,844,164	

Figure 10.6: The SRA search result table.

Reads are imported into sequence lists. Import settings for reads from runs marked as paired are configurable, including the option to import technical reads in addition to biological reads.

Metadata is imported into a CLC Metadata Table. Each sequence list will have an association to the relevant row of the CLC Metadata Table. See section 13.3.1 for details about data associations with CLC Metadata Tables. The CLC Metadata Table can be used directly to define the experimental design for differential expression analyses (section 33.6.4) or edited, if desired (section 13.3.5).

Note: When the "Auto paired end distance detection" option is present in downstream analyses of paired data downloaded from SRA, we recommend it is enabled. This is because some SRA entries have an insert size that includes the length of the reads, while others exclude the length of the reads.

After clicking on **Download Reads and Metadata**, a wizard appears to guide you through the import of the selected runs.

Import Options

- **Discard read names** Check this option to save disk space. Individual read names are rarely relevant for NGS data.
- **Discard quality scores** Checking this option can reduce disk space usage and memory consumption. This should only be done if these scores are not relevant to your work. Quality scores are not used for RNA-Seq and expression analyses, but are used during variant detection and can be shown in views of read mappings.

Space requirements

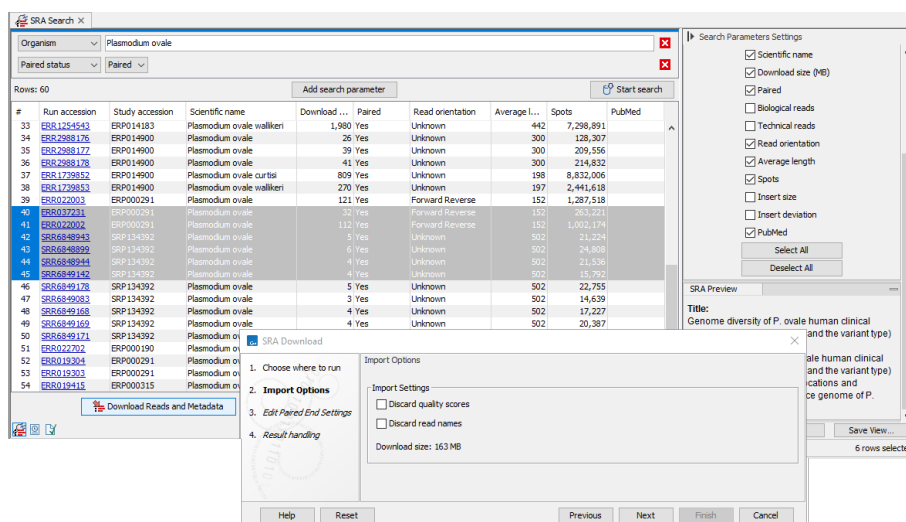


Figure 10.7: Import options in the SRA Download wizard

During download and import: The **Download size** reported is the combined size of all the SRA format files that will be retrieved.

We recommend that at least twice the download size of the largest sample is available as temporary space during download and import.

If the SRA file is reference-compressed, a copy of the genome must also be retrieved before the reads can be imported, which will also require disk space.

For the imported data: The size of the sequence lists after import will often be comparable in size to the SRA files downloaded (often between half to twice the size). The size depends on multiple factors, including whether compression has been turned off, whether read names and quality scores were retained, and whether you imported technical reads as well as biological reads, where relevant.

A few examples of SRA file sizes relative to imported sequence list sizes are given below. Relative sizes may differ on your system depending on your settings.

Description	SRA file	After import, with read names and qual. scores	After import, no read names or quality scores
Single end reads	84 MB	110 MB	32 MB
Paired end biological reads	107 MB	138 MB	59 MB
2 technical reads and 1 biological read, only the biological imported	1311 MB	760 MB	208 MB

Edit Paired End Settings If at least one of the selected runs is marked as paired, the next wizard step allows you to review and edit the paired end settings (figure 10.8).

Values in shaded cells can be configured by selecting rows and clicking on the "Edit Selected Rows" button. The settings in the edit dialog when you click on **OK** are applied to all the selected rows, so we recommend selecting either a single row, or sets of rows where the information should be the same.

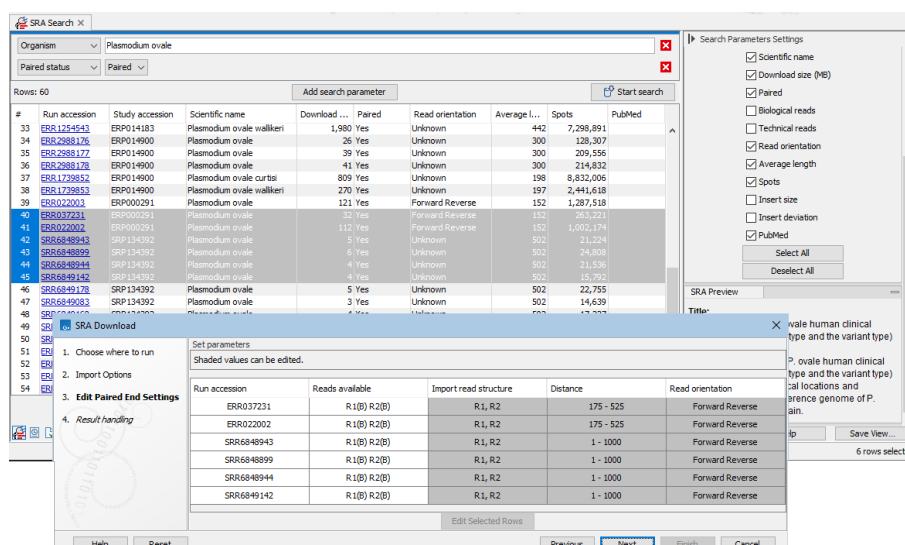


Figure 10.8: Paired end information includes the reads available for that run, as well as the read structure, distance and read orientation. Values in shaded cells can be configured by selecting rows and clicking on the "Edit Selected Rows" button.

- **Reads available** Most entries consist of two reads, both biological (figure 10.8). These are presented as R1 (B) R2 (B), where B stands for "biological".

When there are 3 or more reads, 1 or 2 of these are expected to be biological. E.g. I1 (T) R1 (B) R2 (T) R3 (B) (figure 10.10).

Mouse over a cell in the **Reads available** column to see details about the reads, including the number of reads, their average length and the standard deviation of the length (figure 10.11).

- **Import read structure** Importing just biological reads is the default. To configure this setting, specify the reads to import using the names given in the "Reads available" column, e.g. I1, R1, etc. Use a space to separate reads that should be concatenated on import. Use a comma to separate reads to import as the first sequence in a pair and reads to import as the second sequence in a pair.

For example:

- R2, R1 A sequence list containing paired reads is imported. The first member of each pair is from R2 and the second from R1.
- I1 R1 R2 A sequence list containing single reads is imported. Each sequence in the list is a concatenation of I1, R1 and R2, in that order (figure 10.12).
- R2 R1, R3 A sequence list containing paired reads is imported. The first member of each pair is a concatenation of R2 and R1, in that order. The second member is the corresponding R3 read. This could represent a situation where R1 contains forward reads, R3 contains reverse reads, and R2 contains molecular indices.

If "Use SRA Defaults" appears in this column, we recommend explicitly defining the read structure. See section 10.4.3 for further details.

- **Distance** The minimum and maximum distance depend on whether an "Insert Size" and an "Insert Deviation" were supplied to SRA by the depositor.

- If no insert size was supplied, we set the minimum to 1 and the maximum to 1,000.
- If an insert size was supplied, we do the following calculation:
 - * $Min\text{distance} = \text{insertsize} - 5 * \text{insertdeviation}$
 - * $Max\text{distance} = \text{insertsize} + 5 * \text{insertdeviation}$
- If no deviation was supplied, we estimate this to be $0.1 * \text{insertsize}$ and perform the same calculation as above.

- **Read orientation** The orientation to use when importing paired reads.

For runs marked as paired with 2 biological reads per spot and no read structure information, we use the SRA default, which is to handle the reads as Forward Reverse pairs (figure 10.8).

N/A values in the Distance and Read orientation columns are expected when only one of the reads is biological (figure 10.9). If the read structure is edited such that paired reads will be imported, values will appear in these columns.

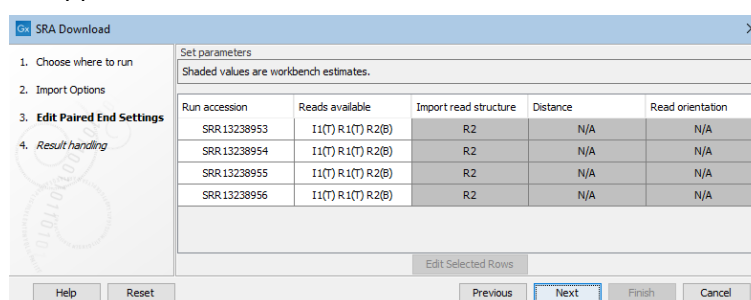


Figure 10.9: Only R2 is biological, so by default a sequence list containing single sequences from R2 would be imported.

When the settings match your expectations, click on **Next** to select where to save the data, and then start the download.

Show Metadata for Selection

Information about SRA entries of interest can be downloaded to a CLC Metadata Table without downloading the sequence data. Select the rows of interest in the results table and then click on **Show Metadata for Selection**. Sequence data can be downloaded later if desired.

The first columns of the resulting CLC Metadata Table contain the same database identifiers as in the original results table. Later columns contain details associated with the biosample, which are pulled from SRA. In the side panel, to the right, the columns to show in the table can be configured.

Tips relating to retrieving sequence data later using the CLC Metadata Table:

- CLC Metadata Tables can be filtered so that only relevant rows are shown (see section 9.2).
- To copy just the accessions from the visible rows in the table, and retrieve these entries from SRA:
 - Select only the "Run Accession" column in the side panel settings. (It can be fastest to click on the "Deselect All" button at the bottom of the column listing in the side panel and then re-select Run Accession.)

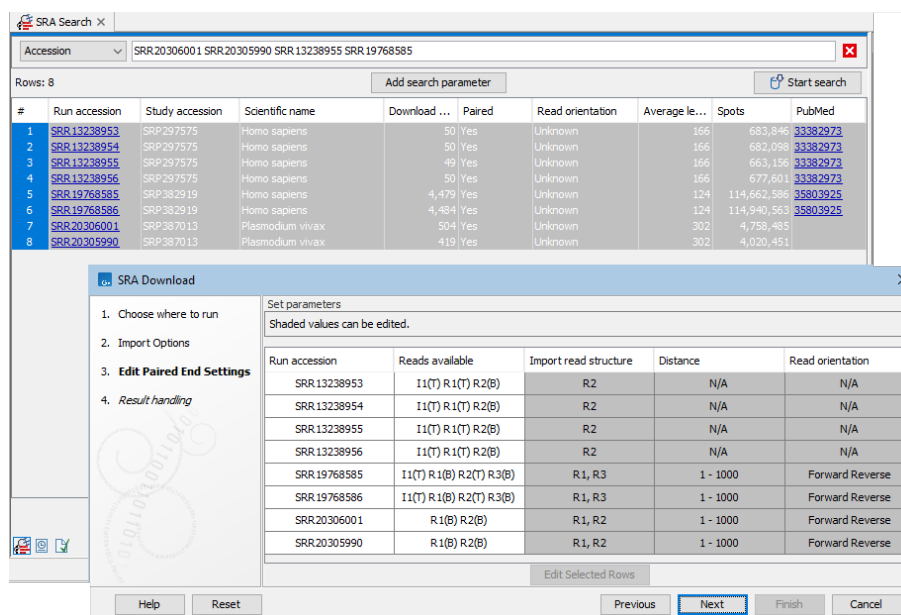


Figure 10.10: Paired end information includes the reads available for that run, as well as the read structure, distance and read orientation. Reads specified as technical by the SRA submitter are marked with (T), while other reads are marked with (B) for biological. The first 4 entries listed in the SRA Download wizard are examples for runs marked as paired with only one read specified as biological. These are imported as single end reads by default.

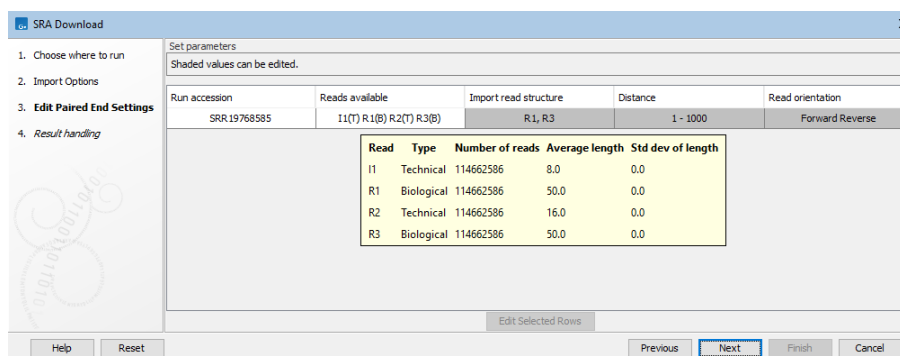


Figure 10.11: Mousing over an entry in the Reads available column in the Edit Paired End Settings wizard step reveals a tooltip with details for each read in that run.

- Select all the rows and then go to Edit | Copy in the top level menu (or use Ctrl + C).
- In **Search for Reads in SRA**, choose "Accession" in the search options area and paste the copied accessions into that field using Edit | Paste from the top level menu (or use Ctrl + V).
- Remove the text "Run Accession" from the start of the pasted text.
- Run the search at SRA.
- Downloading reads using **Search for Reads in SRA** creates a new CLC Metadata Table with the resulting sequence lists associated to the relevant rows. Sequence lists can be associated with any CLC Metadata Table you wish. For more details, see section 13.2.

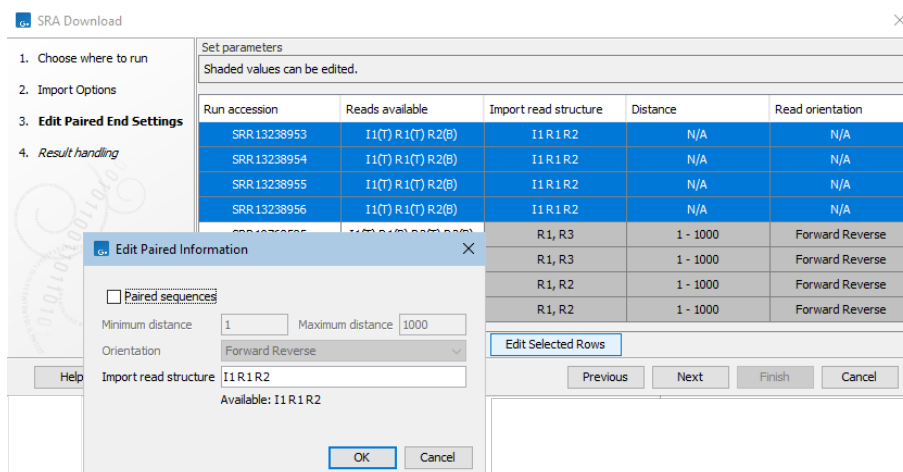


Figure 10.12: The read structure for import has been edited in the first 4 entries listed in the SRA Download wizard, using the settings shown in the Edit Paired Information dialog. Technical reads from I1 and R1 will be prepended to R2 reads and these sequences imported into a single end sequence list.

10.4.3 Troubleshooting SRA downloads

General

1. Reads cannot be downloaded from SRA

If reads cannot be downloaded for certain SRA runs, the runs are listed in a Problems panel together with a description of the problem. It is usually still possible to download reads for the remaining runs.

The most common problems are:

- "The selected SRA reads contain no spots, and cannot be imported in the workbench.": The run has no associated sequencing data.
- "The selected SRA reads are dbGaP restricted.": For data protection reasons, you must request access to these reads. Requests and download cannot happen within the workbench, but you can follow the procedures here: <https://www.ncbi.nlm.nih.gov/books/NBK5295/>.
- "The selected SRA reads are made with an unsupported sequencing platform.": For example, Complete Genomics reads consist of eight regions separated by gaps of variable lengths, and should be analyzed by specialist tools.

2. No values in the Biological reads or Technical reads columns of the results table

If there are no values in the Biological reads or Technical reads columns, the SRA entry may contain inconsistent information in an SRA entry. Downloading such entries is usually possible, but what is downloaded will depend on the circumstances.

For example, if a run has more than one read per spot, but is not marked as paired, the Biological reads and Technical reads columns will be blank. Downloading this data will result in a sequence list containing single reads from R1. This would be fine in a case like SRR16530746, where R1 contains the read information (and R2 contains no bases). However, it may not be fine for other entries.

In such cases, we recommend checking the imported sequence list contains the expected data.

Paired end settings table

1. "?" in Reads Available column and "Use SRA Defaults" in Import Read Structure column

When there are more than 2 reads available for a run marked as paired, but no information about the read structure is available, "?" appears beside each read set in the "Reads Available" column, and "Use SRA Defaults" appears in the Import Read Structure column. If not configured further, reads that SRA defines as biological by default are imported.

In many cases, this is fine, but we recommend explicitly defining the read structure. If you do not, please check the resulting sequence lists to ensure the expected data is present. Click on a run accession in the results table to go directly to the SRA webpage about that run.

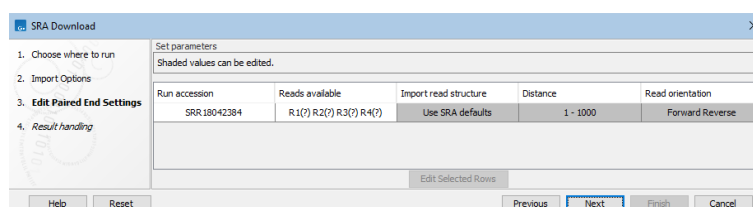


Figure 10.13: This run contains more than 2 reads per spot but no explicit read structure information. Unless configured further, reads will be imported according to SRA default handling, with only reads SRA interprets as biological being imported.

2. "Use SRA defaults" warning under Import read structure column

In the "Edit Paired End Settings" step of the SRA Download wizard, text in the "Import read structure" column will state "Use SRA defaults" and include a warning exclamation mark if the nature of the reads in SRA does not match what is expected. For example, a run is not expected to have three or more biological reads. (figure 10.14).

If you choose to import data from such runs, we recommend the read structure is explicitly defined, as described in section 10.4.2. If it is not, biological reads will be imported using default settings. This may or may not result in a sequence list containing data relevant for downstream processing.

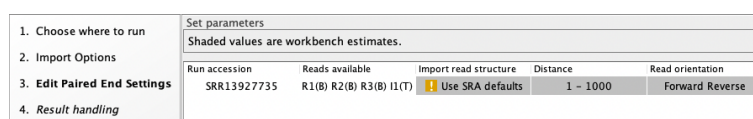


Figure 10.14: This run has 3 biological reads, which is not expected. A warning icon is shown in the Import read structure column. If the data needs to be downloaded, then explicitly defining the read structure in a case like this is recommended.

10.5 Sequence web info

CLC Genomics Workbench provides direct access to web-based search in various databases and on the Internet using your computer's default browser. You can look up a sequence in the databases of NCBI and UniProt, search for a sequence on the Internet using Google and

search for Pubmed references at NCBI. This is useful for quickly obtaining updated and additional information about a sequence.

The functionality of these search functions depends on the information that the sequence contains. You can see this information by viewing the sequence as text (see section 15.5). In the following sections, we will explain this in further detail.

The procedure for searching is identical for all four search options (see also figure 10.15):

Open a sequence or a sequence list | Right-click the name of the sequence | Web Info (🌐) | select the desired search function

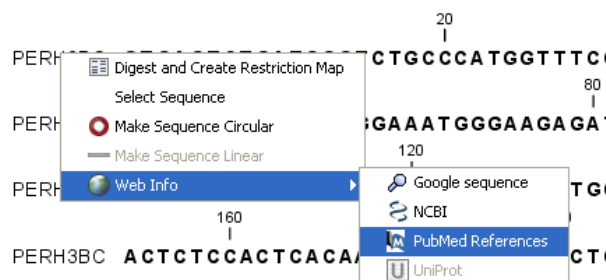


Figure 10.15: Open webpages with information about this sequence.

This will open your computer's default browser searching for the sequence that you selected.

Google sequence The Google search function uses the accession number of the sequence which is used as search term on <https://www.google.com>. The resulting web page is equivalent to typing the accession number of the sequence into the search field on <https://www.google.com>.

NCBI The NCBI search function searches in GenBank at NCBI (<https://www.ncbi.nlm.nih.gov>) using an identification number (when you view the sequence as text it is the "GI" number). Therefore, the sequence file must contain this number in order to look it up at NCBI. All sequences downloaded from NCBI have this number.

PubMed References The PubMed references search option lets you look up Pubmed articles based on references contained in the sequence file (when you view the sequence as text it contains a number of "PUBMED" lines). Not all sequence have these PubMed references, but in this case you will see a dialog and the browser will not open.

UniProt The UniProt search function searches in the UniProt database (<https://www.uniprot.org/>) using the accession number. Furthermore, it checks whether the sequence was indeed downloaded from UniProt.

Additional annotation information When sequences are downloaded from GenBank they often link to additional information on taxonomy, conserved domains etc. If such information is available for a sequence it is possible to access additional accurate online information. If the db_xref identifier line is found as part of the annotation information in the downloaded GenBank file, it is possible to easily look up additional information on the NCBI web-site.

To access this feature, simply right click an annotation and see which databases are available. For tracks, these links are also available in the track table.

Chapter 11

References management

Contents

11.1 Download Genomes	218
11.2 QIAGEN Sets	221
11.3 Reference Data Sets and defining Custom Sets	224
11.3.1 Copy to References	228
11.3.2 Export a Custom Data Set	229
11.3.3 Import a Custom Data Set	231
11.4 Storing, managing and moving reference data	233
11.4.1 Imported Data	236
11.4.2 Exporting reference data outside of the Reference Data Manager framework	236

Management of reference data is primarily done through the Reference Data Manager, which supports the download and management of reference data provided by QIAGEN and some other public resources (figure 11.1).

Data downloaded using the Reference Data Manager can only be deleted using Reference Data Manager functionality. If the data is being stored on a CLC Server, it can only be deleted by administrative users.

The Reference Data Manager

To open the Reference Data Manager, click on the **Manage Reference Data** (📁) button in the top Toolbar or go to the **Utilities** menu and select **Manage Reference Data** (📁).

Tabs at the top left of the Reference Data Manager provide access to different data resources and related functionality, which are the focus of the rest of this chapter:

- **Download Genomes** Download reference data from public repositories such as Ensembl, NCBI, and UCSC. See section 11.1.
- **QIAGEN Sets** Download reference data from the QIAGEN repository, including references relevant for analyzing QIaseq panels and kits, using QIAGEN supplied workflows. Data is available as individual elements, or grouped into sets of related elements referred to as **Reference Data Sets**. See section 11.2.

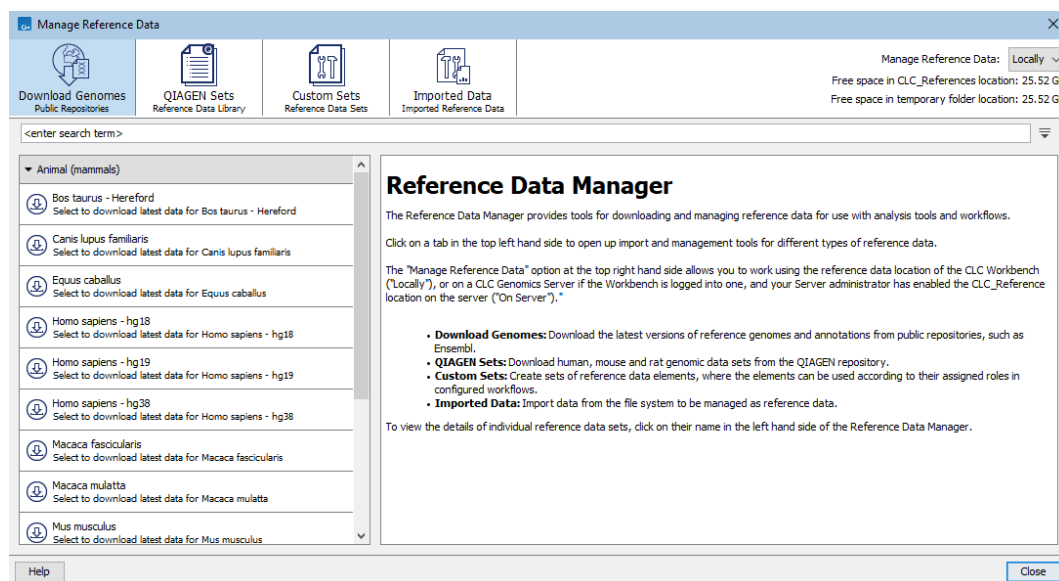


Figure 11.1: The Reference Data Manager. Tabs at the top left provide access to different data resources and related functionality. Terms entered in the search field are used to find matches among elements or data sets available under the selected tab.

- **Custom Sets** Functionality to create your own Reference Data Sets, and to work with custom sets already made. See section 11.3.
- **Imported data** Functionality to move data under the control of the Reference Data Manager. See section 11.4.1.

Configuration options relating to where data downloaded using the Reference Data Manager is stored are described in section 11.4.

11.1 Download Genomes

Download Genomes provides access to genomes and associated reference data, such as annotations and known variants, from public repositories (figure 11.2)¹.

To access **Download Genomes**, open the Reference Data Manager by clicking on the **Manage Reference Data** (🔗) button in the top Toolbar or go to the **Utilities** menu and select **Manage Reference Data** (🔗). Then click on the **Download Genomes** tab at the top, left.

Click on an organism name to select it. Information about the data available for download for that organism is then shown in the right hand panel, including a name describing the data, the data provider, a version (if available), and the size (figure 11.2).

Searching for data available under the Download Genomes tab

Use the search field under the top toolbar to search for terms in the names of organisms, resources and resource providers. To search for just an exact term, put the term in quotes.

¹The data listed under the Download Genomes tab is not provided by nor hosted by QIAGEN.

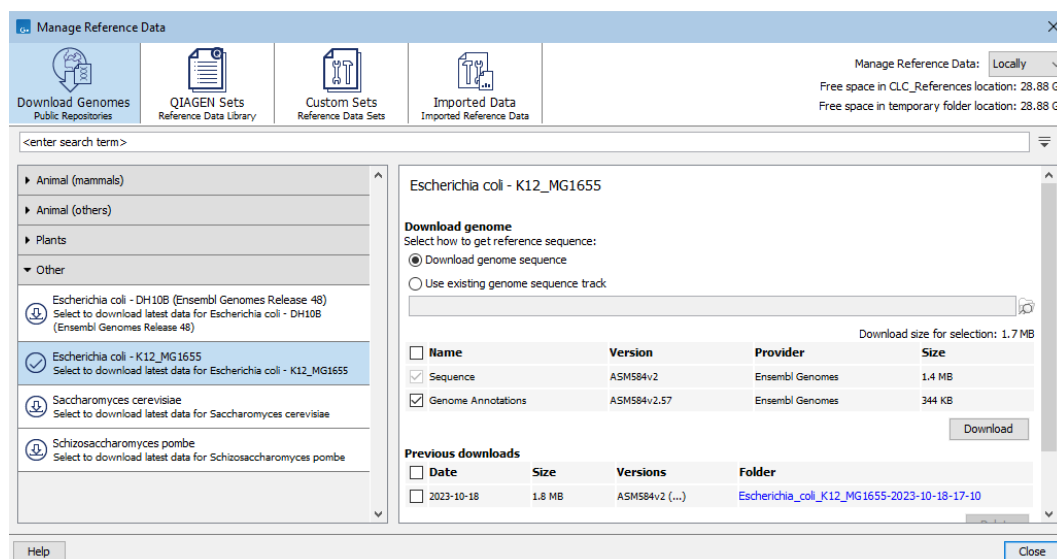


Figure 11.2: Under the Download Genomes tab, reference sequences and associated reference data for a variety of organisms can be downloaded from public repositories.

The results include the name of the element or set the term was found in, followed in brackets by the tab it is listed under. Hover the cursor over a hit to see what aspect of the result matched the search term (figure 11.3). Double-click on a search result to open it.

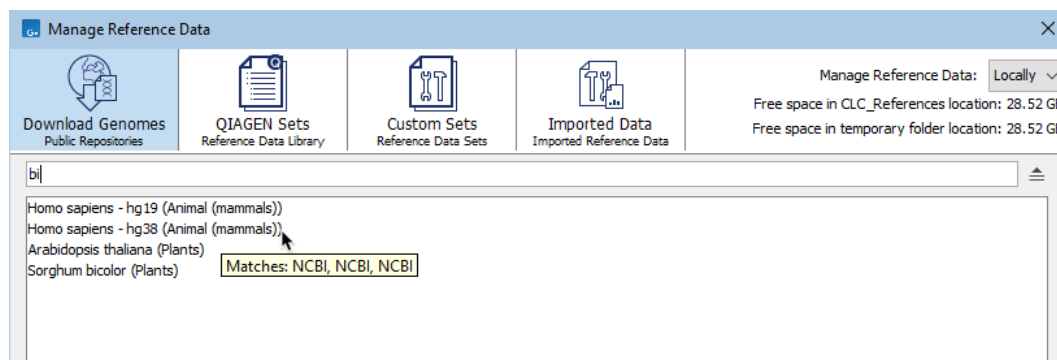


Figure 11.3: When the Download Genomes tab is selected, terms entered in the search field are searched for in the names of organisms, resources and resource providers. Hovering the cursor over a hit reveals a tooltip with information about the match.

Downloading resources

To download resources, select the data types of interest and click on the Download button. "Sequence" is always selected.

When reference data is stored on a CLC Server with grid nodes, the grid preset to use to download data can be specified via a drop-down menu to the left of the **Download** button (figure 11.4).

After download, each file is imported and stored under the CLC_References File System Location. A folder for each downloaded set is created under the "Genomes" folder. Its name contains the species name and the date of the download.

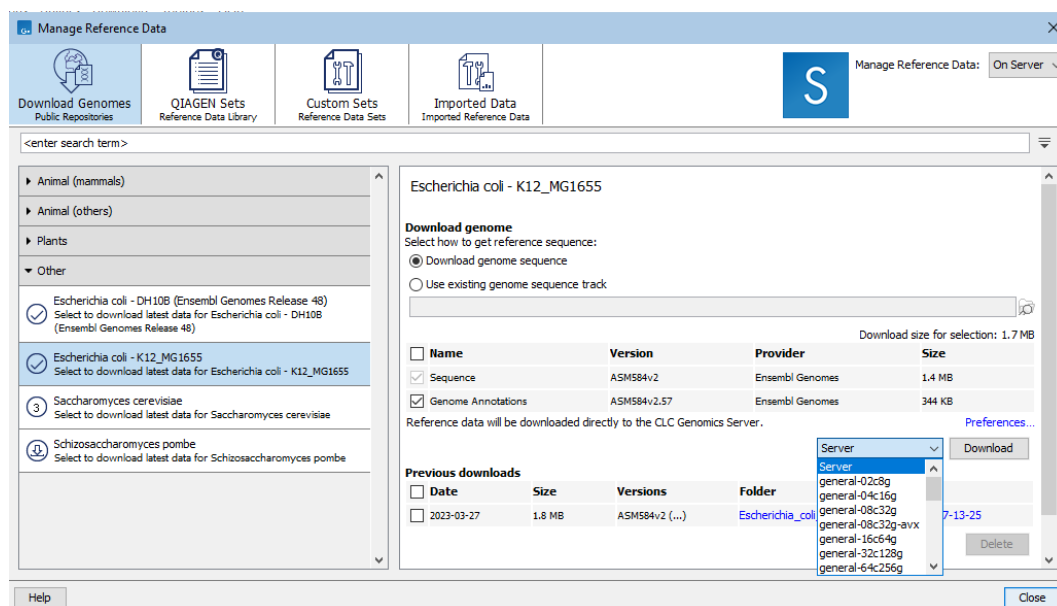


Figure 11.4: When the "On Server" option is selected and grid presets are configured on the CLC Server, a grid preset to use for the download can be selected.

Previous downloads of data for the selected organism are listed in the right hand panel of the Reference Data Manager under "Previous downloads".

To delete downloaded data, select the entries in this list and then click on the Delete button. When reference data is stored on a CLC Server, you need be logged in from the Workbench as an administrative user to delete reference data.

Note: Most data is supplied as a compressed text file. After download, each file is decompressed and the data is imported. CLC data is compressed by default, but the size of the compressed data after import will generally be different to the size reported for the original data file.

Notes about particular data types

When GFF3 files are imported, a track is created for each feature type present in the file (see section 7.2.1). In addition, an (RNA) track and a (Gene) track are created. The (RNA) track contains entries for all "RNA" type annotations. I.e. all the children of "mature_transcript", which is the parent of "mRNA", which is the parent of the "NSD_transcript". The (Gene) track contains genes and gene-like annotation types, such as ncRNA_gene, plastid_gene, and tRNA_gene. These broader sets of annotations can make these tracks particularly useful for some types of analyses, e.g. RNA-Seq.

For some genomes, chromosome bands (ideograms) are available (figure 11.5).

Please note that hg18 and hg19 variants downloaded from UCSC do not include variants on the mitochondrial genome.



Figure 11.5: The ideogram is particularly useful when used in combination with other tracks in a track list. In this figure the ideogram is highlighted with a red box.

11.2 QIAGEN Sets

QIAGEN provides access to much common reference data using functionality under the **QIAGEN Sets** tab of the Reference Data Manager. Data is distributed as **Reference Data Elements**, which can be individually downloaded, or downloaded as part of a **Reference Data Set**. Many template workflows are configured to make use of QIAGEN Reference Sets, making them simple to launch while helping to ensure that the same reference data is used consistently. Using such workflows, the relevant reference data can also be downloaded via the workflow launch wizard (figure 11.6). When logged into a CLC Server with a CLC_References location defined, you can choose whether to download the data to the Workbench or Server.

Using the Reference Data Manager for QIAGEN reference data

To access **QIAGEN Sets**, open the Reference Data Manager by clicking on the **Manage Reference Data** (📁) button in the top Toolbar or go to the **Utilities** menu and select **Manage Reference Data** (📁). Then click on the **QIAGEN Sets** tab at the top left. Under this tab, there are subsections for **Reference Data Sets** and **Reference Data Elements** (figure 11.7).

When a Reference Data Set is selected, information about it is displayed in the right hand pane. This includes the size of the whole data set, and a table listing the workflow roles defined in the set, with information about the data element specified for each role. Further details about the

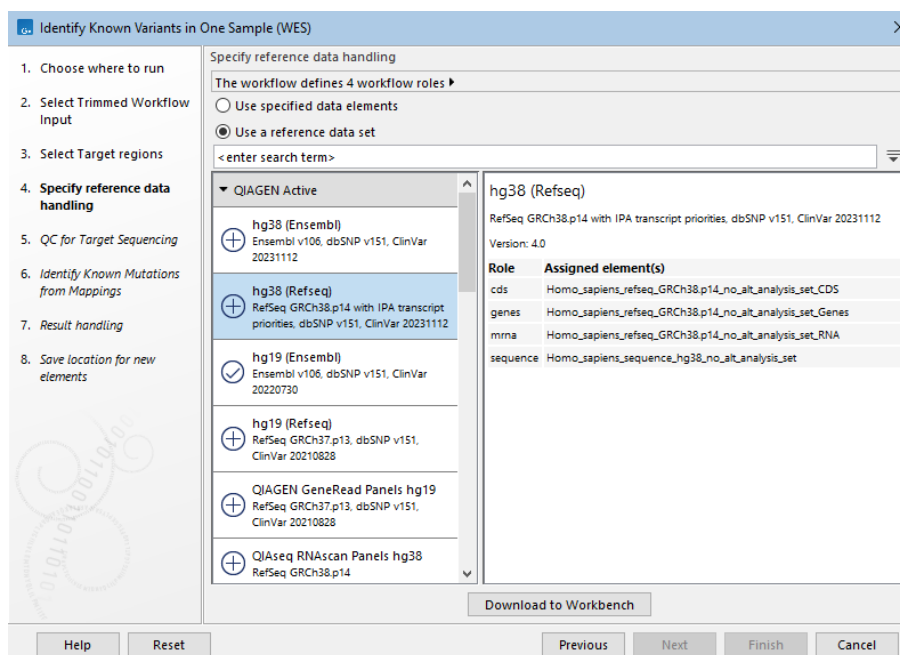


Figure 11.6: When launching workflows configured to use data from Reference Data Sets, the relevant reference data can be downloaded via the workflow launch wizard.

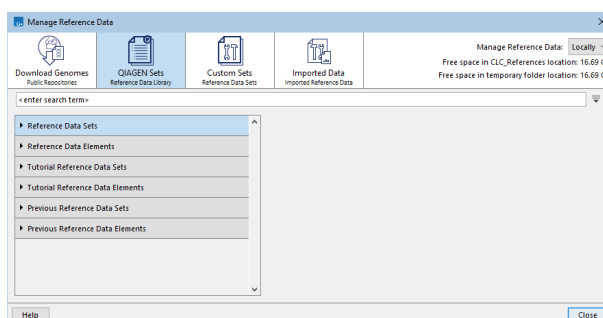


Figure 11.7: Subheadings under the QIAGEN Sets tab provide access to Reference Data Sets and Reference Data Elements

element assigned to a role can be found by clicking on the link in the Version column. An icon to the left of each set indicates whether data for this set has already been downloaded (☑) or not (⬇). The same icons are used to indicate the status of each element in a Reference Data Set (figure 11.8).

If you have permission to delete downloaded data, the **Delete** button will be enabled. When reference data is stored on a CLC Server, you need be logged in from the Workbench as an administrative user to delete reference Sets data.

Searching for data available under the QIAGEN Sets tab

Use the search field under the top toolbar to search for terms in element and set names, workflow role names, and versions. To search for just an exact term, put the term in quotes.

The results include the name of the element or set the term was found in, followed in brackets

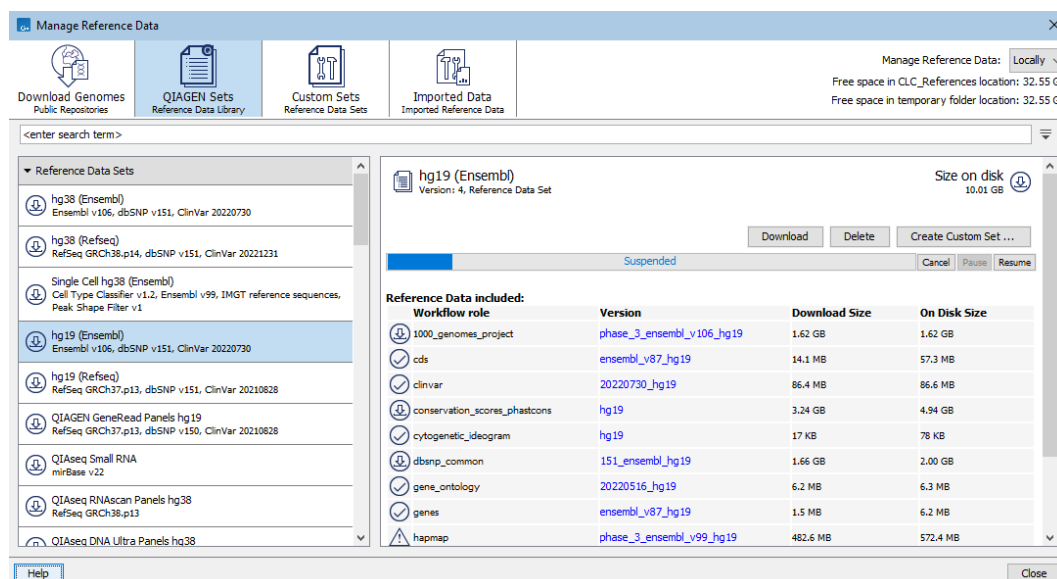


Figure 11.8: The elements in a Reference Data Set are being downloaded. The full size of the data set is shown at the top, right hand side. The size of each element is reported in the "On Disk Size" column. Below the row of tabs at the top is a search field that can be used to search for data sets or elements.

by the tab it is listed under, e.g. (Reference Data Elements), (Tutorial Reference Data Sets), etc. Hover the cursor over a hit to see what aspect of the result matched the search term (figure 11.9). Double-click on a search result to open it.

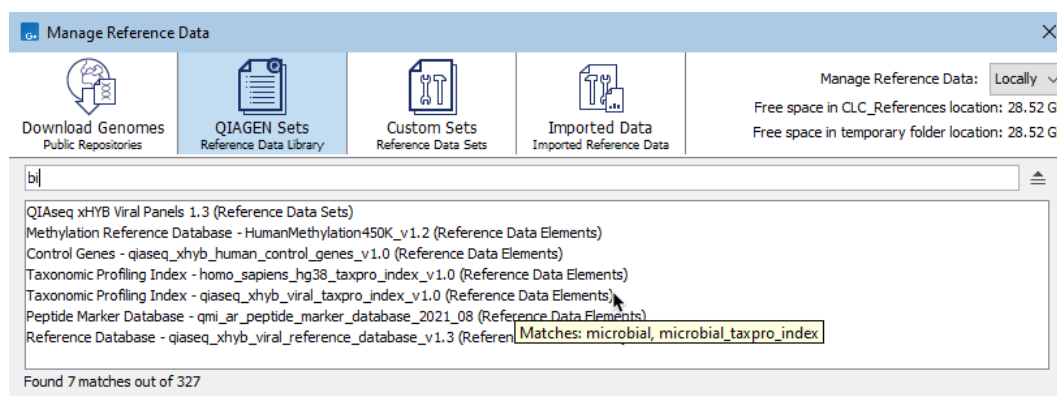


Figure 11.9: Terms entered in the search field when the QIAGEN Sets tab is selected are searched for in element and set names, workflow role names, and versions of the resources available under that tab. Hovering the cursor over a hit reveals a tooltip with information about the match.

Downloading resources

To download a Reference Data Element or a Reference Data Set (i.e. all elements in that set), select it and click on the **Download** button.

The progress of the download is indicated and you have the option to **Cancel**, **Pause** or **Resume** the download (figure 11.8).

When the "Manage Reference Data" option at the top of the Reference Data Manager is set to "Locally", data is downloaded to the CLC_References location in the *CLC Workbench*. When set to "On Server", the data is downloaded to the CLC_References location in the *CLC Server*.

Additional information

The HapMap (<https://www.sanger.ac.uk/data/hapmap-3/>) databases contain more than one file. QIAGEN Reference Data Sets that include HapMap are initially configured with all the populations available. You can specify specific populations to use when launching a workflow, or you can create a custom reference set that contains only the populations of interest.

General information about Reference Data Sets, and creating Custom Sets, can be found at section [11.3](#).

11.3 Reference Data Sets and defining Custom Sets

General information about Reference Data Sets

A Reference Data Set is a collection of related reference data elements. By configuring a workflow to make use of a Reference Data Set, a set of connected reference data can be specified in a single step when launching the workflow, instead of selecting each data element individually.

The connection between the data available in a Reference Data Set and the data needed for a workflow run is made by matching up *roles*:

- Each reference data element in a Reference Data Set has a *role*.
- Each workflow input that expects reference data can be configured with a *workflow role*.

By matching up the *roles* in a Reference Data Set with the *workflow roles* configured in a workflow, relevant Reference Data Sets can be offered when the workflow is launched.

Many template workflows, delivered with QIAGEN CLC software, are configured with workflow roles. QIAGEN Reference Data Sets and any Custom Sets that contain the corresponding roles are thus available for selection when launching these template workflows.

Any single word can be used as a role/workflow role. QIAGEN Reference Data Sets usually use a term describing the data type, such as "sequence", "gene", "mrna", "cds", etc.

For information about configuring reference data inputs in workflows, see section [14.2.3](#).

Managing Reference Data Sets

Reference Data Sets are downloaded, managed and created using the Reference Data Manager. Reference Data Sets containing some commonly used reference data are available for download under the QIAGEN Sets tab of the Reference Data Manager (see section [11.2](#)). It is also easy to create new sets, known as Custom Sets, that refer to the data of your choice, using roles defined by you.

Creating Custom Sets

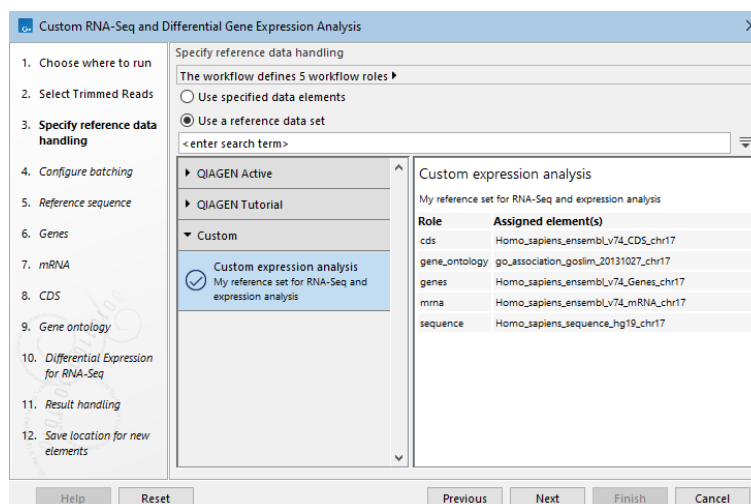


Figure 11.10: Reference Data Sets containing all the workflow roles specified in a workflow are available for selection in the launch wizard, including Custom Sets.

To create a Custom Set, you can:

Base it on an existing Reference Data Set To do this, select a Reference Data Set and click on the **Create Custom Set...** button above the listing of data elements, on the right. This opens the "Create Custom Data Set" dialog, populated with the roles defined in the selected Reference Data Set, and any specified data elements (figure 11.11). This new set can then be customized.

Build it from scratch To start from scratch, click on the Custom Sets tab at the top of the Reference Data Manager and then click on the **Create** button on the right. This opens the "Create Custom Data Set" dialog without any roles or elements predefined (figure 11.12).

Base it on reference data used in a specific workflow To do this, open the "Create Custom Data Set" dialog using one of the methods described above, and then click on the **Add to Match Workflow...** button. You can specify an installed workflow from a drop-down list, or select a workflow from the Navigation Area using the "Workflow design" field (figure 11.13). If buttons are disabled, it usually means the selected workflow does not contain inputs defined with workflow roles.

When basing a Custom Set on an existing Reference Data Set or on the references defined in a workflow, any predefined data elements will be listed in the **Item(s)** column of the relevant roles. Data elements can be selected or updated by double-clicking on the cells in that column.

You can define new roles in Custom Sets, or assign roles already in use in existing Reference Data Sets (figure 11.14). Note that workflow role names cannot contain spaces.

Specifying multiple data elements with a single role

For certain types of reference data, it may be relevant to associate multiple data elements with a single role. For example, if a workflow will be used with the same reference data except for the target regions, it could be efficient to create a single Custom Set with the *target_regions* role linked to multiple elements. A Custom Set with multiple elements assigned to it is shown in

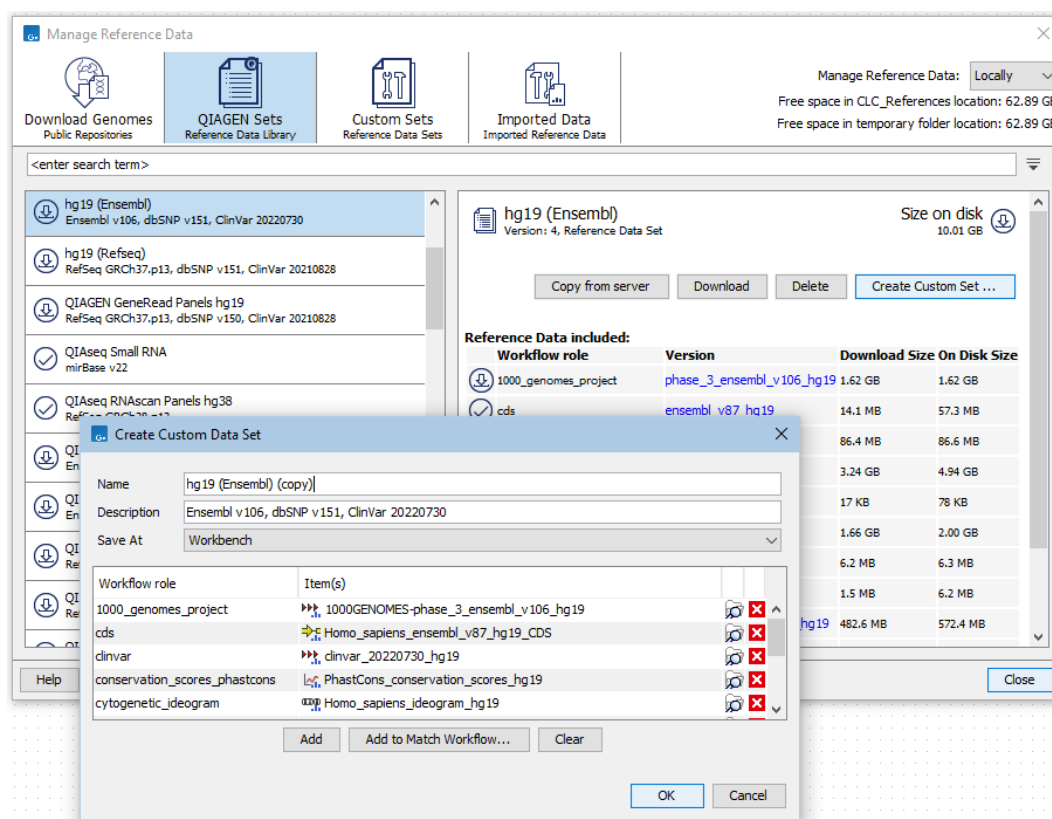


Figure 11.11: After selecting a QIAGEN Set, click on the Create Custom Set button on the right hand side to open the Create Custom Data Set dialog populated with the roles and elements of that reference set.

figure 11.16. Clicking on the arrow at the right hand side of the target_regions row reveals a list of the elements assigned to this role.

Working with Custom Sets

Once saved, the new Custom Set will be listed under the Custom Sets tab of the Reference Data Manager. These sets will also be available to select in launch wizards when:

- Launching workflows where a *workflow role* is specified. A dedicated wizard step is presented for specifying the Reference Data Set to use (figure 11.16).
- Launching tools or launching workflows where a *workflow role* is not specified for the input. In launch wizard data selection steps, click on the **Reference Data** tab to see reference data organized in their Reference Data Sets (figure 11.15).

When a Reference Data Set contains a role that has multiple data elements assigned to it, there will be an arrow beside that role name (figure 11.16). The list of those elements can be revealed by clicking on that arrow (figure 11.17). A subsequent launch wizards step allows the selection of the particular element that should be used for the workflow run (figure 11.18).

Searching for data available in Custom Sets

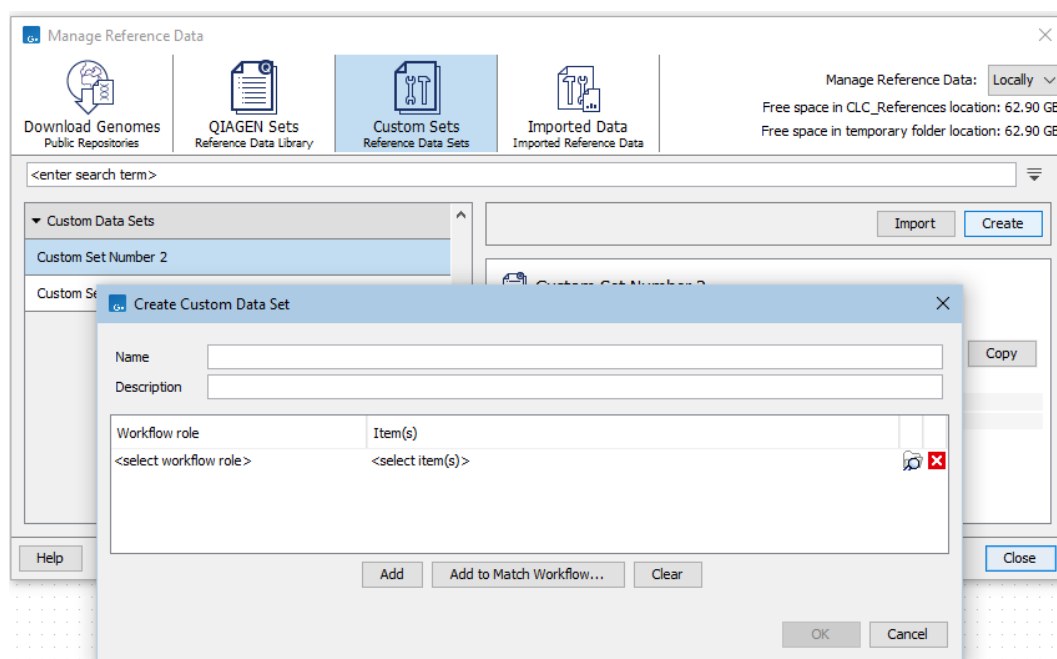


Figure 11.12: Under the Custom Sets tab, click on the Create button, on the right, to open the Create Custom Data Set dialog without any roles or elements predefined.

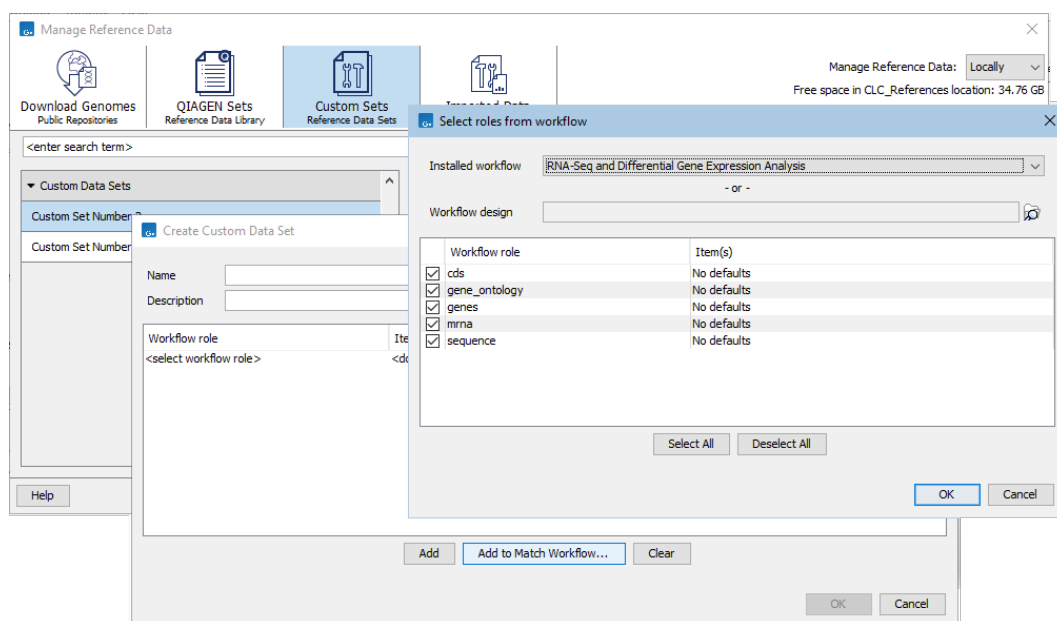


Figure 11.13: Click on the Add to Match Workflow button in the Create Custom Data Set dialog to populate the dialog with the roles and elements defined in a workflow.

Use the search field under the top toolbar in the Reference Data Manager to search for terms in Custom Sets. To search for just an exact term, put the term in quotes.

Hover the cursor over a hit to see what aspect of the result matched the search term (figure 11.19). Double-click on a search result to open it.

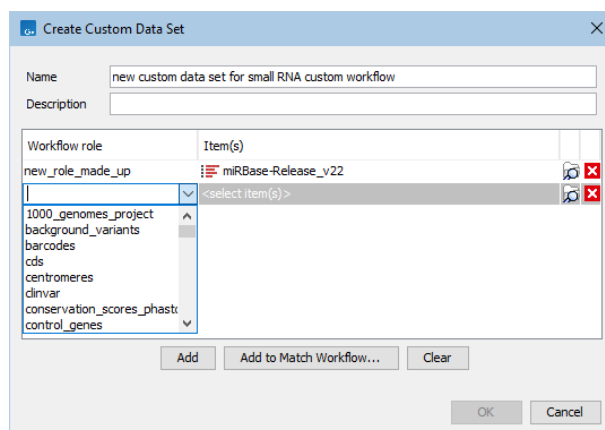


Figure 11.14: The Create Custom Sets dialog showing a newly created role, and the drop down menu of already existing roles.

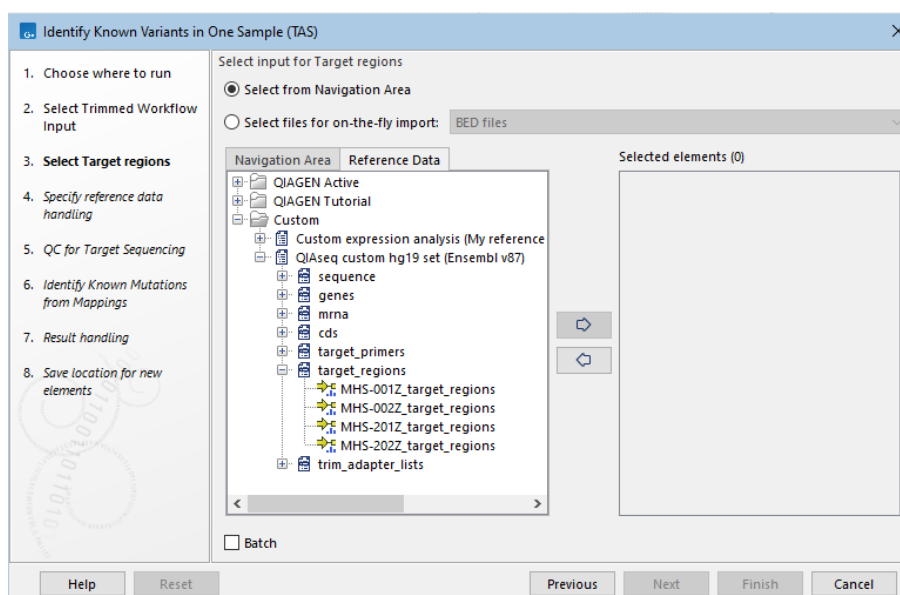


Figure 11.15: Data elements in Custom Sets can be found under the Reference Data tab in launch wizard steps where reference data elements need to be specified.

11.3.1 Copy to References

Any data element can be specified within a Custom Set. If desired, data specified in Custom Sets can be easily copied to a `CLC_References` location, where there is more control over data updates and deletion relative to data stored in other File System Locations (see section 11).

To copy data referred to from a Custom Set from the Navigation Area to a `CLC_References` location, select the Custom Set under the Custom Sets tab and click on the **Copy to References** button (figure 11.20).

The elements to copy can then be selected (figure 11.21).

Once the data has been copied, workflows referring to it will automatically refer to the file under `CLC_References`, rather than the original location.

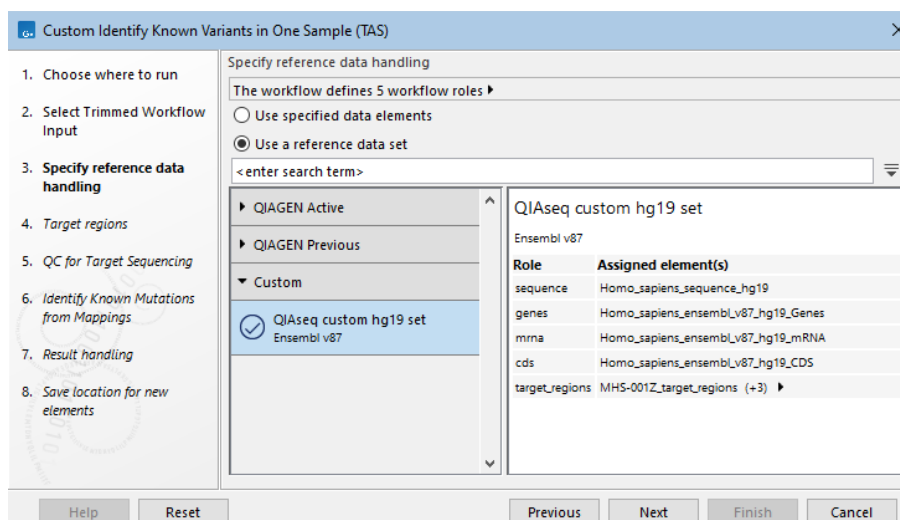


Figure 11.16: When launching a workflow where workflow roles have been defined for one or more inputs, a wizard step will prompt for the relevant Reference Data Set to be selected. Information about the contents of that set are shown in the right hand pane. The small arrow in the `target_regions` row indicates that more than one element has been assigned to that role.

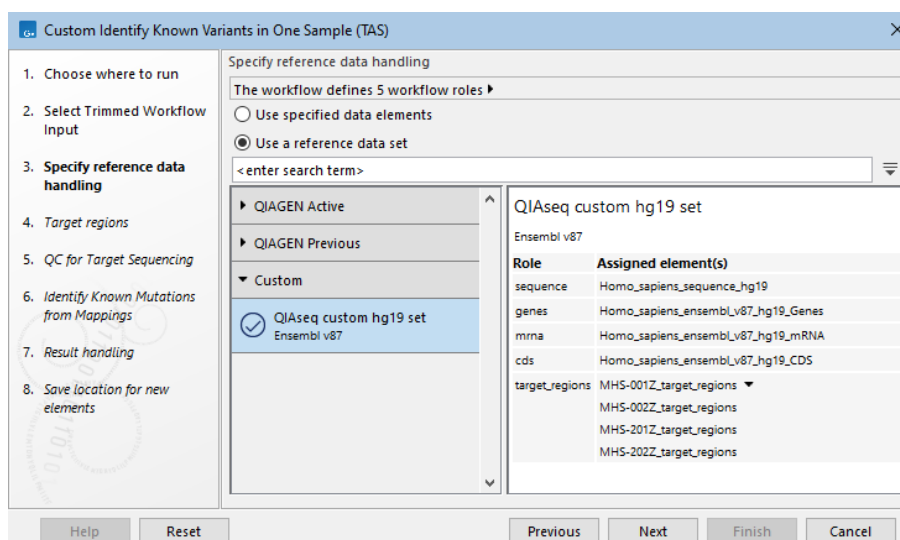


Figure 11.17: Information about the elements assigned to workflow roles in a selected Reference Data Set are shown in the right hand pane in the "Specify reference data handling" wizard step. For roles with multiple elements assigned, clicking on the arrow at the right side of the row reveals the list of these elements.

11.3.2 Export a Custom Data Set

It is possible to export a Custom Set (figure 11.22) to a .cpc file. This set can be imported using any CLC Genomics Workbench, run with a license or in Viewing Mode.

Exporting a Custom Set exports the roles included in the set. The data elements associated with each role can be exported too, but whether this is necessary depends "Reference Data Type", as described below.

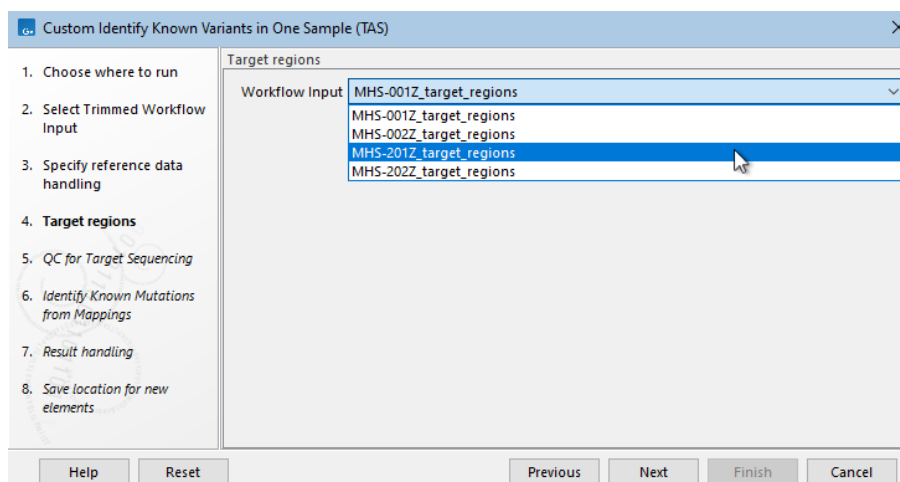


Figure 11.18: In the previous wizard step, a Reference Data Set was selected where multiple elements were assigned to the role `target_regions`. In this wizard step, a drop-down list of those elements is presented, from which the relevant target region track can be selected.

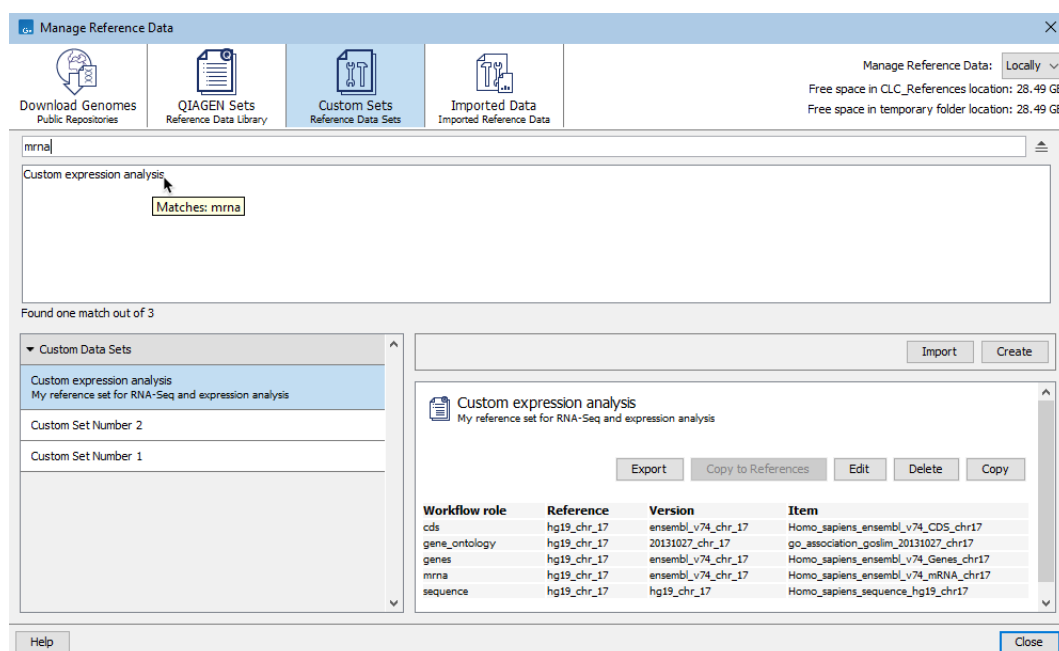


Figure 11.19: Terms entered in the search field when the Custom Sets tab is selected are searched for in the sets available under that tab. Hovering the cursor over a hit opens a tooltip with information about the match.

On the left is an Export column. When a check box in this column is unchecked, only the role is exported. If the box is checked, the data element associated with that role will also be exported.

A "Reference Data Type" is listed in the Export Custom Data Set dialog. The type has the following implications for the Export:

- QIAGEN reference data: Data available under the QIAGEN Sets tab. It is not necessary to include these elements in the export file. The recipient of the Custom Set can download

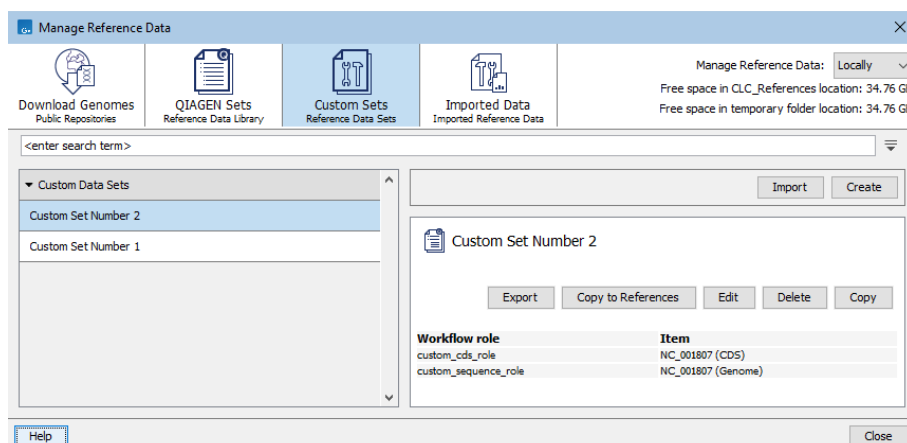


Figure 11.20: Data elements defined in a Custom Set can be copied to a CLC_References location by clicking on the Copy to References button, available when viewing the Custom Set.

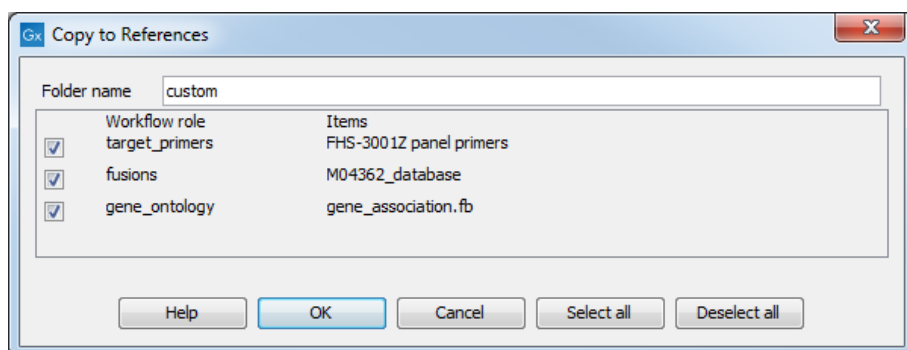


Figure 11.21: Copying an element from the Navigation Area to the CLC_References location.

the relevant data via the Reference Data Manager of their Workbench.

- Custom reference data: Data in the CLC_References location that is not present in the QIAGEN repository. If this Custom Set will be shared with others, you may need to include the data in the export file unless you know they already have this data in the expected location.
- Not reference data: Data not being managed by the Reference Data Manager, i.e. not located under the CLC_References area. Such data cannot be included in the exported set. To remedy this, exit the Export dialog, import the data element into the CLC_References area using the "Copy to References" button, and start the export again.

By exporting only roles and any custom data elements, the size of exported files can be kept to a minimum.

11.3.3 Import a Custom Data Set

A custom data set (.cpc) can be imported using the Import button under the the Custom Sets tab (figure 11.23).

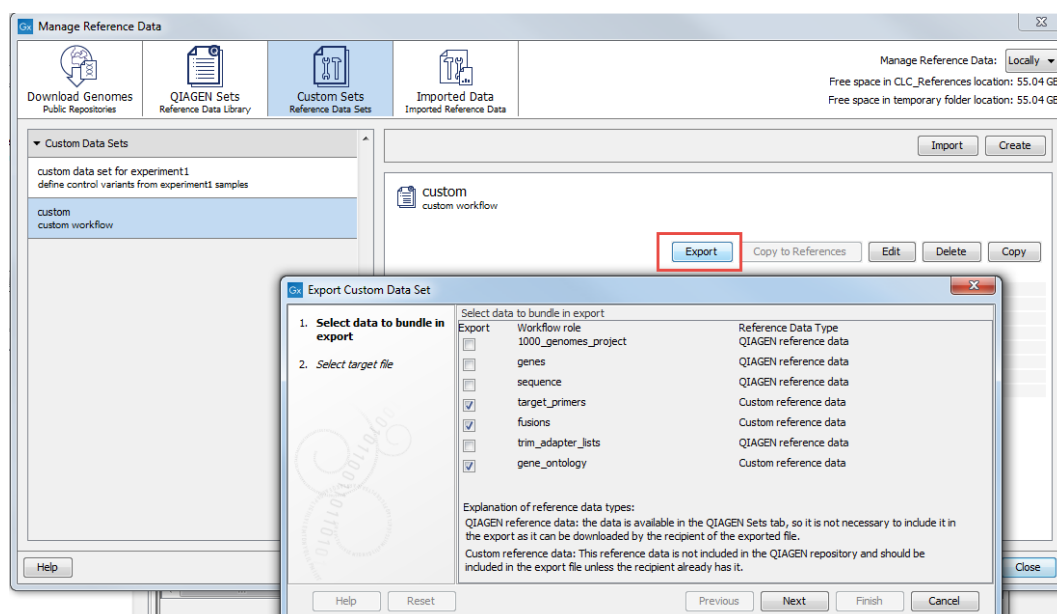


Figure 11.22: Exporting a custom data set from the Custom Sets tab. In this example, we are exporting 7 roles, as well as data for 3 of them.

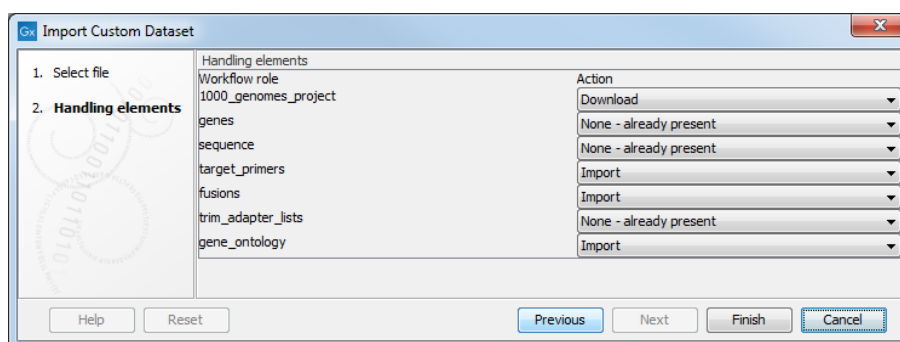


Figure 11.23: Importing a custom data set from the Custom Sets tab.

Import involves importing a set of roles. You are also given the following options for how to handle any associated data:

- **Download** A Custom Set contains reference data that was specified, but not exported and that the data is available from the QIAGEN repository. This will download the data to your CLC_References location. This can take some time for large elements.
- **None - already present** Data specified in the Custom Set is already present in your CLC_References location.
- **Import** The Custom Set includes data elements you do not already have in your CLC_References location. Importing the data set will also import these elements and save them under a folder called "Imported", under the CLC_References.

The progress of downloads and import can be seen in the Processes tab, in the bottom left side of the Workbench.

You can choose not to import or download data by using the drop down menu options present for each role included in the Custom Set.

11.4 Storing, managing and moving reference data

Reference data downloaded using the Reference Data Manager is stored in a CLC File Location called CLC_References. Data can also be added to this location via launch wizards for workflows with input elements configured to use workflow roles (see section 11.3). Data cannot be added by moving files within the Navigation Area.

Data in CLC_References locations can only be deleted using functionality in the Reference Data Manager.

This section describes general functionality relating to managing data downloaded using the Reference Data Manager, including working with a *CLC Server* and how to get reference data onto non-networked machines.

Downloading reference data locally

Data downloaded locally using the Reference Data Manager (figure 11.24) is stored in a CLC File Location called **CLC_References**. By default, this location is configured to refer to a folder of the same name in your home area. If that folder does not already exist when the *CLC Genomics Workbench* is first started up, it is created and added as a CLC File Location.



Figure 11.24: When reference data is stored locally, "Locally" is shown in the top right side of the Reference Data Manager, along with information about how much space is available.

You can see the underlying folder that this location is mapped to by hovering the mouse cursor over the location in the Navigation Area (figure 11.25).

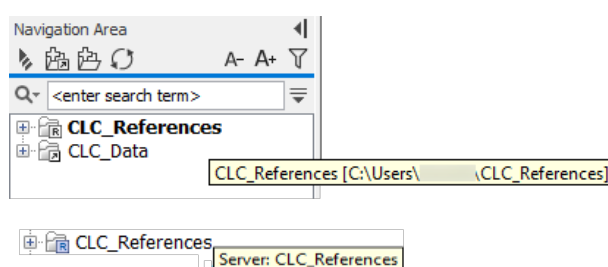


Figure 11.25: Hover the mouse cursor over a CLC_References File Location to see the folder it is mapped to on the file system. By default, this is a folder in your home area (top). When connected to a CLC Server with a CLC_References Location, the tooltip states that the location is on the server (bottom).

Specifying a different folder for reference data

The folder on the file system where reference data should be stored, i.e. the folder that your local CLC_References File Location is mapped to, can be configured by right-clicking on the CLC_References location and choosing Location | Specify Reference Location (figure 11.26). This can be useful when larger amounts of space are needed or when sharing the reference data folder with others.

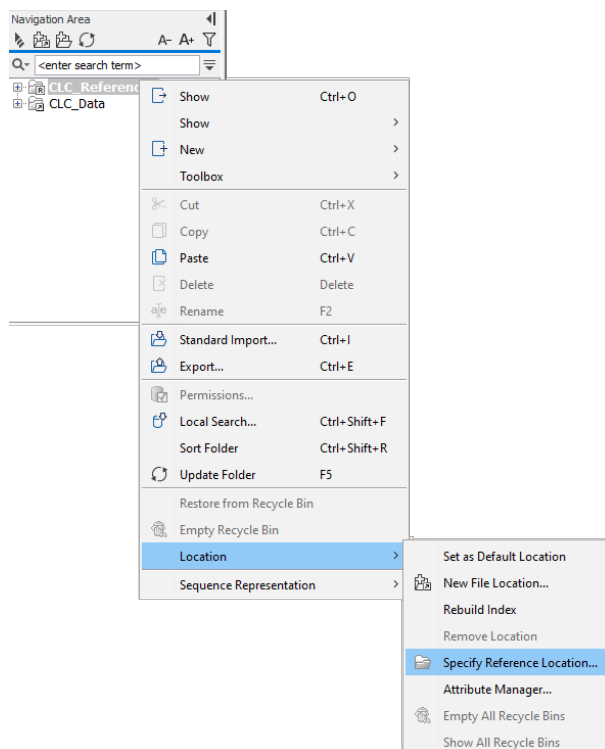


Figure 11.26: To map the local CLC_References location to a different folder on the file system, right-click on CLC_References in the Navigation Area and select Location | Specify Reference Location.

Updating where the CLC_References File Location is mapped to does **not** remove the old CLC_References folder on the file system or its contents. Standard system tools should be used to delete these items if they are no longer needed.

Working with reference data on a CLC Server

When the CLC Genomics Workbench is connected to CLC Genomics Server configured with a File System Location called CLC_References, the option "On Server" can be selected in the Manage Reference Data drop-down list (figure 11.27).

When the "On Server" option is selected, the information shown in the Reference Data Manager refers to data stored in the CLC_References File System Location on the CLC Server, and data downloaded is downloaded to that location. By default, data is downloaded directly to the CLC Server, but downloads can be configured to go via the CLC Genomics Workbench instead using a setting in the Workbench Preferences. This can be useful if the CLC Server does not have access to the external network but the CLC Genomics Workbench does. See section 4.4.

Copying reference data

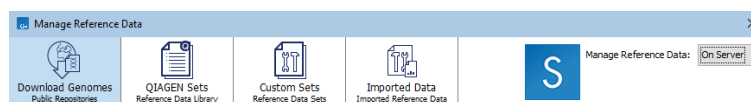


Figure 11.27: When reference data is on the CLC Server the Workbench is connected to, the "On Server" option can be selected in the Manage Reference Data drop down list at the top, right side of the Reference Data Manager.

Reference data can be easily copied from a CLC_References location in a CLC Workbench to a CLC_References location on a CLC Server or vice versa.

A button labeled **Copy from WB** will be visible when the selected data is available in the CLC_References area of your Workbench and you have selected the "On Server" option in the Reference Data Manager. Clicking on this button copies the data from the Workbench to the CLC Server CLC_References location.

Conversely, if you are working with a CLC_References location on your Workbench (i.e. the "Locally" option is selected in the Reference Data Manager) and you are connected to a CLC Server with a CLC_References location configured, a button labeled **Copy from server** will be present. Clicking on this copies the data to your Workbench CLC_References location from the CLC Server CLC_References location.

Copying data from other locations into a CLC_References location is described in section 11.4.1. Copying data from a CLC_References location to elsewhere on the file system is described in section 11.4.2.

Reference data on non-networked systems

If the CLC Genomics Workbench is installed on systems without access to the external network, the following steps can be followed to import reference data to the non-networked Workbench:

1. Install CLC Genomics Workbench on a machine with access to the external network.
2. Download an evaluation license via the Workbench License Manager. If you have problems obtaining an evaluation license this way, please write to us at ts-bioinformatics@qiagen.com.
3. Use the Reference Data Manager on the networked Workbench to download the reference data of interest. By default, this would be downloaded to a folder called CLC_References.
4. When the download is completed, copy the CLC_References folder and all its contents to a location where the machines with the CLC software installed can access it.
5. Get the software to refer to that folder for reference data: in the Navigation Area of the non-networked Workbench, right click on the CLC_References, and choose the option "Specify Reference Location...". Choose the folder you imported from the networked Workbench and click **Select**.

You can then access reference data using the Reference Data Manager.

11.4.1 Imported Data

The Imported Data tab enables you to import Reference Data Sets or data elements to the read-only CLC_References location from two different places.

- **Copy from the Navigation Area** Select a folder in the Navigation Area to import it and its contents into the CLC_References location.
- **Import from file** Specify a *.cpd data package on your computer for import into the CLC_References location. A *.cpd file can be generated by exporting a data package, as explained below.

The following information can be added or edited for the data being imported: Name of the dataset, Description, Author name, Author email, and Organization (figure 11.28).

For folders imported using the **Copy to References** button of the Custom Sets tab, the button **Finalize** can be used to add the above information (figure 11.28). Finalized imported data means that it is not possible to add additional elements to the imported folder.

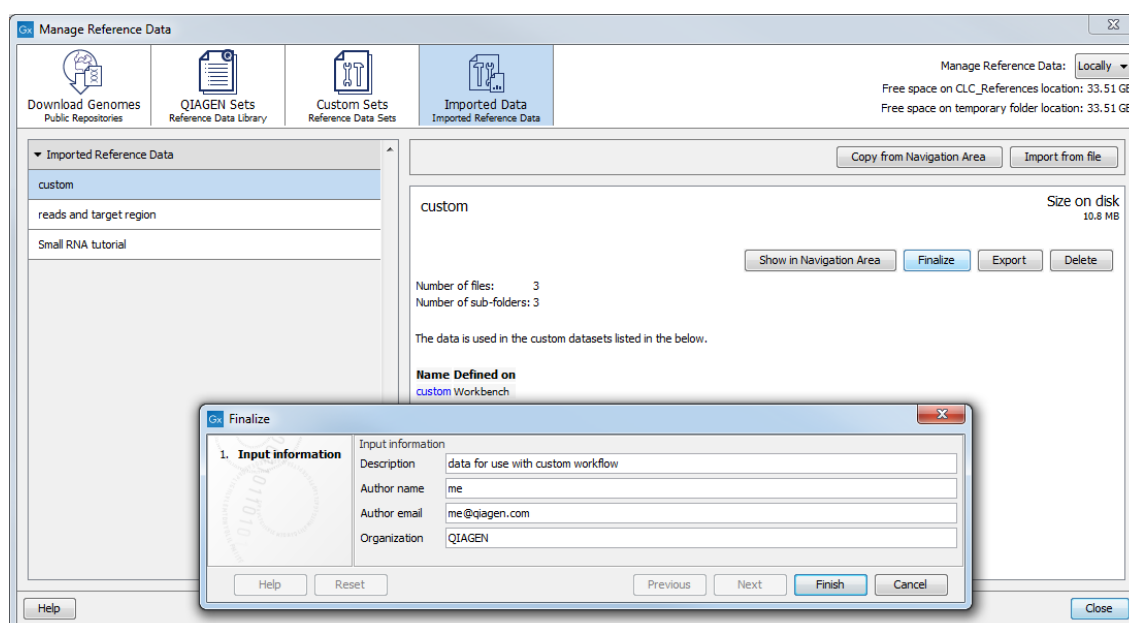


Figure 11.28: Finalizing an imported reference set from the Imported Data tab.

Imported folders are listed in the view to the left, under the "Imported Reference Data" header. Upon selecting an imported Reference Data file, one can access the elements it contains by clicking **Show in Navigation Area**. It is also possible to **Export** such a file (as a *.cpd file), or to **Delete** the folder (for server admin only if the data is on a server). Note that it is never possible to delete a CLC_References file through the Navigation Area as the folder is a read-only location.

11.4.2 Exporting reference data outside of the Reference Data Manager framework

To use the same reference data for running third party applications, including tools configured as External Applications, export the reference data elements using standard **Export** functionality (section 14.6.1). The relevant data elements can be selected in the Navigation Area, under the CLC_References folder.

Tools configured as External Applications in the *CLC Genomics Server* must be configured to point at the relevant exported data files.

Further information about External Applications is provided in the *CLC Genomics Server* administration manual at:

https://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=External_applications.html

Chapter 12

Running tools, handling results and batching

Contents

12.1 Running tools	238
12.1.1 Running a tool on a CLC Server	242
12.2 Handling results	243
12.3 Batch processing	244

This section describes how to run tools, and how to handle and inspect results. We cover launching tools for individual runs, as well as launching them in batch mode, where the tool is run multiple times in a hands-off manner, using different input data for each run.

Launching workflows, individually or in batch mode, as well as running sections of workflows in batch mode, are covered in chapter 14.

12.1 Running tools

Launching tools and workflows involves the same series of general actions:

- Data elements to be used in the analysis are selected.
- Any settings necessary for the tool/workflow to run are configured.
- The job is launched.
- Results are opened or saved when the job completes.

There are several ways to launch a tool or installed or template workflow:

- Double click on its name in a tab the Toolbox panel in the bottom, left side of the Workbench.
- Select it from the Tools or Workflows menu at the top of the Workbench.

- Use the Quick Launch (🔍) tool, described below.
- Select the element(s) to analyze and drag them from the Navigation Area onto the name of a tool or workflow in the Toolbox panel.

Using the Quick Launch tool to start jobs

The Quick Launch tool (figure 12.1) can be used to search for, and to launch tools, installed workflows or template workflows. It can be launched in several ways:

- Click on the **Launch** (🔍) button in the Workbench toolbar.
- Use the keyboard shortcut Ctrl + Shift + T (⌘ + Shift + T on Mac).
- Click on the **Quick Launch** (🔍) option at the top of the Tools menu or Workflows menu.
- Click on the (🔍) icon beside the search field at the top of the Tools tab or Workflows tab in the Toolbox panel at the bottom, left of the Workbench.

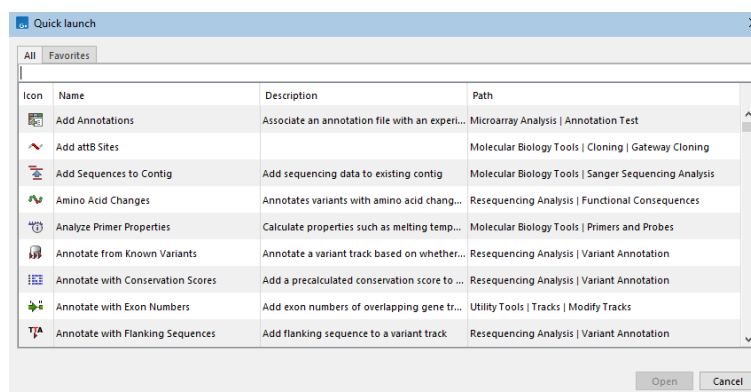


Figure 12.1: Tools, installed workflows and template workflows can be quickly found and launched using the Quick Launch tool.

Double-click on a row to launch a tool or workflow from the Quick Launch dialog. Alternatively, select a row and click on the **Open** button.

When terms are entered into the text field at the top of the Quick Launch dialog, only tools and workflows with matches to those terms in their name, description or path will be listed (figure 12.2). Surround terms with single or double quotes to search for specific terms with spaces in them, for example "sequence list".

The Path column contains the location of tools and workflows relative to the Tools or Workflows menu, respectively. Functionality available under other menus includes the relevant menu name in the path.

Click on the Favorites tab to see the subset of tools that are frequently used or have been selected as favorites (see section 2.3).

For tools where names have changed between Workbench versions, searches using terms in the older name will still find the relevant tool.

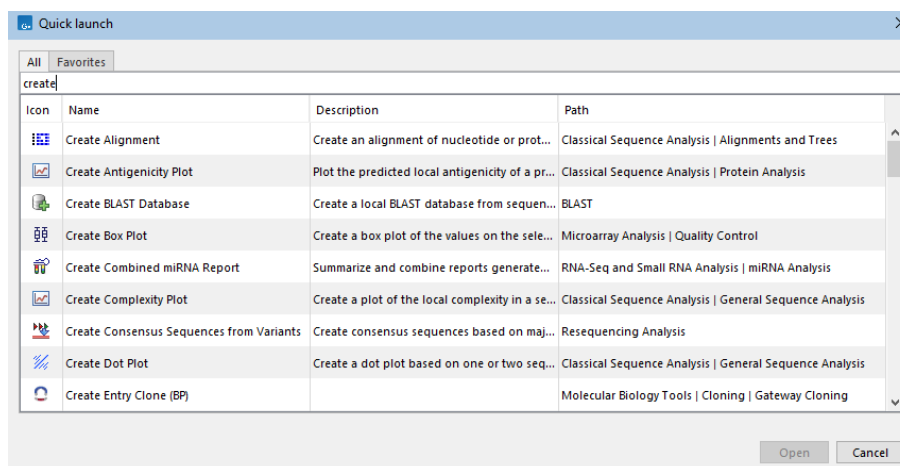


Figure 12.2: Typing a term in the search field limits the list of tools and workflows to those with that term in their name, description or path.

Configuring and submitting jobs

When you launch a tool or an installed workflow, a wizard pops up. The general order of the steps presented via launch wizards are:

1. Specify the execution environment.
2. Select the input data.
3. Configure the available options for the tool.
4. Specify how the results should be handled.

You can move forward and back through the wizard steps by clicking the buttons **Next** and **Previous**, respectively, which are present at the bottom of the wizard. Clicking on the **Help** button in the bottom left corner of the launch wizard opens the documentation for the tool being launched.

The rest of this section covers the general launch wizard steps in more detail.

Specify the execution environment

If more than one execution environment is available, and a default selection has not already been set, the first wizard step will offer a list of the available environments.

For example, if you are logged into a *CLC Server*, or if you have the *CLC Cloud Module* installed and an AWS Connection has been configured with credentials giving access to a *CLC Genomics Cloud* setup, you are offered the option of running the job in different execution environments (figure 12.3).

Information on about launching jobs on a *CLC Server* is provided in section 12.1.1.

Select the input data for analysis tools

When selecting data to use as input to a tool, a view of the Navigation Area is presented, listing the elements that could be selected as input, as well as folders (figure 12.4). The data types that can be used as input for a given tool are described in the manual section about that tool.

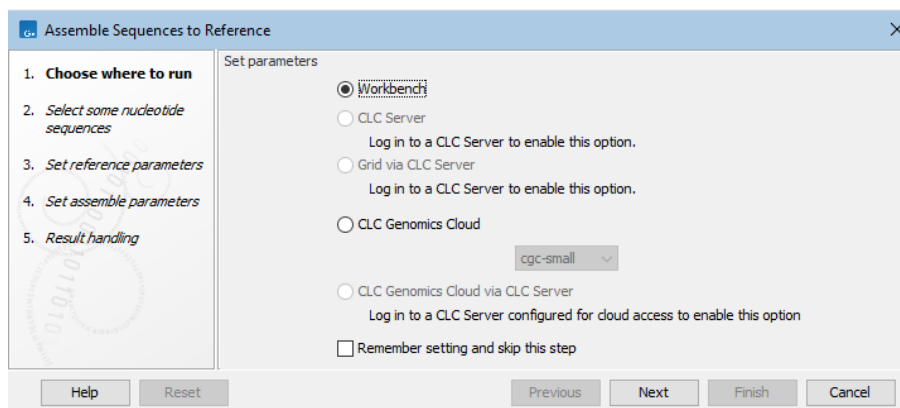


Figure 12.3: This Workbench has the CLC Cloud Module installed and has an active AWS Connection to a CLC Genomics Cloud setup. Thus, this job could be run on the Workbench, or run on AWS by selecting the option CLC Genomics Cloud.

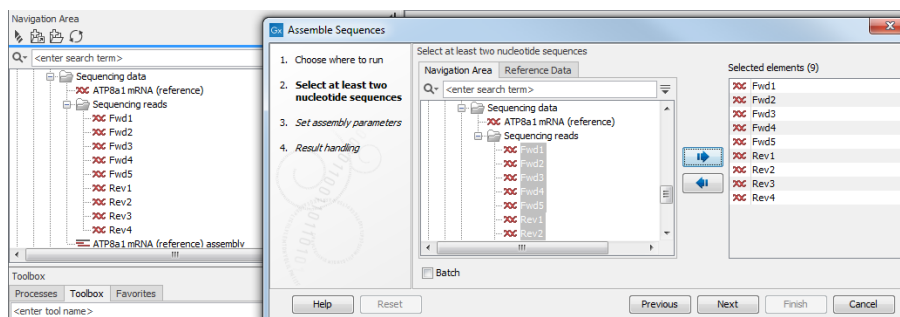


Figure 12.4: You can select input files for the tool from the Navigation Area view presented on the left hand side of the wizard window.

Selected elements will be listed in the right hand pane. To select the inputs, you can:

- Double click on them in the Navigation Area view in the launch wizard, or
- Select them with a single click in the Navigation Area view in the launch wizard and then click on the right hand arrow.
- Before opening the launch wizard, pre-select data elements in the main Navigation Area of the Workbench. When the tool is launched, these elements will automatically be placed in the "Selected elements" list.

To remove entries from the "Selected elements" list, double-click on them or select them with a single click and then click on the left hand arrow.

When multiple elements are selected, most analysis tools will analyze them together, as a single input, unless the "Batch" option at the bottom is checked. With the "Batch" option checked, the tool is run multiple times, once for each "batch unit", which may be a data element, or a folder containing data elements or containing folders of elements. Batch processing is described in section 12.3.

Select the input data for import tools

Selecting files for import is described in chapter 7. It is generally similar to selecting input for analysis tools, but involves selecting files from a file system or remote location. Many import selection wizards also support drag-and-drop for selecting files to import.

Configure the available options for the tool

Depending on the tool, there may be one or more wizard steps containing options affecting how the tool behaves (figure 12.5).

Clicking on the **Reset** button resets the values for the options in that wizard step to their default values.

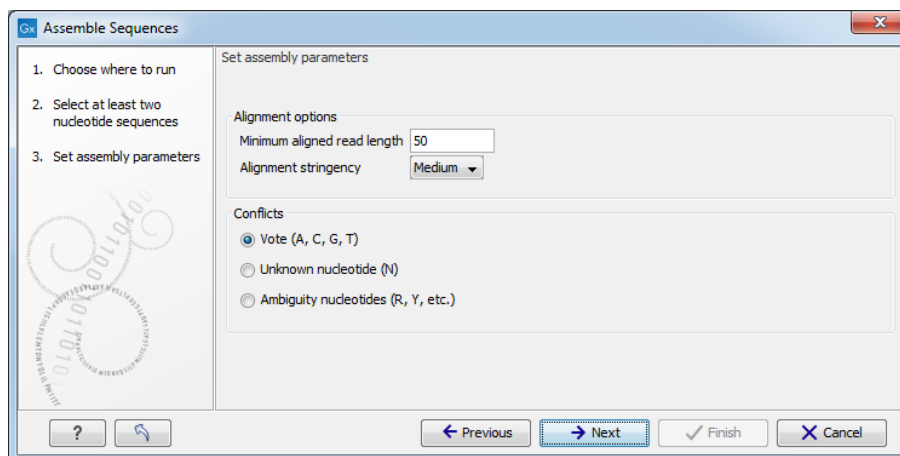


Figure 12.5: An example of a "Set parameters" wizard step.

Specify how the results should be handled

Handling results is described in section 12.2.

12.1.1 Running a tool on a CLC Server

When you launch an analysis from a Workbench that is logged into a CLC Server, you are offered the choice of where the analysis should be run (figure 12.6).

- **Workbench.** Run the analysis on the computer the CLC Workbench is running on.
- **Server.** Run the analysis using the CLC Server. For job node setups, analyses will be run on the job nodes.
- **Grid.** Only offered if the CLC Server setup has grid nodes. Here, jobs are sent from the master CLC Server to be run on grid nodes. The grid queue to submit to can be selected from the drop down list under the Grid option.

You can check the **Remember setting and skip this step** option if you wish to always use the selected option when submitting analyses. If you select this option but later change your mind, just start up an analysis and click on the **Previous** button to open these options again.

Most wizard steps for launching a job on a CLC Workbench or on a CLC Server are the same. There are two minor differences when launching jobs to run on a CLC Server: results are always saved, and a log of the job is always created and saved alongside the results.

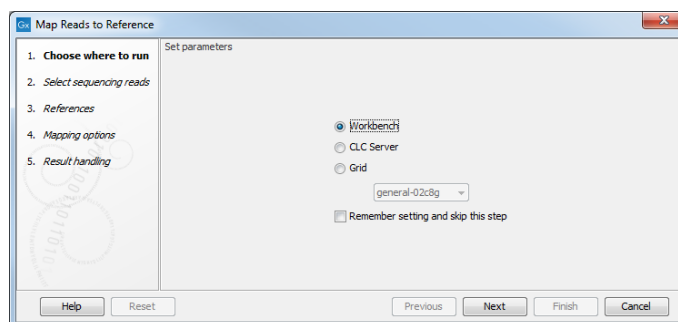


Figure 12.6: When logged into the CLC Server, you can select where a job should be run.

Data access: When you run a job on a CLC Server, you will generally only be able to select data from and save results to areas known to the CLC Server. With default server settings, you will not be able to upload data from your local system. Your server administrator can enable this if they wish. See https://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=Direct_data_transfer_from_client_systems.html.

Disconnecting from the CLC Server: Once the job has been submitted, you can disconnect from the CLC Server if you wish, or close the CLC Workbench entirely. *Exception:* If you are importing data from the local file system, you must wait until the data has been imported before disconnecting. A notification about server jobs that finished is presented the next time you log in to the CLC Server. See section 2.4.

12.2 Handling results

Some tools can generate several outputs. If there is a choice of which ones to generate, you will be able to configure this in the final wizard step, called "Result handling". The kind of output files generated by a tool are described in the tool specific sections of the manual.

For tasks run on a Workbench (as opposed to a CLC Server) the "Result handling" window also allows you to decide whether you want to **Open** or **Save** your results.

- **Open.** This will open the result of the analysis in a view. This is the default setting.
- **Save** The results will be saved rather than opened. You will be prompted for where you wish the results to be saved (figure 12.7). You can save to an existing area or create a new folder to save the results into.

You may also have an option called "Open log". If checked, a window will open in the View area after the analysis has started and the progress of the job will be reported there line by line.

Click **Finish** to start the analysis.

If you chose the option to open the results, they will open automatically in one or several tabs in the View Area. The data will not have been saved at this point. The name of each tab is in bold, appended with an asterisk to indicate this. There are several ways to save the results you wish to keep:

- Drag the tab to the relevant location in the Navigation Area.

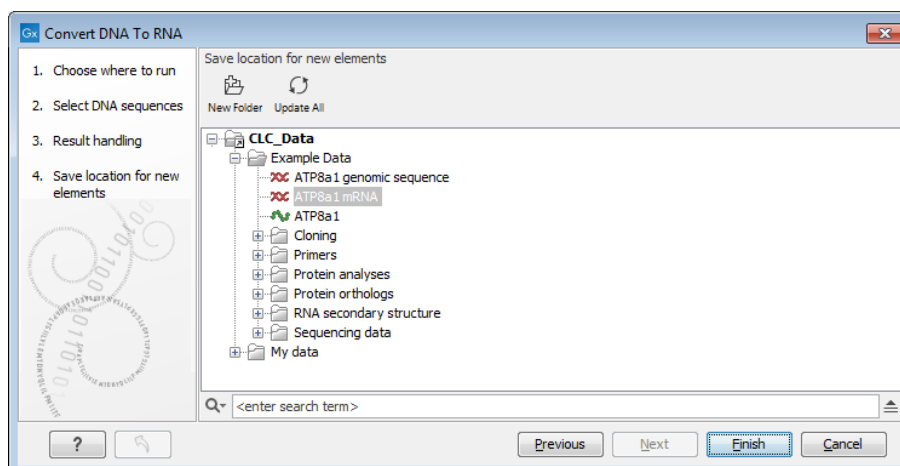


Figure 12.7: Specify where to save the results of an analysis.

- Select the tab and then use the key combination Ctrl + S (or ⌘ + S on macOS).
- Right click on the tab and choose "Save" from the context menu.
- Use the "Save" button in the Workbench toolbar.
- Go to the File menu and select the option "Save" or "Save As..."

If you chose to save the results, they will have been saved in the location specified. You can open the results in the Navigation Area directly after the analysis is finished. A quick way to find the results is to click on the little arrow to the right of the analysis name in the Processes tab and choose the option "Show results" or "Find Results", as shown in figure 12.8.

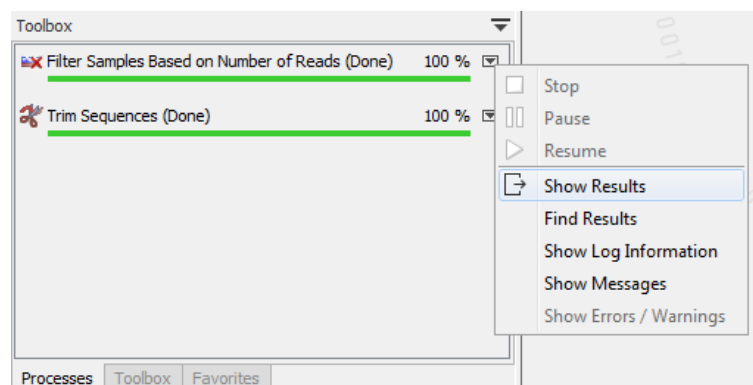


Figure 12.8: Find or open the analysis results by clicking on the little arrow to the right of the analysis name in the Processes tab and choosing the relevant item from the menu.

12.3 Batch processing

Batch processing refers to running an analysis multiple times, once per batch unit. For example, if you have 10 sequence lists and wish to run 10 mapping analyses, one per sequence list, you could launch all 10 analyses by setting up one batch job. Here, each sequence list would be a "batch unit".

This section focuses on batch processing when using **individual tools**. Further details about batch processing of workflows is provided in section 14.3 and section 14.3.3.

Batch mode

Batch mode is activated by clicking the **Batch** checkbox in the dialog where the input data is selected (figure 12.9).

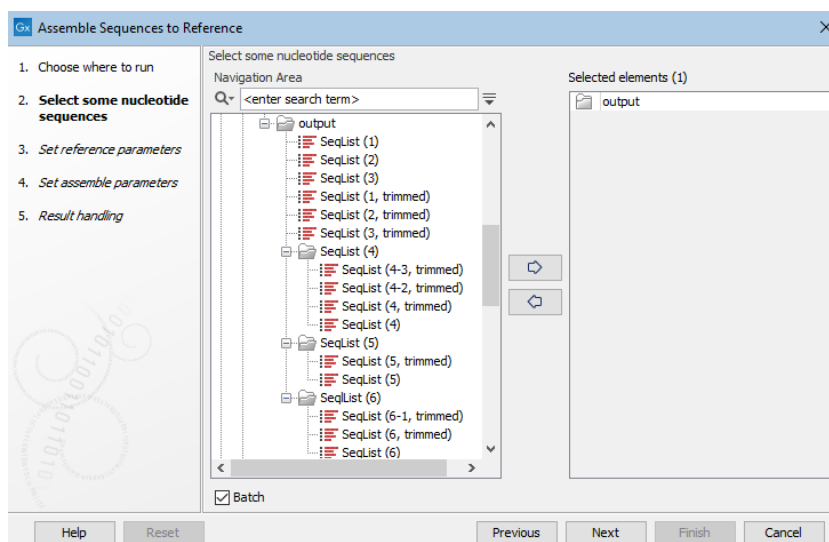


Figure 12.9: When launching an analysis in Batch mode, individual elements and/or folders can be selected. Here, a single folder that contains both elements and subfolders of elements has been selected.

In Batch mode, the analysis is run once per batch unit. A batch unit consists of the data elements to be analyzed together. A batch unit can be a single data element, or can consist of multiple data elements.

Batch units are made up of:

- Each data element selected in the launch wizard.
- Elements and folders within a folder selected in the launch wizard, where:
 - Each data element contained directly within that selected folder is a batch unit.
 - Each subfolder directly under the selected folder is a batch unit. I.e. all elements within that subfolder are analyzed together.
 - Elements within more deeply nested subfolders (e.g. subfolders of subfolders of the originally selected folder) are not used in the analysis.
- Elements with associations to a CLC Metadata Table selected in the launch wizard. Each row in the CLC Metadata Table is a batch unit. Data elements associated with a row, of a type compatible as input to the analysis, are the default contents of a batch unit. See figure 12.10 and figure 12.11.

Batch overview

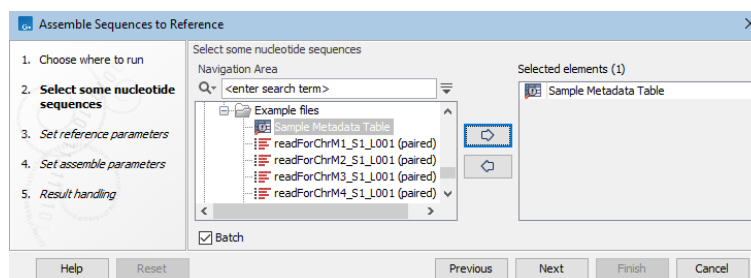


Figure 12.10: When the *Batch* box is checked, a *CLC Metadata Table* can be selected as input.

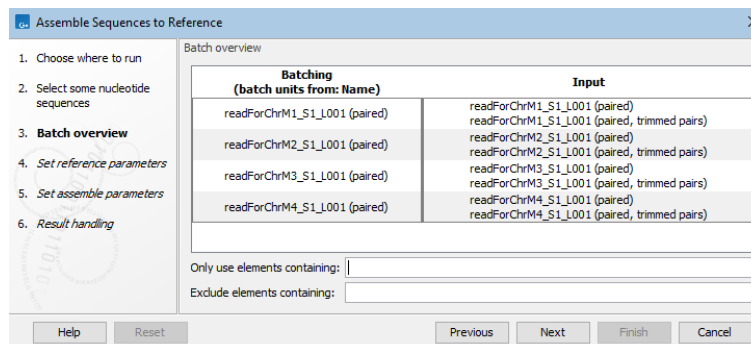


Figure 12.11: Data associated with each row in a *CLC Metadata Table*, of a type compatible with that analysis, make up the default content of batch units.

In the batch overview step, the elements in each batch unit can be reviewed, and refined based on their names using the fields **Only use elements containing** and **Exclude elements containing**.

In figure 12.12, the batch units, i.e. those elements and folders directly under the folder selected in figure 12.9, are shown. In each batch unit, data elements that could be used in the analysis are listed on the right hand side. Some batch units contain more than one data element. Those data elements would be analyzed together. To limit the analysis to just sequence lists containing trimmed sequences, the term "trim" has been entered into a filter field near the bottom.

Folders that do not contain any elements compatible with the analysis are not shown in the batch overview.

Organization of the results

The options for where to save analysis outputs are shown in figure 12.13.

The available options are:

- **Save in input folder** Save all outputs into the same folder as the input data. For batch units defined by folders, the results of each analysis are saved into the folder with the input data. If the batch units were individual data elements, results are put into the same folder as the input elements.
- **Save in specified location** You will be prompted in the next step to select a folder where the outputs should be saved to. The **Create subfolders per batch unit** checkbox allows you to specify whether subfolders should be created to store the results from each batch unit:

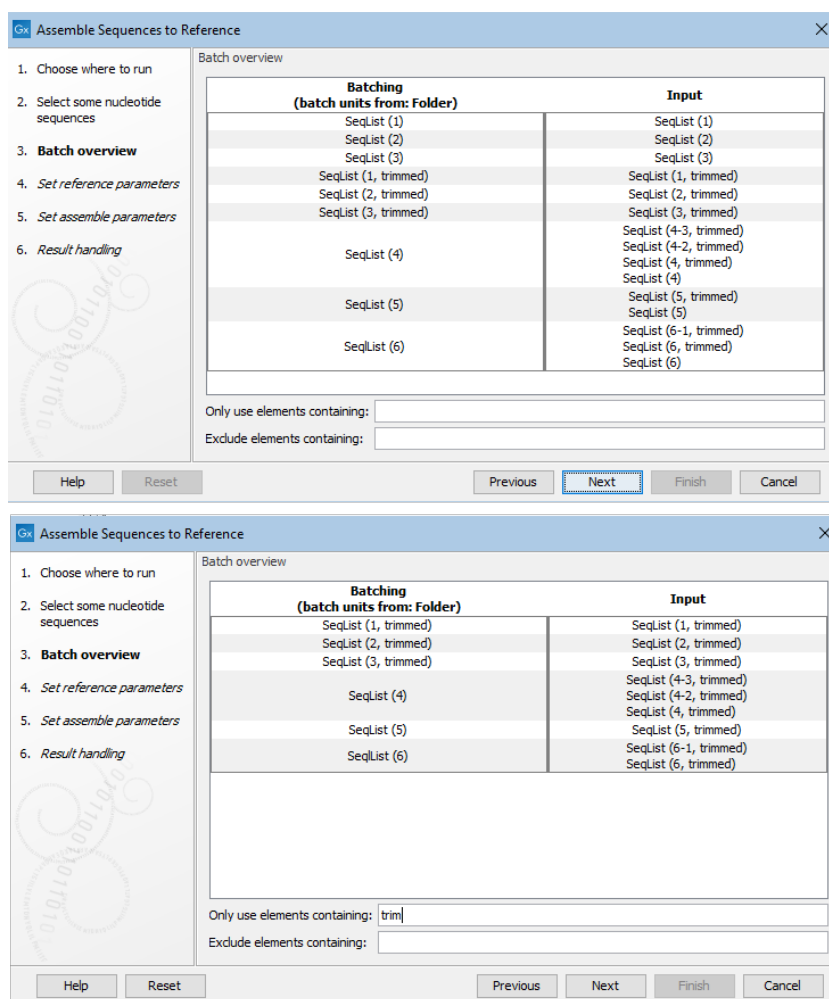


Figure 12.12: Overview of the batch units (left) and the input elements defined by each batch unit (right). By default, all elements that can be used as inputs are listed on the right (top). By entering terms in the filter fields, the list of elements in the batch units can be refined. Here, only sequence lists including trimmed sequences will be included (bottom) .

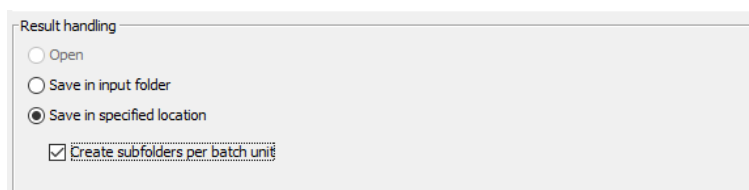


Figure 12.13: Options for saving results when an analysis is run in Batch mode.

- When **checked** results for each batch unit are written to a newly created subfolder under the folder you select in the next step. A subfolder is created for each batch unit. (This is the default option.)
- When **unchecked**, results from all batch units are written to the folder you select in the next step.

The log file

In the final wizard step there is an option to **Create a log**. When checked, a log containing

information about all the batch units will be created. This log includes the term "combined log" in its name. A log is also created for each individual batch unit.

Batch unit processes

When the job is running, there is one "master" process representing the overall batch job, and a separate process for each batch unit.

On a *CLC Workbench*, the batch units are executed sequentially - one batch unit at a time. This avoids overloading the computer.

On a *CLC Server*, all the processes are placed in the queue, and the queue takes care of distributing the jobs. If there are multiple job nodes or grid nodes, batch units may be processed in parallel.

Stopping a batch run

To stop the whole batch run, stop the "master" process.

On a *CLC Workbench*, find the master process in the Processes tab in the bottom left side. Click on the little triangle on the right hand side of the master process and choose the option **Stop**.

Chapter 13

Metadata

Contents

13.1 Creating metadata tables	251
13.1.1 Importing metadata	251
13.1.2 Creating a metadata table directly in the Workbench	253
13.2 Associating data elements with metadata	257
13.2.1 Associate Data Automatically	258
13.2.2 Associate Data with Row	260
13.3 Working with data and metadata	261
13.3.1 Finding data elements based on metadata	261
13.3.2 Viewing metadata associations	262
13.3.3 Removing metadata associations	263
13.3.4 Identifying metadata rows without associated data	264
13.3.5 Editing Metadata tables	264
13.4 Moving, copying and exporting metadata	268

Metadata refers to information about data. In the context of the *CLC Genomics Workbench*, this usually means information about samples. For example a set of reads could come from a particular specimen at a particular time point with particular characteristics. The specimen, time and characteristics would be metadata for that set of reads.

Examples in this chapter refer to tools present in the CLC Genomics Workbench, but the principles apply to other CLC Workbenches.

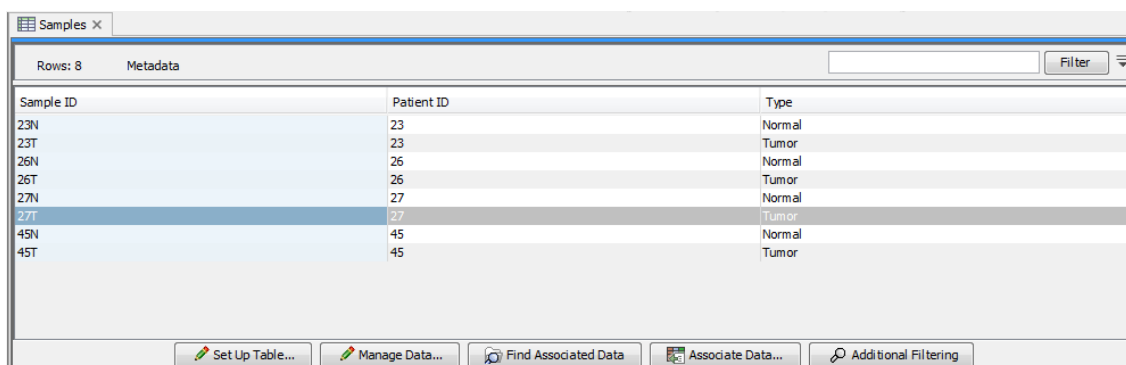
What is metadata used for? Core uses of metadata in CLC software include:

- Defining batch units when launching workflows in batch mode, described in section [14.3.2](#).
- Distributing data to the relevant input channels in a workflow when using Collect and Distribute elements, described in section [14.2.5](#).
- Finding and selecting data elements based on sample information (in a CLC Metadata Table). Workflow Result Metadata tables are of particular use when reviewing results generated by workflows run in batch mode and are described in section [14.3.1](#).

- Running tools where characteristics of the data elements are relevant. Examples are the differential expression tools, described in section 33.6.

Metadata tables

An example of a CLC Metadata Table in the *CLC Genomics Workbench* is shown in figure 13.1. Each column represents a property of a sample (e.g., identifier, height, age, treatment) and each row contains information relevant to a sample. A single column can be designated the key column. That column must contain unique entries.



Sample ID	Patient ID	Type
23N	23	Normal
23T	23	Tumor
26N	26	Normal
26T	26	Tumor
27N	27	Normal
27T	27	Tumor
45N	45	Normal
45T	45	Tumor

Figure 13.1: A simple metadata table, with the key column highlighted in blue.

Each row can have associations with one or more data elements, such as sequence lists, expression tracks, variant tracks, etc. Associating data elements with relevant metadata rows, automatically or manually, is covered in section 13.2

Information from an Excel, CSV or TSV format file can be imported into a CLC Metadata Table, as described in section 13.1.1. CLC Metadata Tables are also generated by workflows, as described in section 14.3.1.

A template workflow for importing sequence data with associated metadata can be found in the Preparing Raw Data folder in the Template Workflows folder under the Workflows menu (see section 14.5.1).

Metadata Elements table

Data elements with an association to a row in a CLC Metadata Table can be listed by selecting the rows of interest and clicking on the **Find Associated Data** button. The Metadata Elements table opens, with a table of information about elements with associations to the selected rows (figure 13.2).

When a data element is associated with a metadata row, the outputs of analyses involving that data usually inherit the metadata association automatically. For example, if a sequence list with an association to a CLC Metadata Table row is used as input to analyses, results of these analyses may also be associated with that row (figure 13.2).

Inheritance of associations to metadata requires that a single association can be unambiguously identified for an output when a tool is run. If an output is derived from two or more inputs with different metadata associations, then no association will be inherited.

Samples Rows: 8 Metadata

Sample ID	Patient ID	Type
23N	23	Normal
23T	23	Tumor
26N	26	Normal
26T	26	Tumor
27N	27	Normal
27T	27	Tumor
45N	45	Normal
45T	45	Tumor

Buttons: Set Up Table... Manage Data... Find Associated Data... Associate Data... Additional Filtering

Metadata Elements Rows: 6 Filter to Selection...

Sample ID	Role	Type	Name	Path
27T	Sample		27T_R1_001 sampled	CLC_Data / RNA / advanced RNASeq / reads / 27T
27T	Gene expression		27T_R1_001 sampled (GE)	CLC_Data / RNA / advanced RNASeq / results
27T	Report		27T_R1_001 sampled RNA-Seq report	CLC_Data / RNA / advanced RNASeq / results
27T	Transcript expression		27T_R1_001 sampled (TE)	CLC_Data / RNA / advanced RNASeq / results / gene_transcripts
27T	Report		27T_R1_001 sampled RNA-Seq report	CLC_Data / GxWB / JSON
27T	RNA-seq read mapping		27T_R1_001 sampled RNA-Seq (Reads)	CLC_Data / RNA / advanced RNASeq / results

Buttons: Find in Navigation Area Refresh Close

Figure 13.2: A CLC Metadata Table and corresponding Metadata Elements table showing elements associated with sample 27T.

13.1 Creating metadata tables

CLC Metadata tables are created in several ways:

- Import metadata from an Excel, CSV or TSV format file.
To do this, use the **Import Metadata** (📄) tool, described in section 13.1.1.
- Create a new CLC Metadata Table containing a subset of the rows in another CLC Metadata Table.
To do this, open an existing CLC Metadata Table, select the rows of interest and click on the **Create New Metadata Table...** (📄) button at the bottom of the editor. This option is also available in the menu that opens when you right-click on the selection (figure 13.3).
Data elements with associations to the selected rows acquire an association with the new CLC Metadata Table also.
- Create a new CLC Metadata Table from scratch.
This is described in section 13.1.2.

Workflow Result Metadata tables, created when a workflow is run, are also CLC Metadata Tables. These are described in section 14.3.1.

The **Import with Metadata** template workflow, described in section 14.5.1, takes advantage of the Workflow Result Metadata element generated by workflows to make import of sequences with associated metadata simple.

13.1.1 Importing metadata

The Import Metadata tool

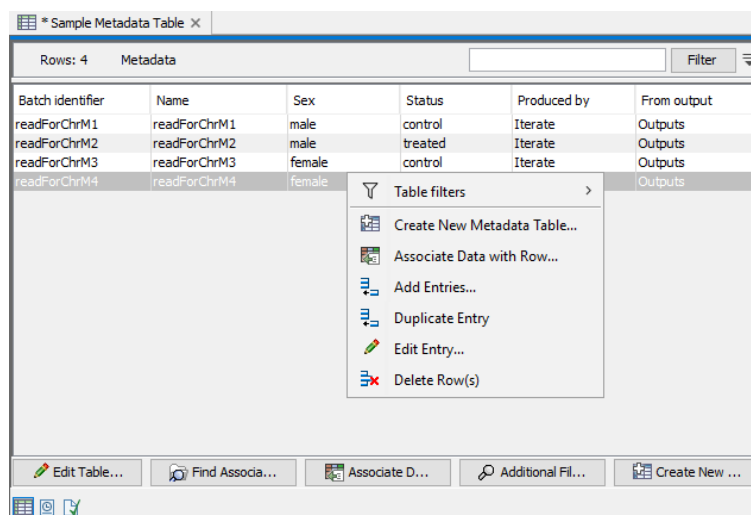


Figure 13.3: Selected rows in a CLC Metadata table can be put into a new CLC Metadata Table using the option "Create New Metadata Table..."

To import metadata into a CLC Metadata Table, go to:

File | Import (📁) | Import Metadata (📄)

The first column in the selected file must have unique entries. That column will be designated as the key column. A different column can be specified as the key column later. For the optional step for association of data elements to work, the first column must contain entries that can be matched with the relevant data element names.

Select the Excel, CSV or TSV format file with metadata to be imported. The rows in that file are displayed in the Metadata preview window (figure 13.4).

The format of the columns should reflect the column contents. The designated format can be changed from within the Metadata Table editor, as described in section 13.3.5. There, you can change the column data types (e.g. to types of numbers, dates, true/false) and you can designate a new key column.

Associating metadata with data (optional) The "Associate with data" wizard step (figure 13.5), is optional. To proceed without associating data to metadata, click on the **Next** button. Associating data with metadata can be done later, as described in section 13.2.

To associate data with the metadata:

- Click on the file browser button to the right of the **Location of data** field
- Select the data elements to be associated.
- Select the matching scheme to use: Exact, Prefix or Suffix. These options are described in section 13.2.1.

The Data association preview area shows data elements that will have associations created, along with information from the metadata row they are being linked with. This gives the opportunity to check that the matching is leading to the expected links between data and metadata.

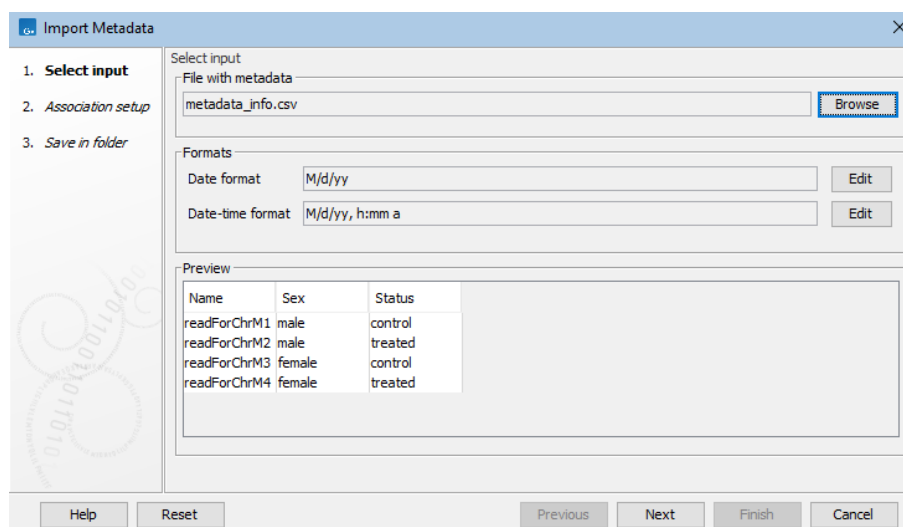


Figure 13.4: Rows being imported from a file containing metadata are shown in the Metadata preview table.

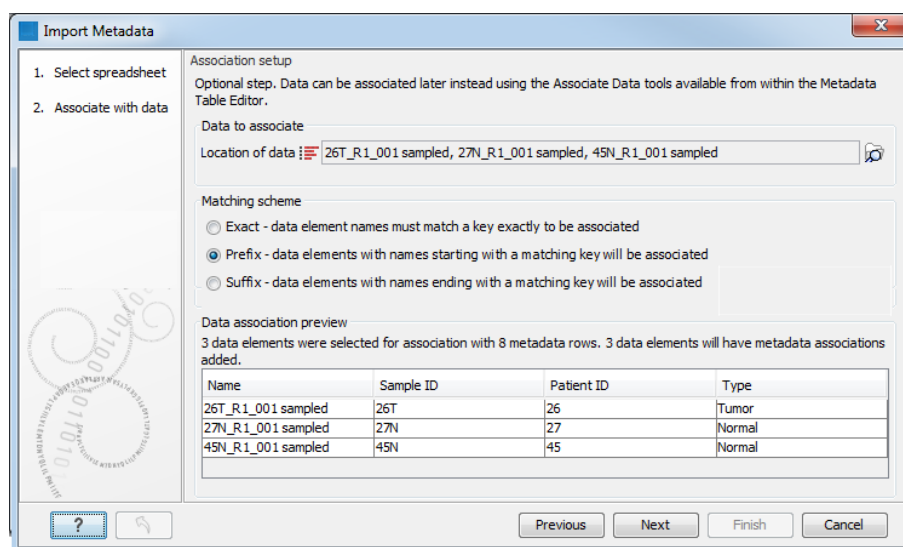


Figure 13.5: Three data elements are selected for association. The "Prefix" partial matching scheme is selected for matching data element names with the appropriate metadata row, based on the information in the Sample ID column in this case.

You can then select where you wish the metadata table to be saved and click on **Finish**.

The associated information can be viewed for a given data element in the Show Element Info view (figure 13.6).

13.1.2 Creating a metadata table directly in the Workbench

Creating CLC Metadata tables from scratch within the *CLC Genomics Workbench* is described in this section.

See also section 13.1.1 for importing information from Excel to create a CLC Metadata Table. If your analysis is contained in a workflow, a CLC Metadata Table can be created automatically. This is described in section 14.3.1.

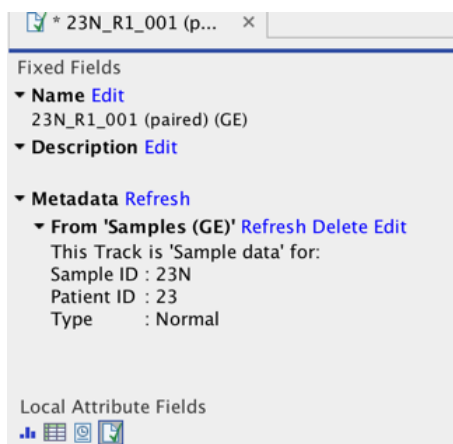


Figure 13.6: Metadata associations can be seen, edited, refreshed or deleted via the Show Element Info view.

To create a CLC Metadata Table manually, go to:

File | New | Metadata Table (📄)

This opens a new metadata table with no columns and no rows. Importing metadata using the Metadata Table Editor requires that the **table structure** is defined first.

Defining the table structure Click **Setup Table** at the bottom of the view (figure 13.7).

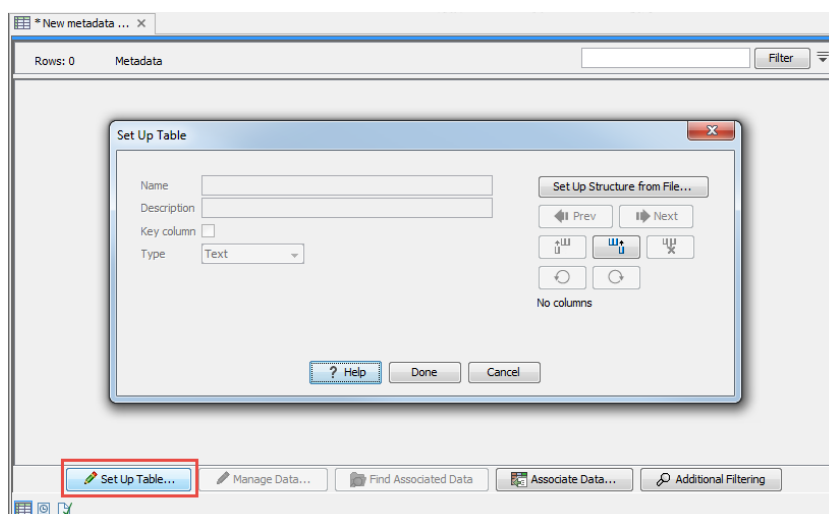


Figure 13.7: Dialog used to add columns to an empty Metadata Table.

To create a metadata table from scratch, use the "Add column right" or "Add column left" buttons (📄) to define the table structure with the amount of columns you will need, and edit the fields of each column as needed.

To import the table from a file, click on **Setup Structure from File**. In the dialog that appears (figure 13.8), you need to provide the following information:

- **Filename** The EXCEL or delimited TEXT file to import. Column names should be in the first row of this file.

- **Encoding** For text files only: the encoding used to create the file. The default is UTF-8.
- **Separator** For text files only: The character used to separate the columns. The default is semicolon (;).

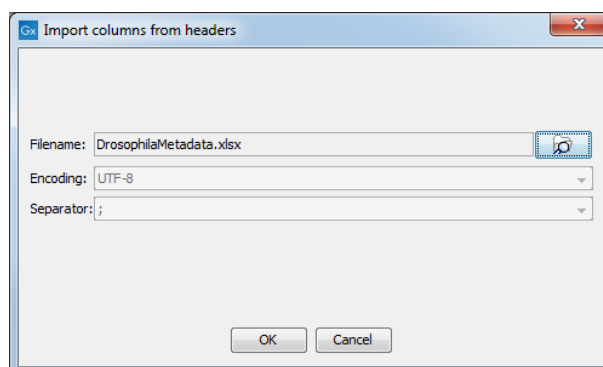


Figure 13.8: Creating a metadata table structure based on an external file.

For each column in the external file, a column will be created in the new metadata table. By default the type of these imported columns is "Text". You will see a reminder to set the column type for each column and to designate one of the columns as the key column.

Populating the table Click on **Manage Data** button at the bottom of the view (figure 13.9).

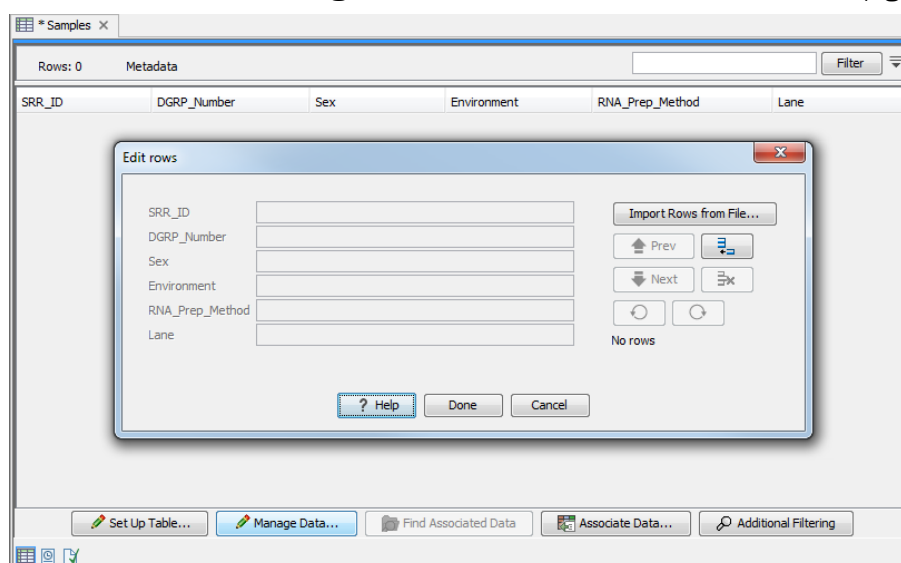



Figure 13.9: Tool for managing the metadata itself. Notice the button labeled *Import Rows from File*.

The metadata table can then be populated by editing each column manually. Row information is added manually by clicking on the  button and typing in the information for each column.

It is also possible to import information from an external file. In that case, the column names in the metadata table in the workbench will be matched with those in the external file to determine which values go into which cell. Only cell values in columns with an exact name match will be imported. If the file used contains columns not in the metadata table, the values in those columns will be ignored. Conversely, if the metadata table contains columns not present in the file, imported rows will have no values for those columns.

Click on **Import Rows from File** and select the external file of metadata. This brings up the window shown in figure 13.10.

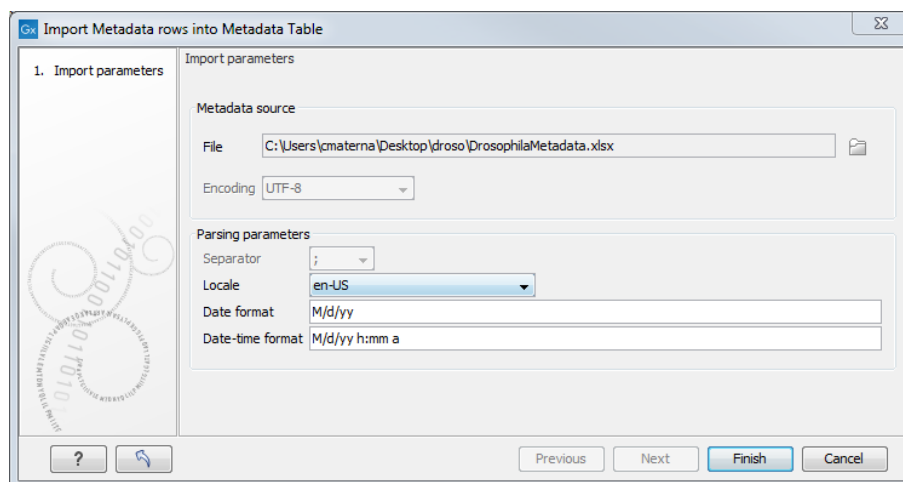


Figure 13.10: Tool to import rows into a Metadata Table.

When working with an existing metadata table and adding extra rows, it is generally recommended that a key column be designated first. If a key column is not present, then all rows in the file will be imported. With no key column designated, if any rows from that file were imported into the same metadata table earlier, a duplicate row will be created. With a key column, rows with a new, unique entry for that column are added to the table and existing rows with a key entry in the file will be updated, incorporating any changes present in the file. Duplicate rows will not be created.

The options presented in the Import Metadata Rows into Metadata Table are:

- **File.** The file containing the metadata to import. This can be Excel (.xlsx/.xls) format or a delimited text file.
- **Encoding.** For text files only: The text encoding of the selected file. Specifying the correct encoding is important to ensure that the file is correctly interpreted.
- **Separator.** For text files only: the character used to separate columns in the file.
- **Locale.** For text files only: the locale used to format numbers and dates within the file.
- **Date format.** For text files only: the date format used in the imported file.
- **Date-time format.** For text files only: the date-time format used in the imported file. The date and date-time templates uses the Java patterns for date and time formatting. Meaning of some of the symbols:

Symbol	Meaning	Example
y	Year	2004; 04
d	Day	10
M/L	Month	7; 07; Jul; July; J
a	am-pm	PM
h	Hour (0-12 am pm)	12
H	Hour (0-23)	0
m	Minute	30
s	Second	55

Examples of using this:

Format	Meaning	Example
dd-MM-yy	Short date	31-12-15
yyyy-MM-dd HH:mm	Date and Time	2015-11-23 23:35
yyyy-MM-dd'T'HH:mm	ISO 8601 (standard) format	2015-11-23T23:35

With a short year format (YY), 2000 will be added when imported as, or converted to, Date or Date and time format. Thus, when working with dates before the year 2000 or after 2099, please use a four digit format for the year (YYYY).

Click the button labeled **Finish** button when the necessary fields have been filled in.

The progress and status of the row import can be seen in the Processes tab in the Toolbox area, at the bottom, left side of the Workbench. Any errors resulting from an import that failed can be reviewed here. The most frequent errors are associated with selecting the wrong separator or encoding, or wrong date/time formats when importing rows from delimited text files.

Once the rows are imported, The metadata table can be saved.

13.2 Associating data elements with metadata

Each row in a metadata table can be associated with one or more data elements, such as sequence lists, expression tracks, variant tracks, etc. Each data element can be associated with one row of a given metadata table. Once data elements are associated with rows of a metadata table, it is then possible to use that metadata table to find data elements that share particular attributes, launch analyses like expression analyses where sample attributes are key, define batch units such that an analysis runs once per sample, or to group samples together according to their attributes when running certain types of workflows.

Each association has a "Role" label assigned to the associated element, which can be used to indicate the nature of the data element. For example, a newly imported sequence list could be given a role like "Sample data", or "NGS reads".

Associating data with metadata rows can happen in several ways, depending on the circumstances:

- By default, when input data for an analysis is associated with metadata, the results will inherit any unambiguous association. Appropriate role labels are assigned by the analysis

tool. For example, a read mapping tool will assign the role "Unmapped reads" to a sequence list of unmapped reads that it produces.

- By default outputs from a workflow are associated with the relevant metadata rows in workflow results metadata tables. In these tables, the role assigned is always "Result data".
- Manually triggering data associations, either through matching the metadata key column entries with data element names, or by specifying the data element to associate with a given row. Here, roles to apply are chosen by you when triggering the associations.

The rest of this section describes this last point, where you associate data elements to metadata.

To do this, open a metadata table, and then click on the **Associate Data** button at the bottom of the Metadata Table view. Two options are available:

- **Association Data Automatically** Associations are set up based on matches between metadata key column entries and data elements names in a specified location of the Navigation Area. This option is only available if a key column has been specified. See section section [13.2.1](#).
- **Associate Data with Row** Manually make associations row by row, by selecting a row of the metadata and a particular data element in the Navigation Area. Here, information in the metadata table does not need to match data element names. This option is also available when right-clicking a row in the table. section [13.2.2](#).

13.2.1 Associate Data Automatically

When using the **Associate Data Automatically** option, associations are created based on matching the name of the selected data elements with the information in the **key column** of a metadata table previously saved in the Navigation Area. Matching is done according to three possible schemes: Exact, Prefix and Suffix (see section [13.2.1](#)).

To associate data automatically, click the **Associate Data** button at the bottom of the Metadata Table view, and select **Associate Data Automatically**.

Select the data the tool should consider when setting up metadata associations (figure [13.11](#)).

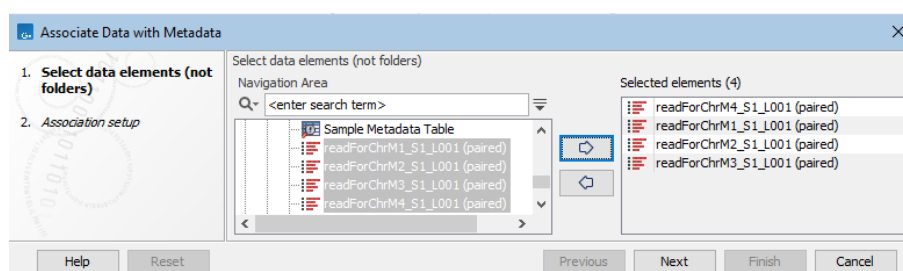


Figure 13.11: Select data elements to be associated to a CLC Metadata Table.

In the Association setup step, you specify whether the matching of the data element names to the entries in the key column should be based on exact or partial matching (described below). A preview showing how elements are matched to metadata rows using the selected matching scheme is shown in the wizard (figure [13.12](#)).

You also specify a role for each element. The default role provided is "Sample data". You can specify any term you wish.

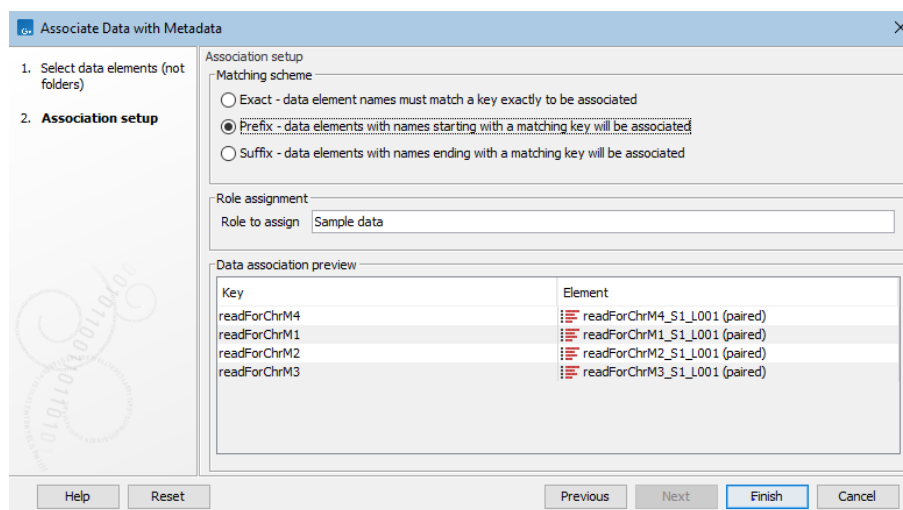


Figure 13.12: Data element names can be matched either exactly or partially to the entries in the key column. Here, the Prefix matching scheme has been selected. A preview showing how elements are matched to metadata rows using that scheme is shown in the Data association preview area, at the bottom.

After the job has run, data associations and roles are saved for all the selected data elements where the name matches a key column entry according to the selected matching scheme.

Note: Data elements selected that already have associations with the CLC Metadata Table will have their associations *updated* to reflect the current information in the CLC Metadata Table. This means associations will be *deleted* for a selected data element if there are no rows in the metadata table that match the name of that data element. This could happen if, for example, you changed the name of a data element with a metadata association, and did not change the corresponding key entry in the metadata table.

Matching schemes A data element name must match an entry in the key column of a metadata table for an association to be set up between that data element at the corresponding row of the metadata table. Two schemes are available in the **Association Data Automatically** for matching up names with key entries:

- **Exact** - data element names must match a key exactly to be associated. If any aspect of the key entry differs from the name of a selected data element, no association will be created.
- **Prefix** - data elements with names partially matching a key will be associated: here the first whole part(s) of a name must match a key entry in the metadata table for an association to be established. This option is explained in more detail below.
- **Suffix** - data elements with names partially matching a key will be associated: here the last whole part(s) of a name must match a key entry in the metadata table for an association to be established. This option is explained in more detail below.

Partial matching rules For each data element being considered, the partial matching scheme involves breaking a data element name into components and searching for the best match from

the key entries in the metadata table. In general terms, the best match means the longest key that matches entire components of the name.

The following describes the matching process in detail:

- Break the data element name into its component parts based on the presence of delimiters. It is these parts that are used for matching to the key entries of the metadata table.

Delimiters are any non-alphanumeric characters. That is, anything that is not a letter (a-z or A-Z) or number (0-9). So, for example, characters like hyphens (-), plus symbols (+), spaces, brackets, and so on, would be used as delimiters.

If partial matching was chosen with a data element called `Sample234-1 (mapped) (trimmed)` would be split into 4 parts: `Sample234`, `-1`, `(mapped)` and `(trimmed)`.

- Matches are made at the component level. A whole key entry must match perfectly to at least the first (with the Prefix option) or the last (with the Suffix option) complete component of a data element name.

For example, a key entry `Sample234` would be a match to the data element with name `Sample234-1 (mapped) (trimmed)` because the whole key entry matches the whole of the first component of the data element name. Conversely, if the key entry had been `Sample23`, no match would be identified, because the whole key entry does not match to at least the whole of the first component of the data element name.

In cases where a data element could be matched to more than one key, the longest key matched determines the metadata row the data will be associated with.

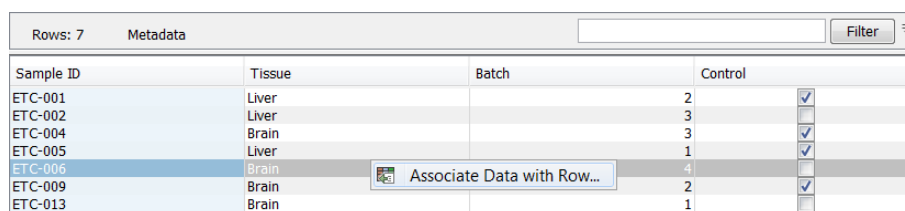
The table below provides examples to illustrate the partial matching system, on a table that has the keys with sample IDs like in figure 13.13) (i.e., `ETC-001`, `ETC-002`, . . . , `ETC-013`),

Data Element	Key	Reason for association
ETC-001 (Reads)	ETC-001	Key ETC-001 matches the first part of the name
ETC-001 un-m. . . (single)	ETC-001	"
ETC-001 un-m. . . (paired)	ETC-001	"
ETC-002	ETC-002	Key ETC-002 matches the whole name
ETC-003	None	No keys match this data element name
ETC-005	ETC-005	Key ETC-005 matches the whole name
ETC-005-1	ETC-005	Key ETC-005 matches the first part of the name
ETC-006-5	ETC-006	Key ETC-006 matches the first part of the name
ETC-007	None	No keys match this data element name
ETC-007 (mapped)	None	"
ETC-008	None	"
ETC-008 (report)	None	"
ETC-009	ETC-009	Key ETC-009 matches the whole name

13.2.2 Associate Data with Row

The **Associate Data with Row** option is best suited for association of a few metadata tables to a few data elements. This type of association does not require a key column in the metadata table, nor a particular relationship between the name of the data element and the metadata to associate it with.

To associate data elements with a particular row in the metadata table, select the desired row in the metadata table by clicking on it. Then either click the **Associate Data** button at the bottom of the Metadata Table view, or right-click on the selected metadata row and choose the **Associate Data with Row** option (as seen in figure 13.13).



Sample ID	Tissue	Batch	Control
ETC-001	Liver	2	<input checked="" type="checkbox"/>
ETC-002	Liver	3	<input type="checkbox"/>
ETC-004	Brain	3	<input checked="" type="checkbox"/>
ETC-005	Liver	1	<input checked="" type="checkbox"/>
ETC-006	Brain	4	<input type="checkbox"/>
ETC-009	Brain	2	<input checked="" type="checkbox"/>
ETC-013	Brain	1	<input type="checkbox"/>

Figure 13.13: Manual association of data elements to a metadata row.

A window will open within which you can select the data elements that should have an association with the metadata row.

If a selected data element already has an association with this particular metadata table, that association will be updated. Associations with any other metadata tables will be left as they are.

Enter a role for the data elements that have been chosen and click **Next** until you can choose to **Save** the outputs. Data associations and roles will be saved for the selected data elements.

13.3 Working with data and metadata

13.3.1 Finding data elements based on metadata

Data elements associated with rows of the metadata table can also be found from within a Metadata Table view. From there, it is possible to highlight elements in the Navigation Area and launch analyses on selected data.

Relevant metadata tables can be found using the Quick Search box, described in section 3.4.1.

To find data elements associated with selected metadata rows in a metadata table:

- Select one or more rows of interest in the metadata table.
- Click on the **Find Associated Data** button at the bottom of the view.

A table with a listing of the data elements associated to the selected metadata row(s) will appear (figure 13.14).

The search results table shows the type, name, and navigation area path for each data element found. It also shows the key entry of the metadata table row with which the element is associated and the role of the data element for this metadata association. In figure 13.14, there are five data elements associated with sample ETC-009. Three are Sequence Lists, two of which have a role that tells us that they are unmapped reads resulting from the Map Reads to Reference tool.

Clicking the **Refresh** button will re-run the search and refresh the search results table.

Click the button labeled **Close** to close the search table view.

Data elements listed in the search result table can be opened by clicking on the button labeled **Show** at the bottom of the view.

Rows: 7 Metadata

Sample ID	Tissue	Batch	Control	
ETC-001	Liver		2	<input checked="" type="checkbox"/>
ETC-002	Liver		3	<input type="checkbox"/>
ETC-004	Brain		3	<input checked="" type="checkbox"/>
ETC-005	Liver		1	<input checked="" type="checkbox"/>
ETC-006	Brain		44	<input type="checkbox"/>
ETC-009	Brain		2	<input checked="" type="checkbox"/>
ETC-013	Liver		1	<input type="checkbox"/>

Buttons: Set Up Table..., Manage Data..., Find Associated Data, Associate Data..., Additional Filtering

Rows: 8 Metadata Elements

Sample ID	Role	Type	Name	Path
ETC-001	Sample data		ETC-001 (Reads)	CLC_Data / Metadata / Example
ETC-001	Sample data		ETC-001 un-mapped reads (single)	CLC_Data / Metadata / Example
ETC-001	Sample data		ETC-001 un-mapped reads (paired)	CLC_Data / Metadata / Example
ETC-009	Un-mapped reads		BC_9_L001_R1 (paired) un-mapped reads [BC_9_L001_R1] (single)	CLC_Data
ETC-009	Un-mapped reads		BC_9_L001_R1 (paired) un-mapped reads [BC_9_L001_R1] (paired)	CLC_Data
ETC-009	Mapping Report		BC_9_L001_R1 (paired) mapping summary report	CLC_Data
ETC-009	Read mapping		BC_9_L001_R1 (paired) mapping	CLC_Data
ETC-009	Sample data		ETC-009	CLC_Data / Metadata / Example

Buttons: Find in Navigation Area, Show, Refresh, Close

Figure 13.14: Metadata Table with search results

Alternatively, they can be highlighted in the Navigation Area by clicking the **Find in Navigation Area** button.

Analyses can be launched on the selected data elements:

- Directly. Right click on one of the selected elements, choose the menu option Tools, and navigate to the tool of interest. The data selected in the search results table will be listed as selected elements in the launch wizard.
- Via the Navigation area selection. Use the **Find in Navigation Area** button and then launch a tool or workflow. The items that were selected in the Navigation area will be pre-selected in the launch wizard.

If no data elements with associations are found and this is unexpected, please re-index the locations your data are stored in. This is described in section 3.4. For data held in a CLC Server location, an administrator will need to run the re-indexing. Information on this can be found in the CLC Server admin manual at https://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=Rebuilding_index.html.

13.3.2 Viewing metadata associations

Metadata associations for a data element are shown using the Element info view, as in figure 13.15.

To show Element Info,

right-click an element in the Navigation Area | Show | Element Info (🔍)

The Element Info view contains the details of each metadata association for the data element. The following operations are available:

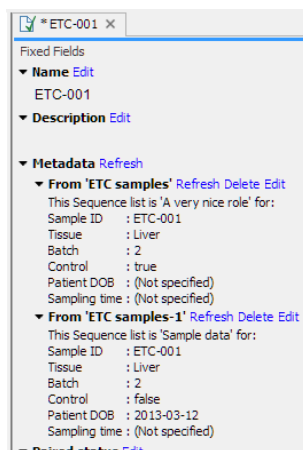


Figure 13.15: *Element Info view with a metadata association*

- **Delete** will remove an association.
- **Edit** will allow you to change the role of the metadata association.
- **Refresh** will reload the metadata details from the Metadata Table; this functionality may be used to attempt to re-fetch metadata that was previously unavailable, e.g. due to server connectivity.

Read more about Element Info view in section [15.4](#).

13.3.3 Removing metadata associations

Any or all associations to data elements from rows of a metadata table can be removed by taking the following steps:

1. Open the metadata table containing the rows of interest.
2. Highlight the relevant rows of the metadata table.
3. Click **Find Associated Data**.
4. In the Metadata Elements table that opens, highlight the rows for the data elements the metadata associations should be removed from.
5. Right-click over the highlighted area and choose the option **Remove Association(s)** (figure [13.16](#)). Alternatively, use the Delete key on the keyboard, or on a Mac, the fn and backspace keys at the same time.

Metadata associations can also be removed from within the Element info view for individual data elements, as described in section [13.3.2](#).

When an metadata association is removed from a data element, this update to the data element is automatically saved.

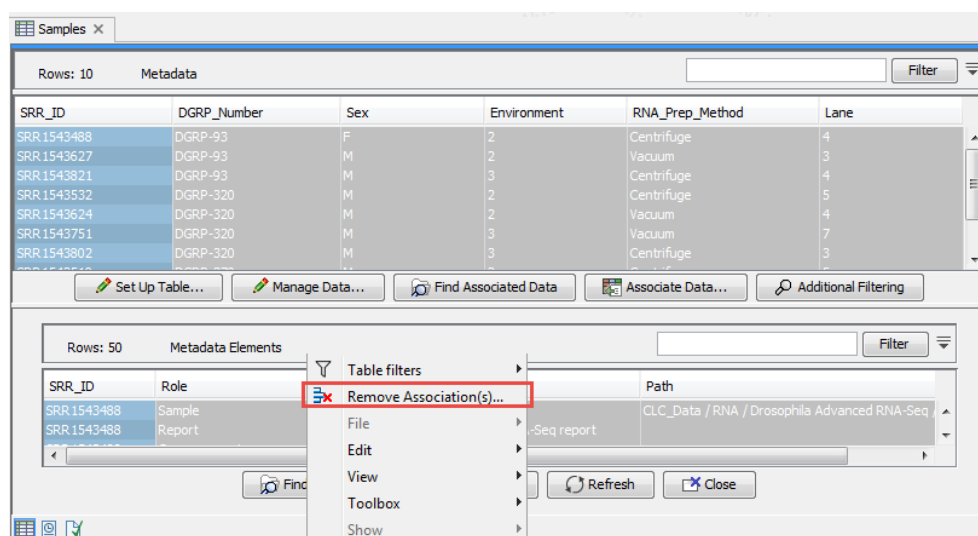


Figure 13.16: Removing metadata associations to two data elements via the Metadata Elements table.

13.3.4 Identifying metadata rows without associated data

Using the Metadata Table view you can apply filters using the standard filtering tools shown at the top of the view as well as by using special metadata filtering in the **Additional Filtering** shown at the bottom. Using the special metadata filtering option **Show only Unassociated Rows**, you can filter the rows visible in the Metadata Table view so only the rows to which no data elements are associated are shown. If desired, these rows could then be used to launch one of the tools for associating data, described in section 13.2.

This filter can be cancelled by clicking on the **Additional Filtering** button again and clicking on the **Show only Unassociated Rows** again. When the filter is active, it has a checkmark beside it. When it is inactive, it does not.

This filter can take a long time if many rows are shown in the table. When working with many rows, it can help if the full table is filtered using the general filters in advance, using the standard filters at the top of the table view. Alternatively you can pre-select some rows and filtering with the Additional filtering option **Filter to Selected Rows**. This filter can be applied multiple times. If the search takes too long, you can cancel it by unselecting the filter from the menu.

This filter can be cancelled by clicking on the **Additional Filtering** button again and clicking on the **Clear Selection Filter** option.

13.3.5 Editing Metadata tables

CLC Metadata Tables can be edited using functionality under the **Edit Table...** button at the bottom of the Metadata Table editor (figure 13.17) and by right-clicking on a row or rows and selecting from the options presented (figure 13.18). Information can be added directly, or can be take from another CLC Metadata Table or an Excel, CSV or TSV format file.

Edit existing rows in a CLC Metadata Table

A dialog to edit entries in a CLC Metadata Table can be opened by:

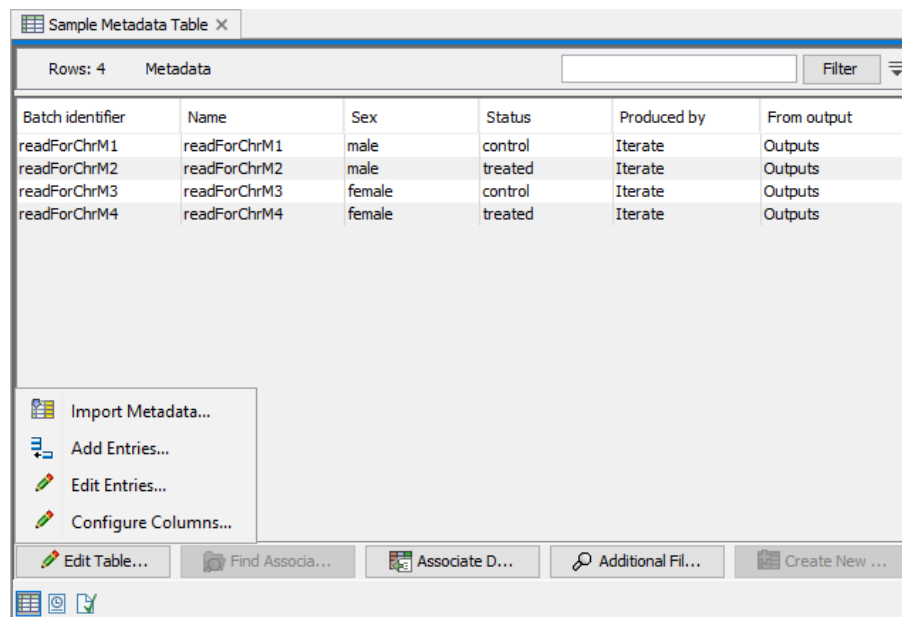


Figure 13.17: Click on the *Edit Table...* button to open a menu with options for adding, editing or removing information in a CLC Metadata table.

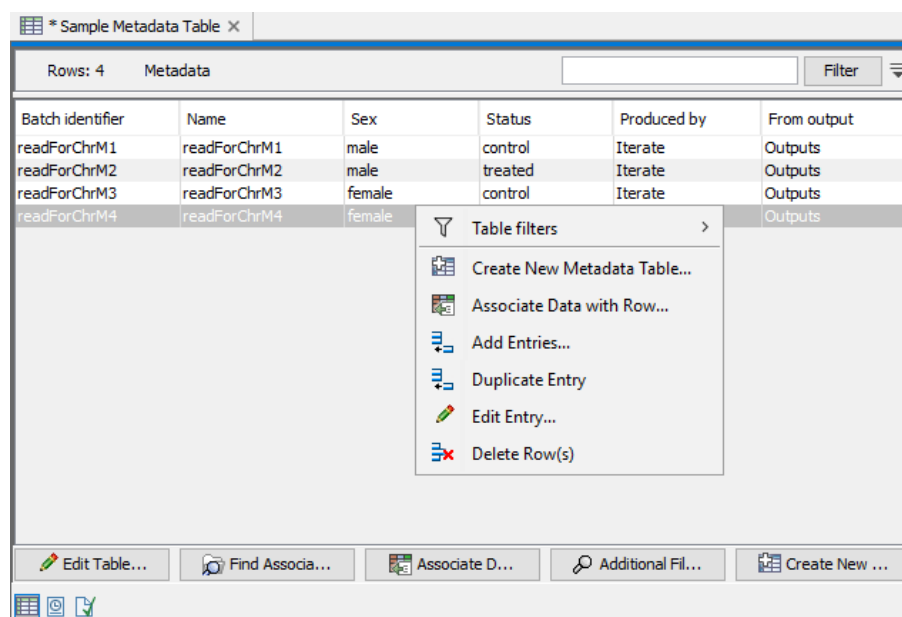


Figure 13.18: Right-click on selected rows of a CLC Metadata Table to open a menu actions that can be taken.

- Clicking on the **Edit Table...** button and selecting **Edit Entries..** (✎).
- Double clicking on an entry.
- Selecting an entry and hitting the Return key.

Navigate between entries using the buttons on the right. Modifications made take effect as you navigate to another row, or if you close the dialog using **Done**.

Right-click on an individual row in the table and select the **Edit Entry..** (✎) option to edit just that entry. An option to delete rows is also in this menu: **Delete Row(s)** (figure 13.18).

Import further rows into a CLC Metadata Table

More information can be imported into an existing CLC Metadata table by clicking on the **Edit Table...** button and selecting **Import Metadata...** (📁).

Information can be supplied in an Excel, CSV or TSV file, by clicking on the standard folder icon (📁), or from another CLC Metadata Table, by clicking on the browse folder (🔍) icon.

For CLC Metadata Tables with a key column, you choose whether new information should be added to existing entries, and whether rows should be added for new entries (figure 13.19). Matching is based on column names.

If there is no key column designated in the original CLC Metadata Table, as is the case in Workflow Result Metadata tables, then data from the new source can only be entered into new rows. A key column can be specified using the **Configure Columns** functionality described in the **Add or edit columns in a CLC Metadata Table** section below.

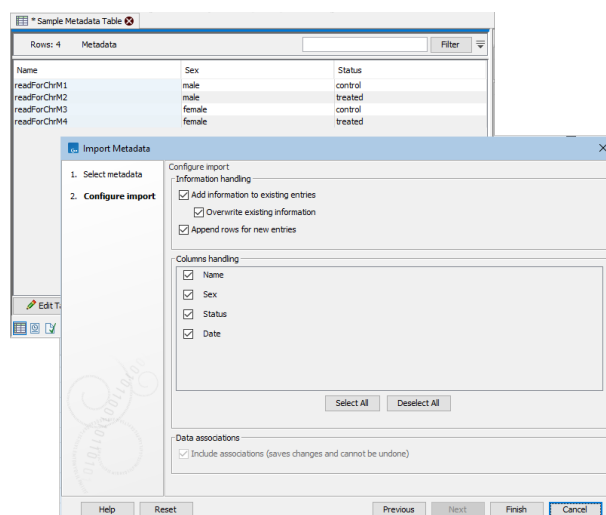




Figure 13.19: Additional information can be imported to an existing CLC Metadata table. You can choose whether new information should be added to existing entries, and whether rows should be added for new entries. The columns to import can also be specified.



Add and delete rows manually

Rows can be added to the end of the CLC Metadata table by clicking on the **Edit Table...** button and selecting **Add Entries...** (📁). This option is also available in the menu that opens when you right-click on a row in the table (figure 13.18).

The option **Duplicate Entry**, available when you right-click on a row in the table, will duplicate that row and put it at the bottom of the table. When a key column has been designated, you will be prompted for a new value for that column for the new row. Data with an association to the original row will *not* automatically have an association to the new row. (A given data element can only have one association to a given CLC Metadata Table.)

Individual rows can also be added using the  button, which inserts a new row after the current one.



Rows may be deleted using the  button.




The  and  buttons are used to undo and redo changes respectively.



Add or edit columns in a CLC Metadata Table

Existing columns can be edited, and new columns added, by clicking on the **Edit Table...** button and selecting **Configure columns....**

A dialog opens with information about the left-most column in the CLC Metadata Table (figure 13.20). Use the buttons on the right to navigate to other columns and to add or delete columns.

Navigate between the columns using the  **Prev** and  **Next** buttons, or by using left/right arrow keys with Alt key held down. Modifications made to a particular column take effect as you navigate to another column, or if you close the dialog using **Done**.

Individual columns can be added using the  and  buttons, which insert new columns before and after the current column respectively. Columns may be deleted using the  button.

The  and  buttons are used to undo and redo changes respectively.

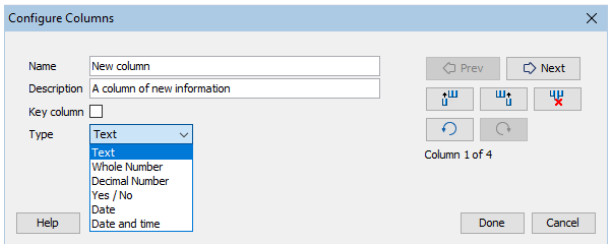
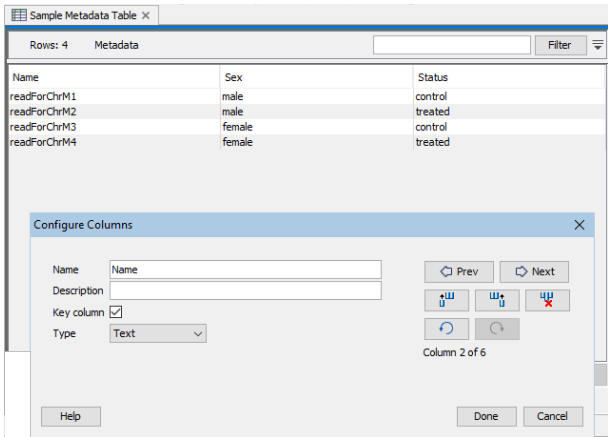


Figure 13.20: When adding a new column, a name, description and data type is specified. If it should become the key column, the Key column box should be checked. Use the buttons on the right to navigate to other columns or add further new columns.



Name	Sex	Status
readForChrM1	male	control
readForChrM2	male	treated
readForChrM3	female	control
readForChrM4	female	treated

Figure 13.21: The Name column has been designated as the key column.

For each column, the following can be configured:

- **Name.** A mandatory header name or title for the column.
- **Description.** An optional description of the information that will be held in the column. The description will appear as a tool tip, visible when you hover the mouse cursor over the column name in the metadata table.
- **Key column.** Any column containing only unique values can be designated as the key column. If a table already has a key column, this option is disabled for other columns. Information in the key column is used when automatically creating associations from data elements, described in (section section 13.2.1).
- **Type.** The type of value allowed. The default data type for columns on import is text, but this can be edited to the following types:
 - **Text** Simple text.
 - **Whole number** Integer values, like 42 or -7.
 - **Decimal number** Decimal values, like 3.14 or 1.72e13.
 - **Yes / No** Yes/No or True/False values are accepted. Capitalization is not necessary.
 - **Date** Local dates such as 2015-04-23 for April 23rd, 2015.
 - **Date and time** Local date and time such as 2015-04-23 13:37 for 1:37pm on April 23rd, 2015. Note the use of 24-hour clock and that no time zone information is present.

13.4 Moving, copying and exporting metadata

Moving and copying metadata

This section focuses on what happens to data associations when copying metadata tables. This information also pertains to moving metadata tables *between* File Locations, as this is equivalent to a copy action.

To copy a metadata table and the data associated with it, such that the new data element copies are associated with the copy of the metadata table, *select both the metadata table and the data elements and copy them in a single operation*. Of note here:

- The data element copies will have associations with the new copy of the metadata table. The original elements keep their associations with the original metadata table.
- If a metadata table is copied but data elements with associations to it are not also copied in that action, those data elements will be associated with both the copy and the original metadata table.
- If data elements with associations to metadata are copied, but no metadata table is involved in the same copy action, each data element copy will be associated to the same metadata as the original element.

If a metadata table and some, but not all, data elements with associations to it, are copied in a single action, then:

- The data element copies will have associations to the copy of the metadata table, while the original elements (that were copied) remain associated with the original metadata table.
- Elements with associations to the original metadata table that were not copied will have associations to both the original metadata table and the copy. However, if these data elements are later copied (in a separate copy operation), those copies will only be associated with the original metadata table. If they should be associated with the copy of the metadata table, those association must be added as described in section 13.2.

Exporting metadata

The standard Workbench export functionality can be used to export metadata tables to various formats. The system's default locale will be used for the export, which will affect the formatting of numbers and dates in the exported file.

See section 8.1 for more information.

Chapter 14

Workflows

Contents

14.1 Creating and editing workflows	272
14.1.1 Adding elements to a workflow	273
14.1.2 Connecting workflow elements	275
14.1.3 Ordering inputs	277
14.1.4 Validating a workflow	278
14.1.5 Viewing the flow of elements in a workflow	280
14.1.6 Adjusting the workflow layout	281
14.1.7 The Configuration Editor view	281
14.1.8 Snippets in workflows	282
14.1.9 Customizing the Workflow Editor	287
14.2 Workflow elements	292
14.2.1 Anatomy of workflow elements	292
14.2.2 Basic configuration of workflow elements	294
14.2.3 Configuring Workflow Input elements	298
14.2.4 Configuring Workflow Output and Export elements	305
14.2.5 Control flow elements	309
14.2.6 Track lists as workflow outputs	324
14.2.7 Input modifying tools	324
14.3 Launching workflows individually and in batches	325
14.3.1 Workflow Result Metadata tables	327
14.3.2 Running workflows in batch mode	328
14.3.3 Running part of a workflow multiple times	332
14.3.4 Specifying reference data in the launch wizard	333
14.4 Advanced workflow batching	338
14.4.1 Batching workflows with more than one input changing per run	339
14.4.2 Multiple levels of batching	341
14.5 Template workflows	342
14.5.1 Import with Metadata	343
14.5.2 Prepare Raw Data	344
14.5.3 De Novo Assemble Long Reads and Polish with Short Reads	347

14.5.4	Identify DNA Germline Variants workflow	349
14.5.5	RNA-Seq and Differential Gene Expression Analysis workflow	353
14.5.6	Trim and Map Sanger Sequences	357
14.6	Managing workflows	358
14.6.1	Updating workflows	361
14.6.2	Workflow installation	364
14.6.3	Using workflow installation files	367
14.7	QIAseq Panel Analysis Assistant	368
14.7.1	Adding QIAseq analyses	370
14.7.2	Reference data for QIAseq analyses	370
14.7.3	Running a QIAseq analysis	373
14.7.4	Non-standard parameters in QIAseq analyses	374
14.7.5	Configuring QIAseq analyses	374
14.7.6	QIAseq custom panels	376

Workflows contain tools linked together, making up an analysis pipeline.

Data from CLC Data Locations can be selected as input, as can raw data in supported standard formats. Results can be saved to CLC locations for viewing and downstream analysis, or exported to standard formats and saved to a location of your choice.

The flow of data through a workflow can be defined in various ways, making it simple to execute complex analyses consistently. Launching workflows in Batch mode, the same analysis is run on many samples using a single launch action.

Workflows are created and customized using the Workflow Editor (figure 14.1). Workflows can be saved and re-opened, just like data elements. You can run workflows from the Workflow Editor, or install them and run them from under the Installed Workflows folder under the Workflows menu (see section 14.6.2).

Template workflows are pre-installed in the *CLC Genomics Workbench*, and can be referred to as examples, or copied and edited. This is described further in section 14.5.

Template workflows are also provided by several plugins.

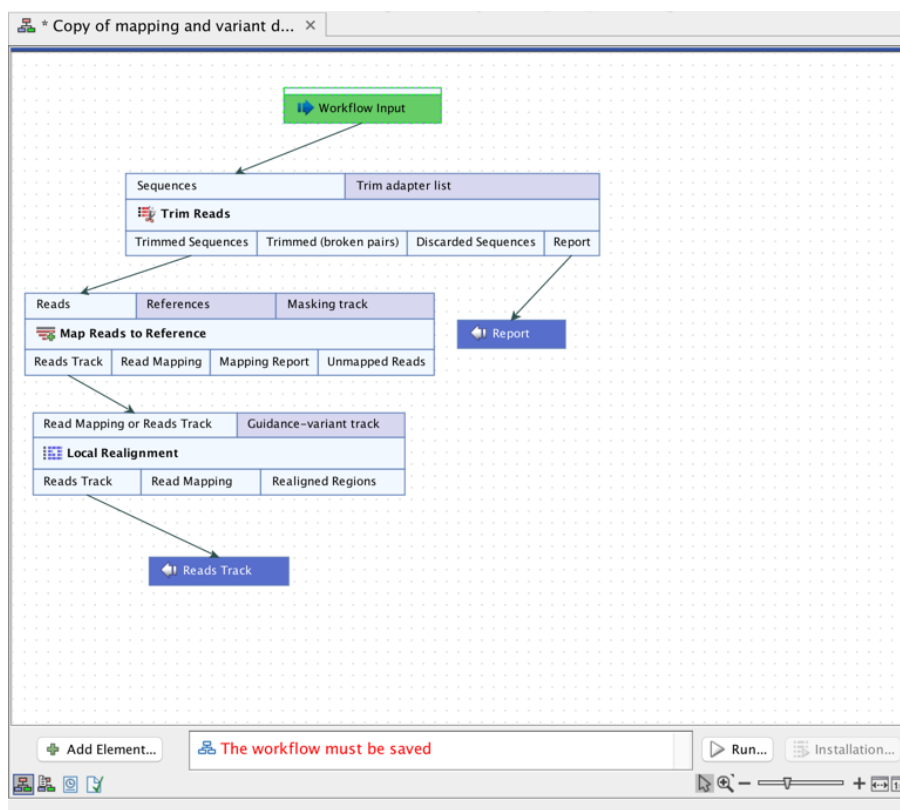


Figure 14.1: A workflow open in the Workflow Editor. Workflows consist of connected tools, where the output of one tool is used as input for another tool.

14.1 Creating and editing workflows

Workflows are created and edited using the Workflow Editor.

To create a new workflow, click on the **Workflows** button () in the Toolbar and then select "New Workflow" ().

Alternatively, use the menu option:

File | New | Workflow ()

To open a copy of an installed workflow in the Workflow Editor, right-click on the workflow name under the Workflows menu in the Toolbox at the bottom, left side of the Workbench, and choose the option **Open Copy of Workflow**. Customizing existing workflows rather than building new workflows from scratch, can save much time.

Template workflows are provided with the *CLC Genomics Workbench* (figure 14.2). Template workflows are also provided by several plugins.

To copy the image of a workflow design, select the elements in the workflow design (click in the workflow editor and then press keys Ctrl + A), then copy (Ctrl + C), and then paste where you wish the image to be placed, for example, in an email or presentation program.

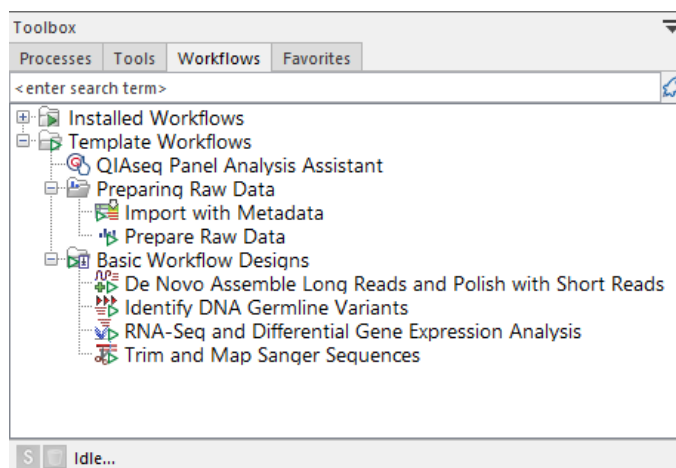


Figure 14.2: Template workflows are available under the Workflows menu.

14.1.1 Adding elements to a workflow

Elements are the building blocks of workflows. They are used to define the inputs, the outputs, all the analysis steps to be done, and can be used to control the way data flows through the workflow. Workflow elements are defined in detail in section 14.2. Here we focus on adding them and connecting them to each other within a workflow.

To add elements to a workflow:

- Drag tools from the **Tools** tab in the Toolbox panel in the bottom, left side of the Workbench into the canvas area of the Workflow Editor, or
- Use the **Add Element** dialog (figure 14.3). The following methods can be used to open this dialog:
 - Click on the **Add Element** (✚) button at the bottom of the Workflow Editor.
 - Right-click on an empty area of the canvas and select the **Add Element** (✚) option.
 - Use the keyboard shortcut Shift + Alt + E.

Select one or more elements and click on **OK**. Multiple elements can be selected by keeping the Ctrl key (⌘ on Mac) depressed while selecting them.

- Use one of the relevant options offered when right-clicking on an input or output channel of a workflow element, as shown in figure 14.4 and figure 14.5.
 - **Connect to Workflow Input** or **Connect to Configured Workflow Input** and
 - **Use as Workflow Output** to add data to be processed or saved.
 - **Add Element to be Connected...** to open the Add Elements pop-up dialog described above.
 - **Connect to Iterate** and **Connect to Collect and Distribute** to create an iterative process within a workflow.

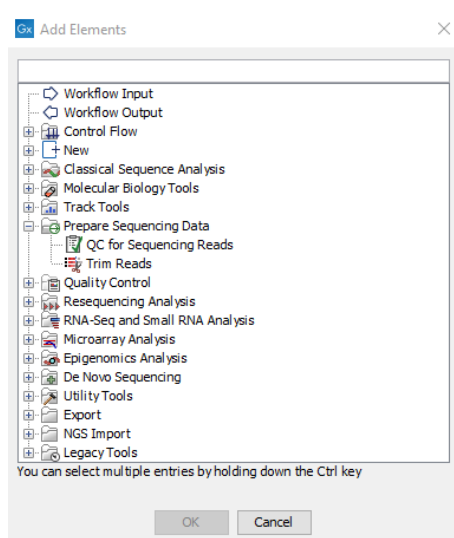


Figure 14.3: Adding elements to a workflow.

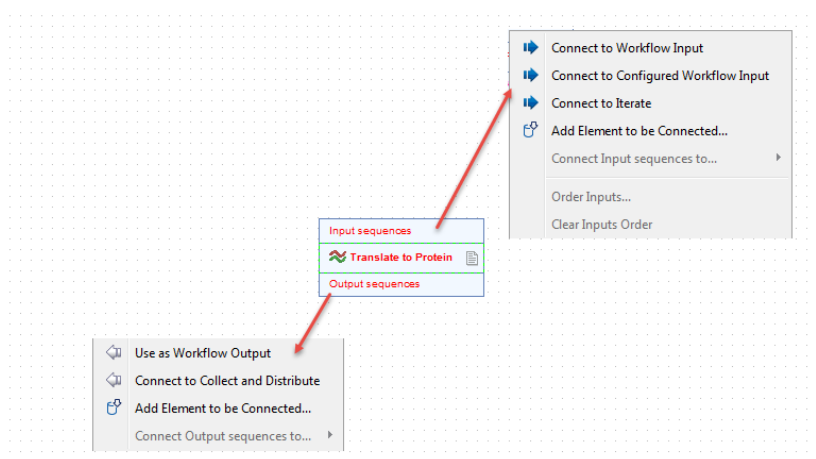


Figure 14.4: Connection options are shown in menus when you right click on an input or output channel of a workflow element.

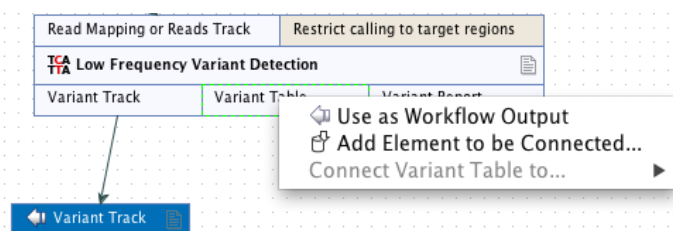


Figure 14.5: Right clicking on an output channel brings up a menu with relevant connection options.

Once added, workflow elements can be moved around on the canvas using the 4 arrows icon (↔) that appears when hovering on an element.

Workflow elements can be removed by selecting them and pressed the delete key, or by right-clicking on the element name and choosing **Remove** from the context specific menu, as shown in figure 14.6.

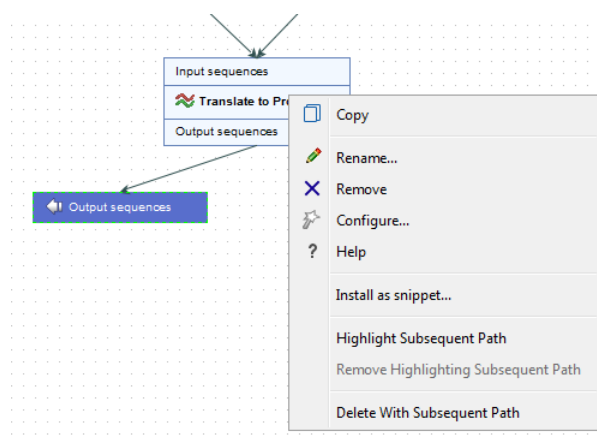


Figure 14.6: Right clicking on an element name brings up a context specific menu that includes options for renaming or removing elements.

14.1.2 Connecting workflow elements

Connections between workflow element output channels and input channels define where data flows from and to.

The names of output channels usually indicate the type of data generated and the names of input channels usually indicate the type of data expected. Connections can only be made between compatible output and input channels.

An output channel can be connected to more than one input channel and an input channel can accept data from more than one output channel (figure 14.7).

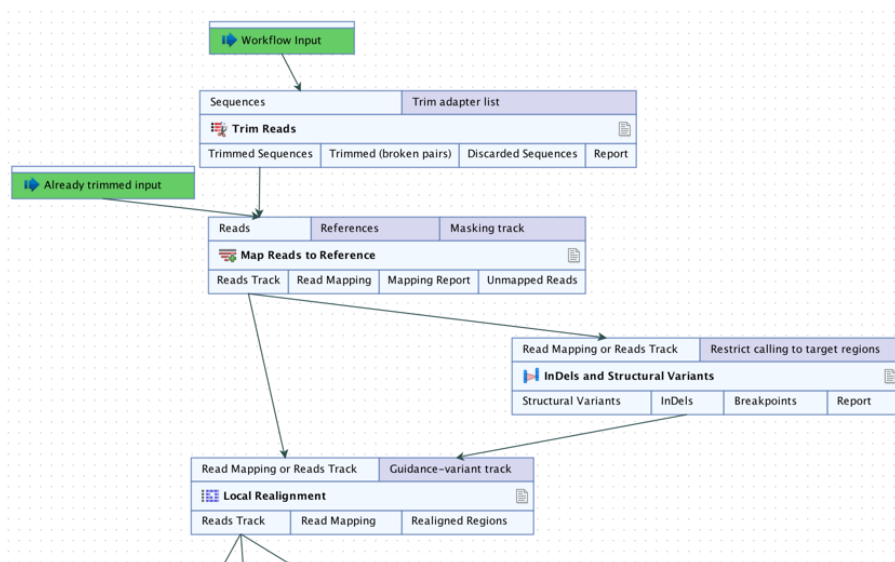


Figure 14.7: In this workflow, two elements are supplying data to the Reads input channel of the Map Reads to Reference element, while data from the Reads Track output channel of Map Reads to Reference is being used as input to two elements.

Connecting output and input channels

Two ways that compatible input and output channels can be connected are:

- Click on an output channel and, keeping the mouse button depressed, drag the cursor to the desired input channel. A green border around the input channel name indicates when the connection has been made and the mouse button can be released. An arrow is drawn, linking the channels (figure 14.8).

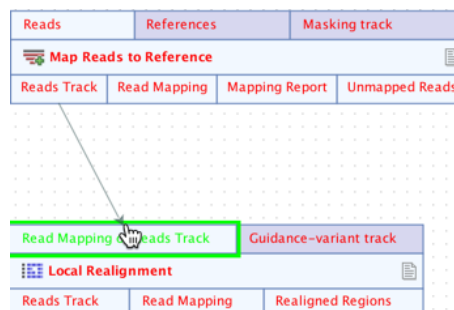


Figure 14.8: Connecting the "Reads Track" output channel from a Map Reads to Reference element to the "Read Mapping or Reads" input channel of a Local Realignment element.

- Use the **Connect <channel name> to...** option in the right-click menu of an output or input channel. Hover the cursor over this option to see a list of elements in the workflow with compatible channels. Hovering the cursor over any of these items then shows the particular channels that can be connected to (figure 14.9).

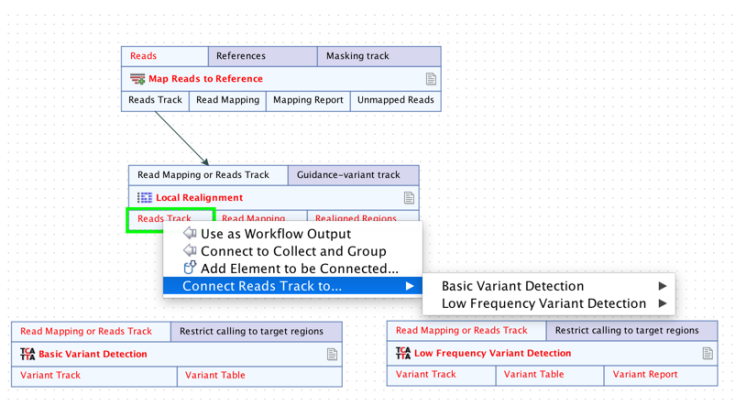


Figure 14.9: Right-clicking on an output channel displays a context specific menu, with options supporting the connection of this channel to input channels of other workflow elements.

Information about what elements and channels are connected In a small workflow, it is easy to see which elements are connected and how they are connected. In large workflows, the following methods can be helpful:

- Mouse-over the connection line. A tooltip is revealed showing the elements and channels that are connected (figure 14.10).
- Right-click on a connection line and choose the option **Jump to Source** to see the upstream element or **Jump to Destination** to see the downstream element (figure 14.11).

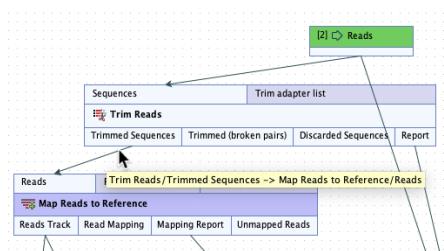


Figure 14.10: Hover the mouse cursor over a connection to reveal a tooltip containing the names of the elements and channels connected.

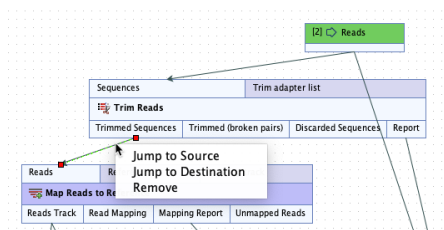


Figure 14.11: Right-click on a connection to reveal options to jump to the source element or the destination element of that connection.

Removing connections

To remove a connection, right-click on the connection and select the **Remove** option (figure 14.11).



14.1.3 Ordering inputs

Workflow inputs can be re-ordered when editing a workflow. This can affect the user experience or can have an effect on the contents of outputs generated by the workflow.

1. **Improve the user experience by changing the order that launch wizard steps are presented.**

By default, the order of the launch wizard steps reflects the order that Input elements were added to the workflow when it was created.

To make changes to this order, right-click on an empty area of the canvas and choose **Order Workflow Inputs...** from the menu that appears.

An **Order Inputs** dialog appears (figure 14.12). Select an input and move it up or down in the list by clicking on the up arrow () or down arrow (), respectively.

A number next to an Input element's name indicates its position in the order. These numbers are updated when the ordering is updated.



These same numbers can be used in Output element naming patterns (see section 14.2.4). If output names using such patterns have already been configured, they may need to be updated.

2. **Influence the content of outputs in cases where input processing order has an effect.**

For example, the order of the sections in a report generated by **Combine Reports** reflects the order that inputs to that tool are processed.

By default, the main inputs to a tool are processed in the order that the connections to that input channel were added when the workflow was created.

To make changes to this order, right-click on the relevant input channel and choose the option **Order Inputs...** from the menu that appears.

An **Order Inputs** dialog appears (figure 14.12). Select an input and move it up or down in the list by clicking on the up arrow () or down arrow (), respectively.

Numbers on the connection arrows are added or updated if any changes are made in this **Order Inputs...** dialog.

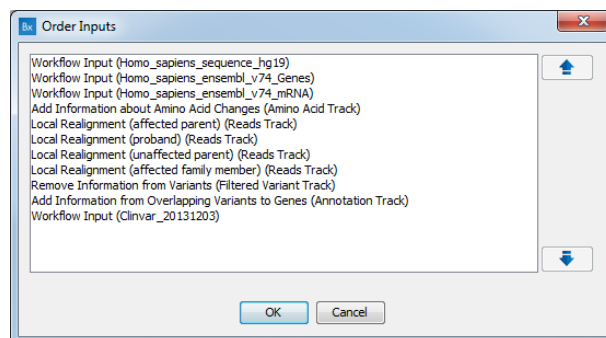


Figure 14.12: The order of inputs is displayed and can be updated in an Order Inputs dialog.

14.1.4 Validating a workflow

A workflow must be valid before it can be run or saved.

In general terms, a valid workflow has at least one route for data to flow into it, at least one result saved from it, and it contains no unconnected elements. In more detail:

- There must be at least one Input element connected to the main input channel of the element where data starts its flow through the workflow. Where there are multiple independent arms in the workflow, this requirement pertains to each of those arms.
- There must be at least one result saved from the end of each branch within a workflow. In practice this means that at least one Output or Export element must be connected to each terminal element with an output channel.
- All elements must have at least one connection to another element in the workflow.

Validation status is continuously monitored, with messages relating to this reported at the bottom of the editor.

The validation status of a workflow will fall into one of three categories:

1. **Valid and saved** When a workflow is valid and has been saved, the message "Validation successful" is displayed in green text at the bottom of the editor (figure 14.13).

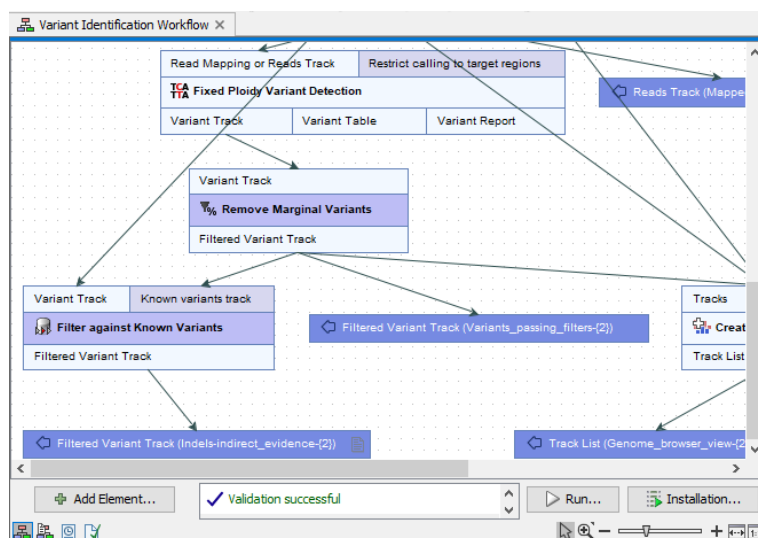


Figure 14.13: The "Validation successful" message indicates that this workflow is valid and has been saved.

2. **Valid, with changes not yet saved** When a workflow is valid but there are unsaved changes, a single message is displayed at the bottom of the editor saying "The workflow must be saved". The unsaved state is also represented by the asterisk in the name of the tab (figure 14.14).

Valid workflows can be run before they are saved, allowing changes to be tested before overwriting any previously saved version.

The **Installation...** button is enabled when a workflow is valid and has been saved. See section or chapter 14.6.2 for information about workflow installation.

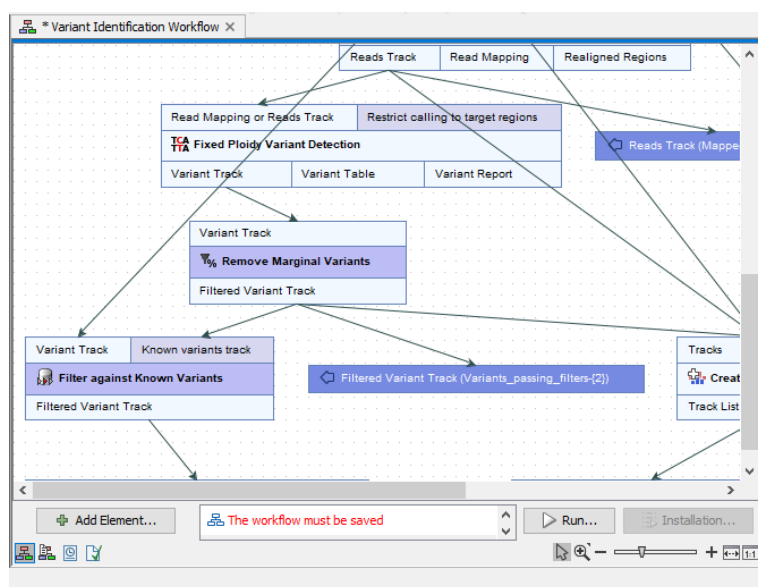


Figure 14.14: This workflow has changes that have not yet been saved, as indicated by the message at the bottom of the editor and the asterisk beside the workflow name in the tab at the top.

3. **Invalid** Each problem in a workflow is reported at the bottom of the editor (figure 14.15).

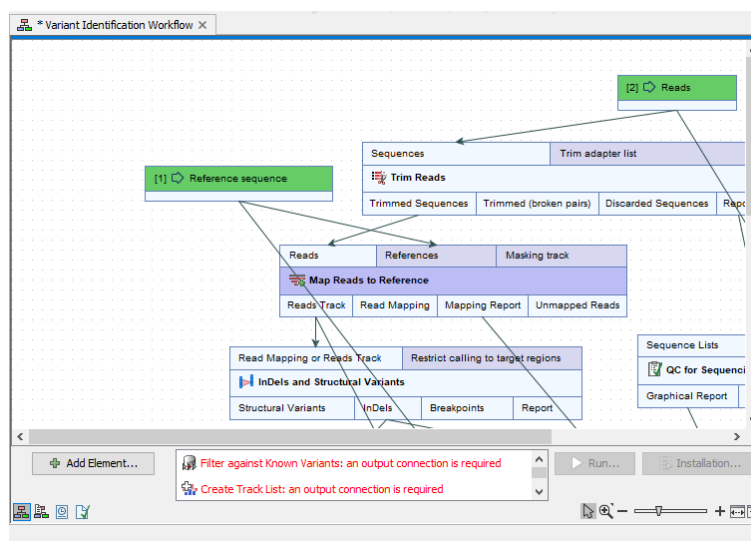


Figure 14.15: Problems are reported at the bottom of the workflow editor.

Clicking on a message about a specific element redirects the focus within the editor to that element (figure 14.16).

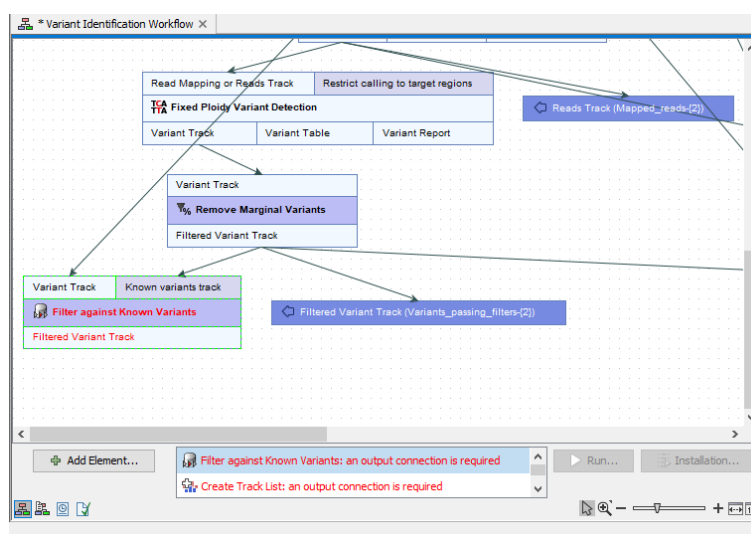


Figure 14.16: Clicking on the error message about *Filter against Known Variants* at the bottom of the editor moved the focus in the editor to that element.

14.1.5 Viewing the flow of elements in a workflow

Following the path through a workflow from a particular element can help when authoring complex workflows. To highlight the path through the workflow from a particular element, right-click on the element name and select the **Highlight Subsequent Path** option from the context specific menu (figure 14.17). Select **Remove Highlighting Subsequent Path** to remove the highlighting.

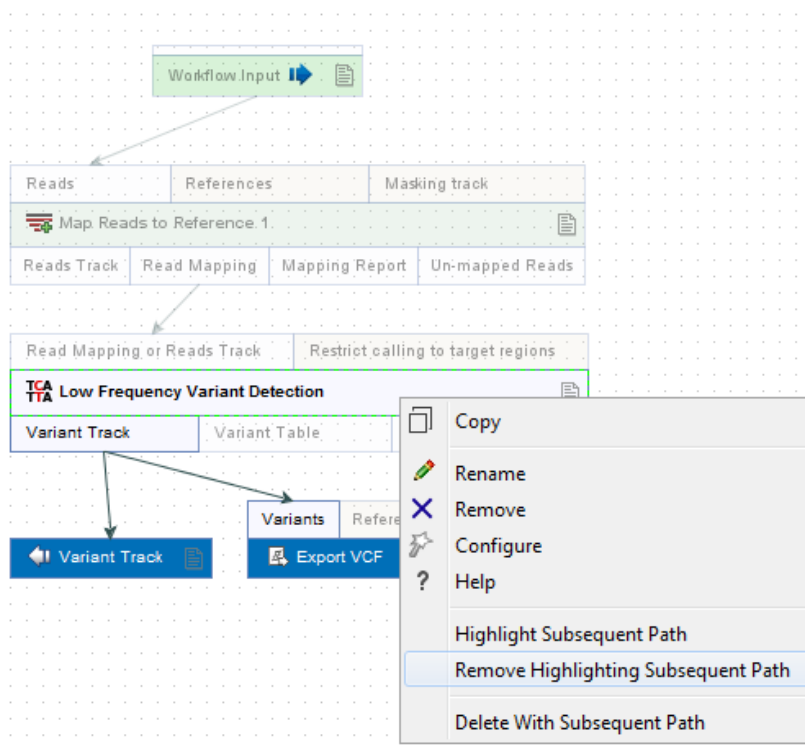


Figure 14.17: All elements connected downstream of a selected element are highlighted after selecting the *Highlight Subsequent Path* menu option.

14.1.6 Adjusting the workflow layout


The layout of elements within a workflow can be adjusted manually or automatically.

- **Manually:** Select one or more workflow elements and then, with the left mouse button depressed, drag these elements to where you want them to be on the canvas.
- **Automatically:** Right-click anywhere on the canvas and choose the option "Layout" (figure 14.18), or use the quick command Shift + Alt + L. The layout of all connected elements in the workflow will be adjusted.

See also section 14.1.9 for information about the **Auto Layout** setting. When enabled that setting causes the layout to be adjusted automatically every time an element is added and connected.

14.1.7 The Configuration Editor view

The **Configuration Editor** view allows the configuration of all of the settings in a workflow through a single window (figure 14.19). This can be more convenient than configuring each workflow element individually, especially for experienced workflow authors.

Click on the  icon located in the lower left corner of the Workflow Editor to open this view.

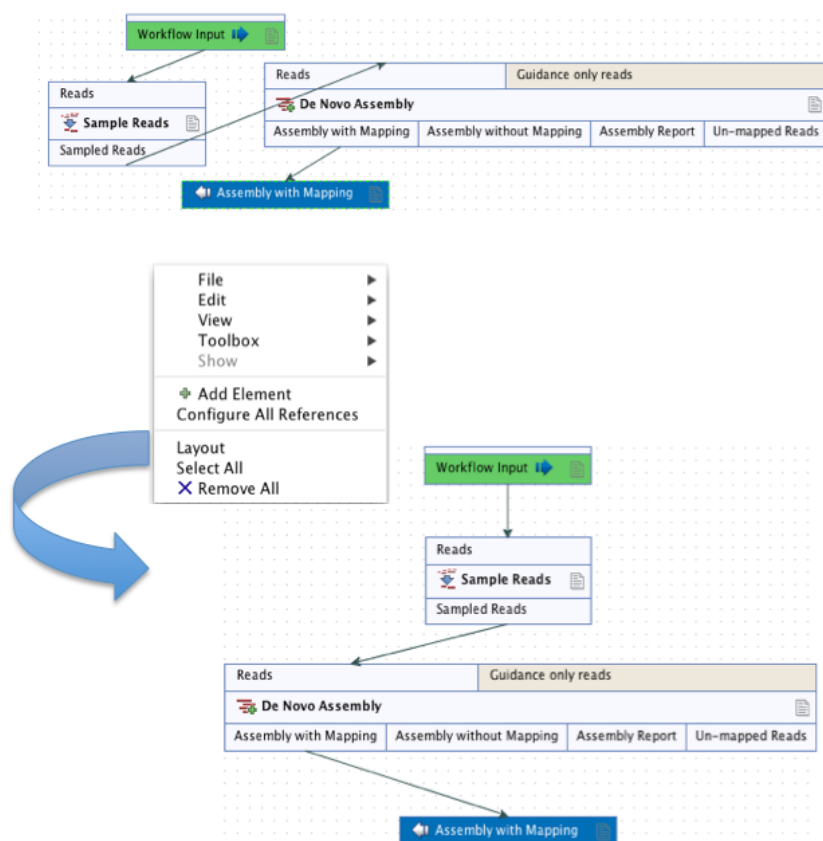


Figure 14.18: The alignment of workflow elements can be improved using the "Layout" function.

14.1.8 Snippets in workflows

When creating a new workflow, you will often have a number of connected elements that are shared between workflows. These components are called snippets. Instead of building workflows from scratch it is possible to reuse components of an existing workflow.

Snippets can be created from an existing workflow by selecting the elements and the arrows connecting the selected elements. Next, you must right-click in the center of one of the selected elements. This will bring up the menu shown in figure 14.20.

When you have clicked on "Install as snippet" the dialog shown in figure 14.21 will appear. The dialog allows you to name the snippet and view the selected elements that are included in the snippet. You are also asked to specify whether or not you want to include the configuration of the selected elements and save it in the snippet or to only save the elements in their default configuration.

Click **OK**. This will install your snippet and the installed snippet will now appear in the **Side Panel** under the "Snippets" tab (see figure 14.22)

Right-clicking on the installed snippet in the **Side Panel** will bring up the following options (figure 14.23):

- **Add**. Adds the snippet to the current open workflow
- **View**. Opens a dialog showing the snippet, which allows you to see the structure

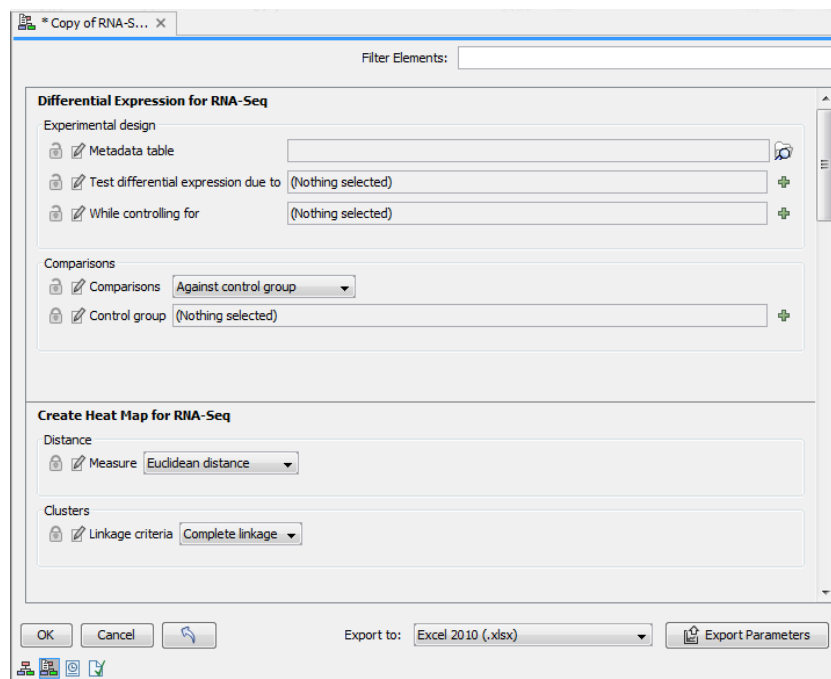


Figure 14.19: Use the Configuration Editor to edit configurable parameters for all the tools in a given Workflow.

- **Rename.** Allows renaming of the snippet.
- **Configure.** Allows to change the configuration of the installed snippet.
- **Uninstall.** Removes the snippet.
- **Export.** Exports the snippet to ones computer, allowing to share it.
- **Migrate.** Updates the snippet (if required).

If you right-click on the top-level folder you get the options shown in figure 14.24:

- **Create new group.** Creates a new folder under the selected folder.
- **Remove group.** Removes the selected group (not available for the top-level folder)
- **Rename group.** Renames the selected group (not available for the top-level folder)

In the **Side Panel** it is possible to drag and drop a snippet between groups to be able to rearrange and order the snippets as desired. An exported snippet can either be installed by clicking on the 'Install from file' button or by dragging and dropping the exported file directly into the folder where it should be installed.

Add a snippet to a workflow Snippets can be added to a workflow in two different ways; It can either be added by dragging and dropping the snippet from the **Side Panel** into the workflow editor, or it can be added by using the "Add element" option that is shown in figure 14.25.

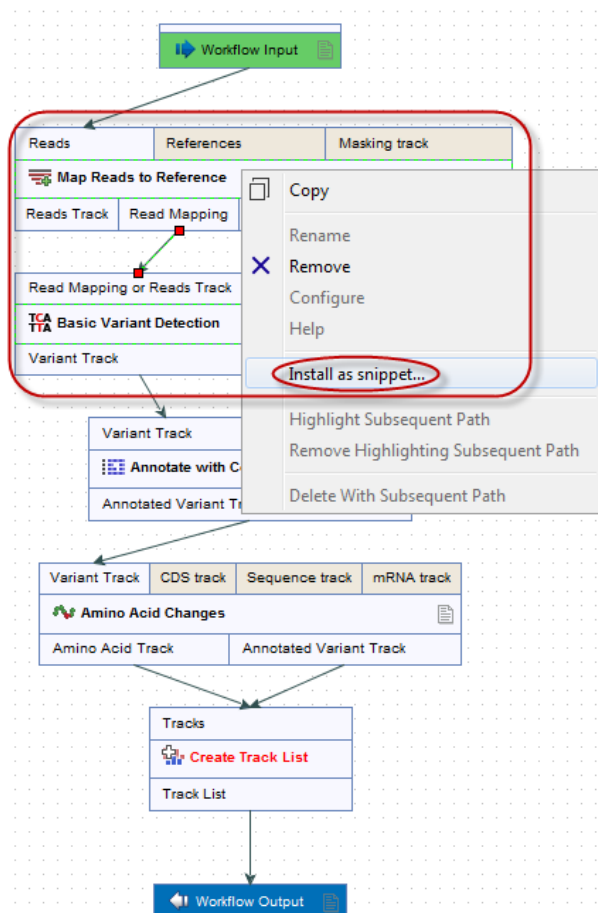


Figure 14.20: The selected elements are highlighted with a red box in this figure. Select "Install as snippet".

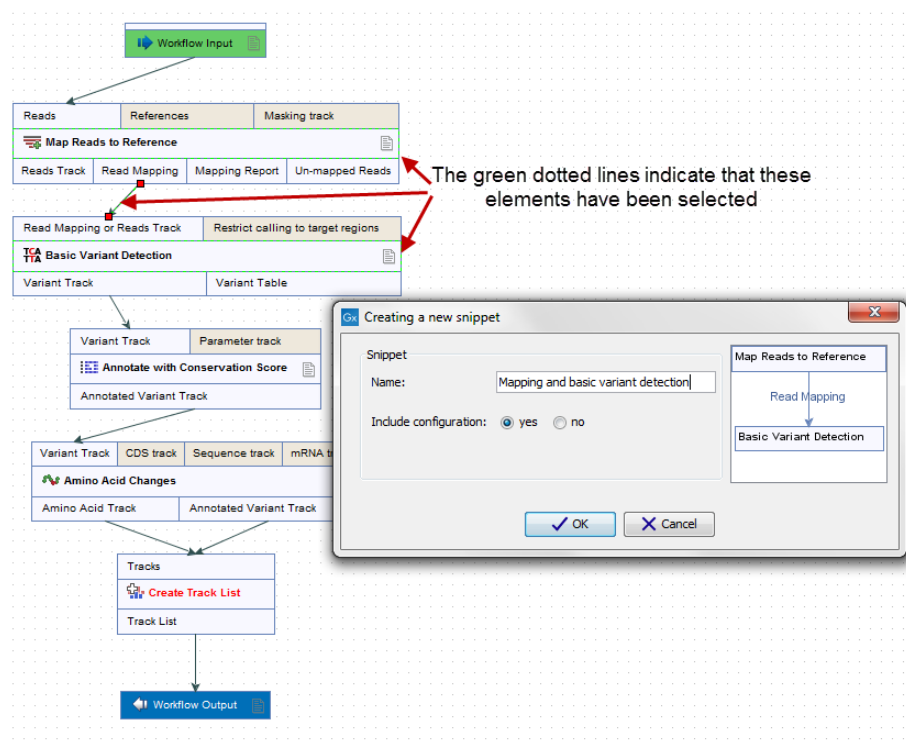


Figure 14.21: In the "Create a new snippet" dialog you can name the snippet and select whether or not you would like to include the configuration. In the right-hand side of the dialog you can see the elements that are included in the snippet.

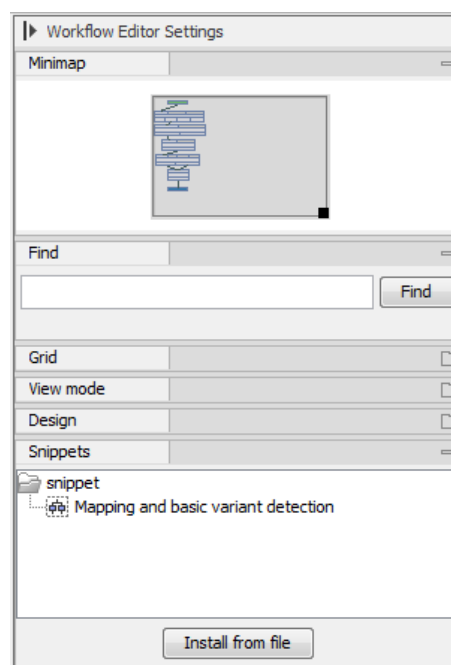


Figure 14.22: When a snippet is installed, it appears in the Side Panel under the "Snippets" tab.

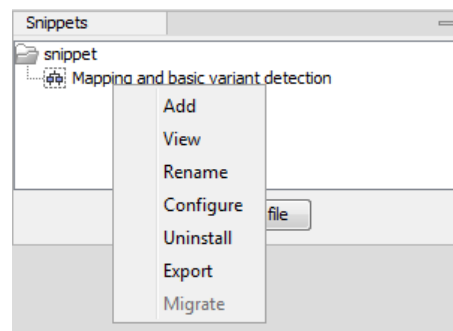


Figure 14.23: Right-clicking on an installed snippet brings up a range of different options.

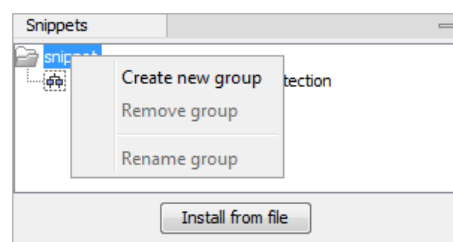


Figure 14.24: Right-clicking on the snippet top-level folder makes it possible to manipulate the groups.

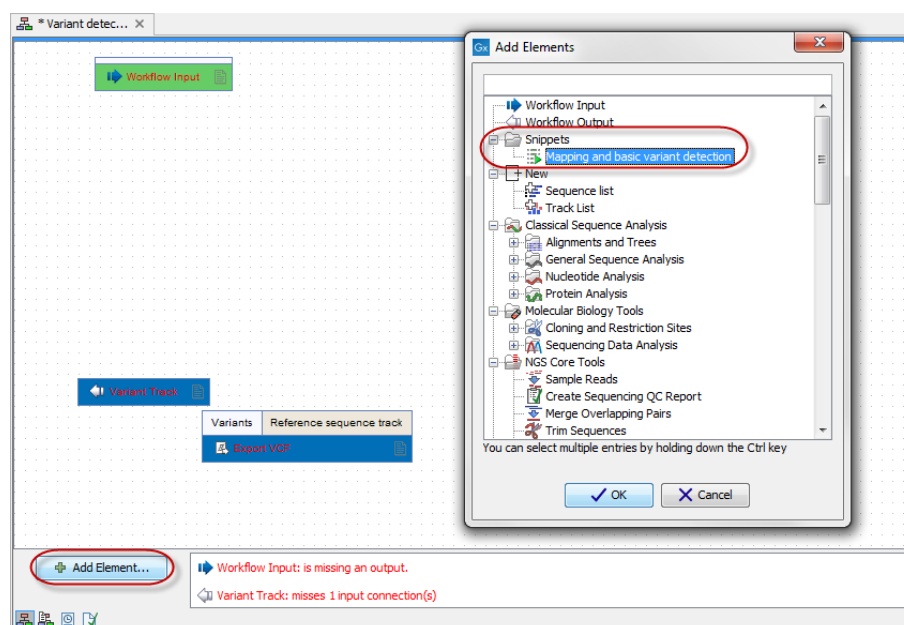


Figure 14.25: Snippets can be added to a workflow in the workflow editor using the 'Add Element' button found in the lower left corner.

14.1.9 Customizing the Workflow Editor

The Workflow Editor side panel (figure 14.26) contains settings that can make navigating through and editing workflows easier. They can be particularly useful when working with large workflows.

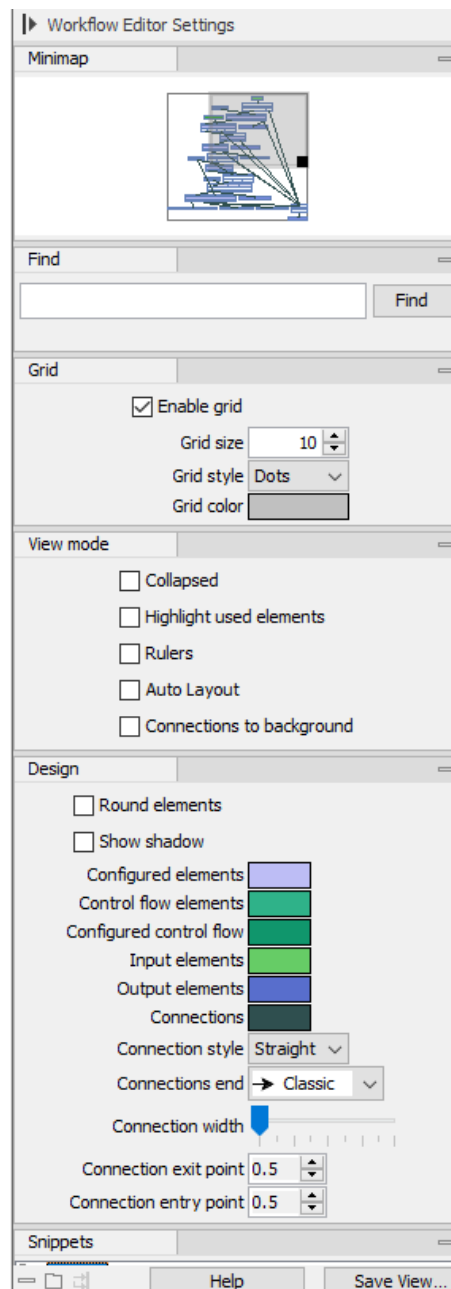


Figure 14.26: The Workflow Editor side panel

The Workflow Editor side panel contains the following:

Minimap A zoomed-out overview of the workflow. The darker grey box in the minimap highlights the area of the workflow visible in the editor. Drag that box within the minimap to quickly navigate to a specific area in the editor. The location of this dark grey box is updated when you navigate to another area of the workflow.

Find Search for workflow elements based on names.

After you enter a term and click on the Find button, the number of elements found with the term in their name is reported below the search field. The elements identified are highlighted in the workflow editor (figure 14.27). When multiple elements were found, the view will update so the first element found is visible in the editor. Click on the Find button again to bring the second element into view. Click on Find again to bring the third element into view, and so on.

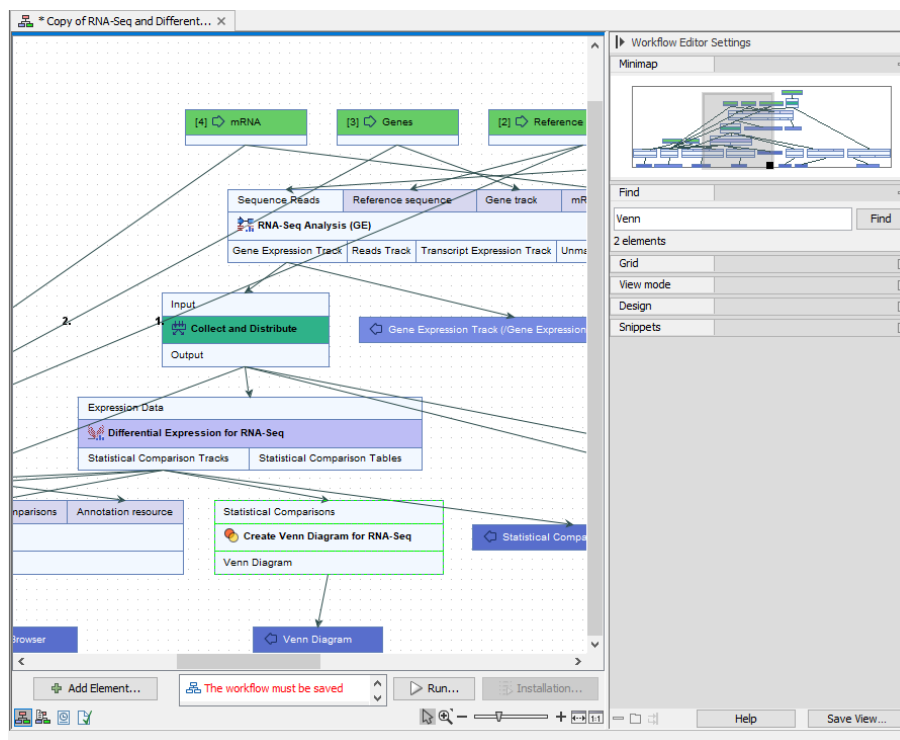


Figure 14.27: Two elements with names including the term "venn" were found using the Find tool in the side panel. Both are visible in this view, with the first element found highlighted.

Grid Customize the spacing, style and color of the symbols used in the grid on the canvas, or choose not to display a grid. Workflow elements snap to the grid when they are added or moved around.

View mode Settings under the View tab are particularly useful when working with large workflows, as they can be used to remove aspects of the design that are not of immediate interest.

- **Collapsed** Enable this to hide the input and output channels of workflow elements (figure 14.28).

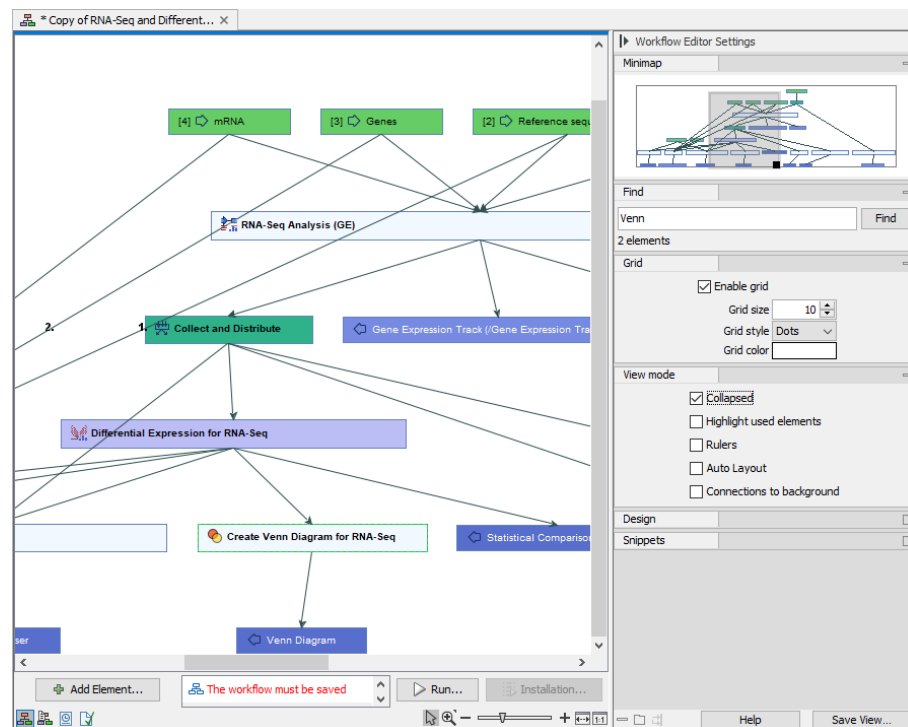


Figure 14.28: The same workflow as above but with the "Collapse" option in the View mode settings enabled.

- **Highlight used elements.** Enabling this option results in elements without at least one input and one output connection to appear faded. Elements connected to those missing connections are also faded (figure 14.29). (Shortcut: Alt + Shift + U)

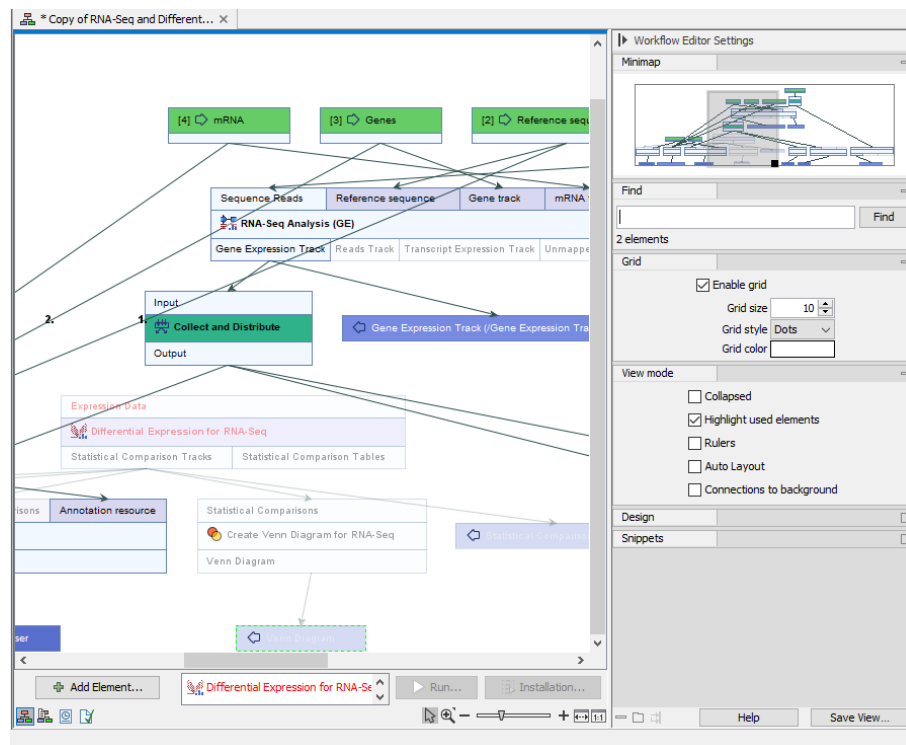


Figure 14.29: A similar workflow to those above but with the "Highlight used elements" option in the View mode settings enabled. The faded coloring makes it easy to spot that the workflow arm starting with *Differential Expression for RNA-Seq* is not connected to the rest of the workflow.

- **Rulers** Adds rules along the left vertical and top horizontal edges of the canvas.
- **Auto Layout** Enable this option to adjust the layout automatically every time an element is added and connected. Depending on the workflow design, using the "Layout" option in the right-click menu over the canvas can be preferable (see section 14.1.6).
- **Connections to background** Enable this to put connection lines behind workflow elements (figure 14.30).

See also the Design options, described below, where you can change the color and design of connections.

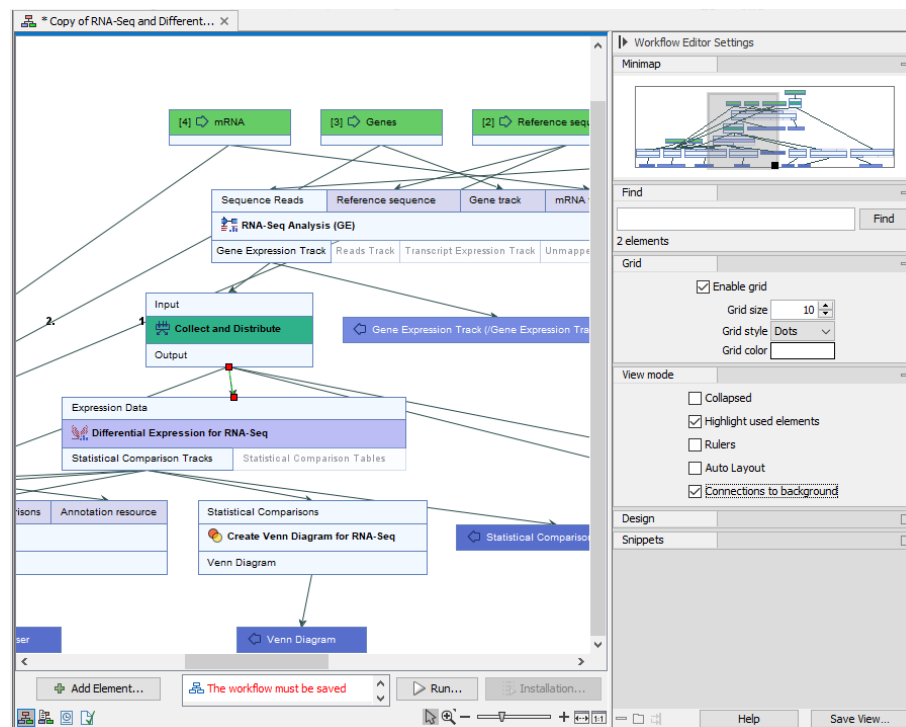


Figure 14.30: A similar workflow to those above but with the "Connections to background" option in the View mode settings enabled.

Design Options under the Design tab allow the shapes and colors of element and connections to be defined. Of particular note is the ability to color elements with non-default configurations differently to those with default settings.

- **Round elements** Round the corners of elements.
- **Show shadow** Add shadows under elements.
- **Coloring** Customize element background colors. Default colors assigned to different types of elements can be specified, and elements with non-default configurations can be assigned a different color to elements with default configurations (figure 14.31). See section 14.2.1) for information on element categories.
- **Connection related options** Customize connection lines between elements, including whether they should be straight or elbow lines, their width and where they should enter and exit elements.

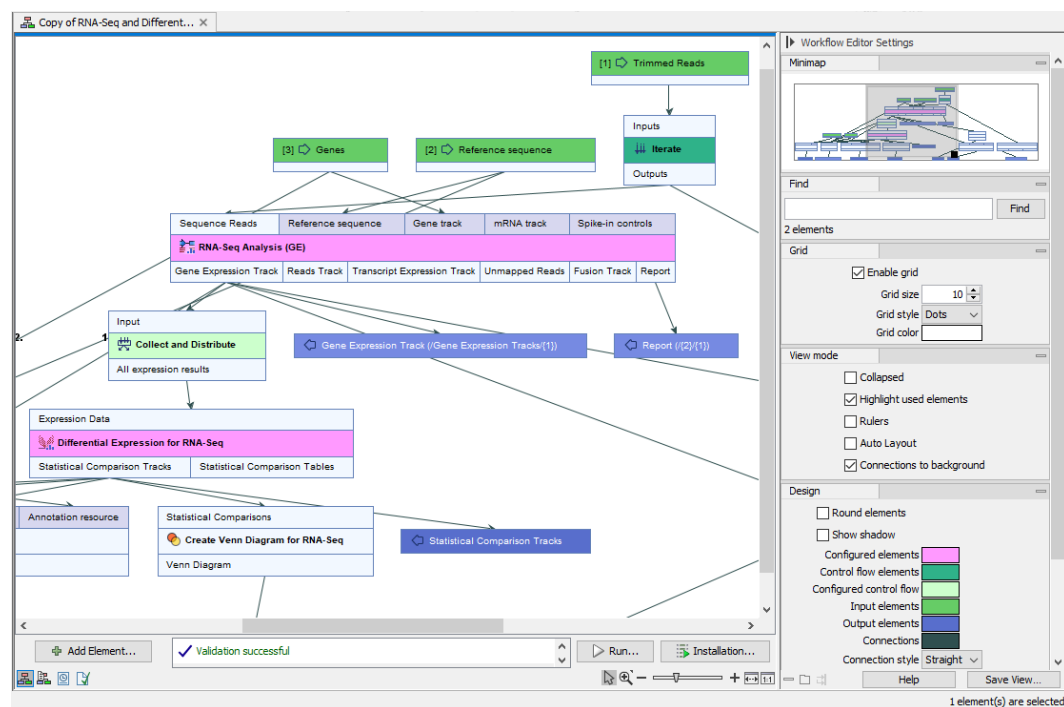


Figure 14.31: A similar workflow to those above, but where standard elements with non-default configuration have been assigned the color pink and control flow elements with non-default configurations have been assigned a pale green color, making them easy to spot.

Snippets Snippets are sections of workflows, which have been saved and can be easily added to a new workflow. These are described in section 14.1.8.

14.2 Workflow elements

Workflow elements are the building blocks of workflows. Detailed customization of how the workflow will behave can be done by adding and connecting relevant elements, and also by setting individual parameter values, and choosing which parameter values can be edited when launching the workflow. Customization of how the workflow will look when used is also possible by editing workflow element names and the names of their parameter values. These names are used in the wizard generated when the workflow is launched.

In this section, we describe general aspects of workflow elements and then focus on particular element types, the role they play in workflows, and their configuration.

14.2.1 Anatomy of workflow elements

Workflow elements can pass data to other elements, accept data from other elements, or do both of these. Elements that data can enter into and flow out of consist of 3 regions: input channels at the top, output channels at the bottom, and the core section in the middle where the element name is (figure 14.32). A page symbol on the right hand side of the middle section of a workflow element indicates the element can be configured.

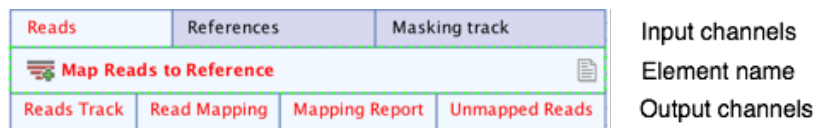


Figure 14.32: Anatomy of a workflow element.

Workflow element coloring

Workflow elements are colored according to their category (figure 14.33):

Light green Input elements. Elements for taking in the data to be analyzed.

Dark blue Output and Export elements. Elements that indicate data should be saved to disk, either in a CLC location (Output elements) or any other accessible file location (Export elements).

Light grey An analysis element where the default values are used for all settings.

Purple A configured analysis element, i.e. one or more values in that element have been changed from the defaults.

Emerald green Control flow elements with default settings.

Forest green Configured control flow elements. i.e. one or more values in that element have been changed from the defaults.

Background colors can be changed under the Design tab in the side panel settings of the Workflow editor.

The name of a new element added to a workflow is shown in red text until it is properly connected to other elements.

Configuring Input and Output elements is described in section 14.2.3 and section 14.2.4.

Control flow elements, used to fine tune control of the execution of whole workflows or sections of workflows, are described in section 14.2.5.

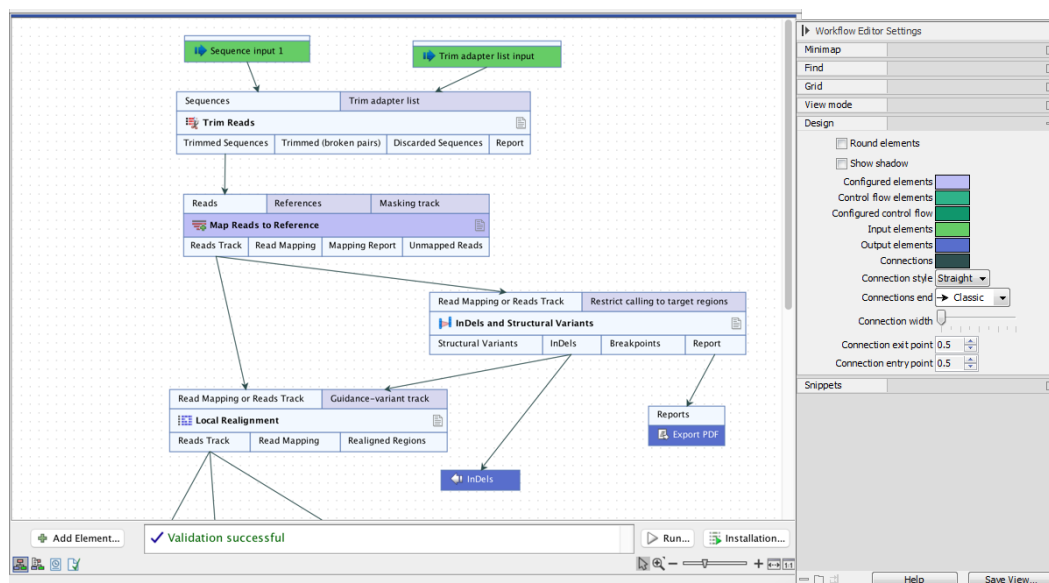


Figure 14.33: An element's color indicates the role it plays and its state. Here, *Trim Reads* means is using only default parameter values, whereas the purple background for *Map Reads to Reference* indicates that one or more of its parameter values have been changed. The green elements are Input elements. The blue color of the *InDels* element and parts of the *Export PDF* element indicate that data sent to these elements will be saved to disk.

14.2.2 Basic configuration of workflow elements

Workflow elements can be renamed, and options within the elements can be customized, affecting what is seen when editing the workflow, what is seen when launching the workflow, and the settings used by default when the workflow is run.

Renaming workflow elements

Renaming workflow elements can aid understanding of a workflow design, especially when there are multiple elements for the same tool. Updated element names are reflected in the workflow launch wizard.

To rename an element, right click on the element name and choose the option "Rename...", or select the element in the Workflow Editor and press the F2 key.

The **Find** functionality in the side panel finds elements based the new name.

Both the new name and the original, default name of elements are provided in the Workflow Configuration view (figure 14.34). Terms in the new name or the original, default name can be used in the **Filter Elements** functionality of this view to find settings for configurable elements. This view is described in section 14.1.7.

Note: The Workflow Configuration view can be opened linked to the Workflow view. Linked views are described further in section 2.1.

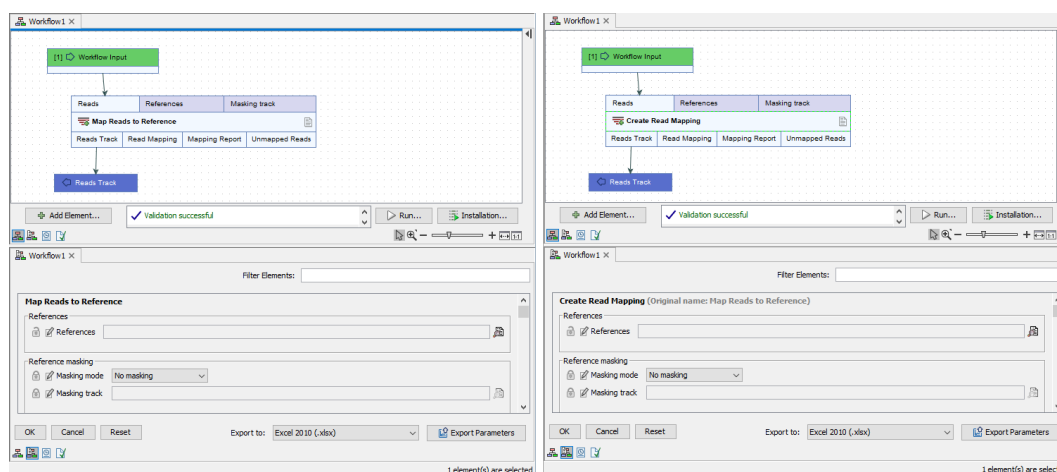


Figure 14.34: A workflow before (left) and after (right) the *Map Reads to Reference* element was renamed. In the linked *Workflow Configuration* view at the bottom right, both the original and updated element names are listed.

Configuring options in workflow elements

Configuring options can be done in a workflow element's configuration dialog by:

- Double-clicking on an element name in the Workflow view, or
- Right-clicking on an element name and choosing the *Configure...* option from the menu that appears.

Options can also be edited in the Workflow Configuration view (figure 14.34).

Workflow element customization can include:

1. Specifying which values can be changed when launching the workflow

Beside each option is a lock symbol. Values of unlocked (🔓) options can be changed when launching the workflow. Values of locked (🔒) options cannot be changed when launching the workflow.

Click on a lock icon to change it from locked (🔒) to unlocked (🔓), or vice versa.

The values of all locked settings can be seen in the workflow launch wizard, but they are not presented by default (figure 14.36).

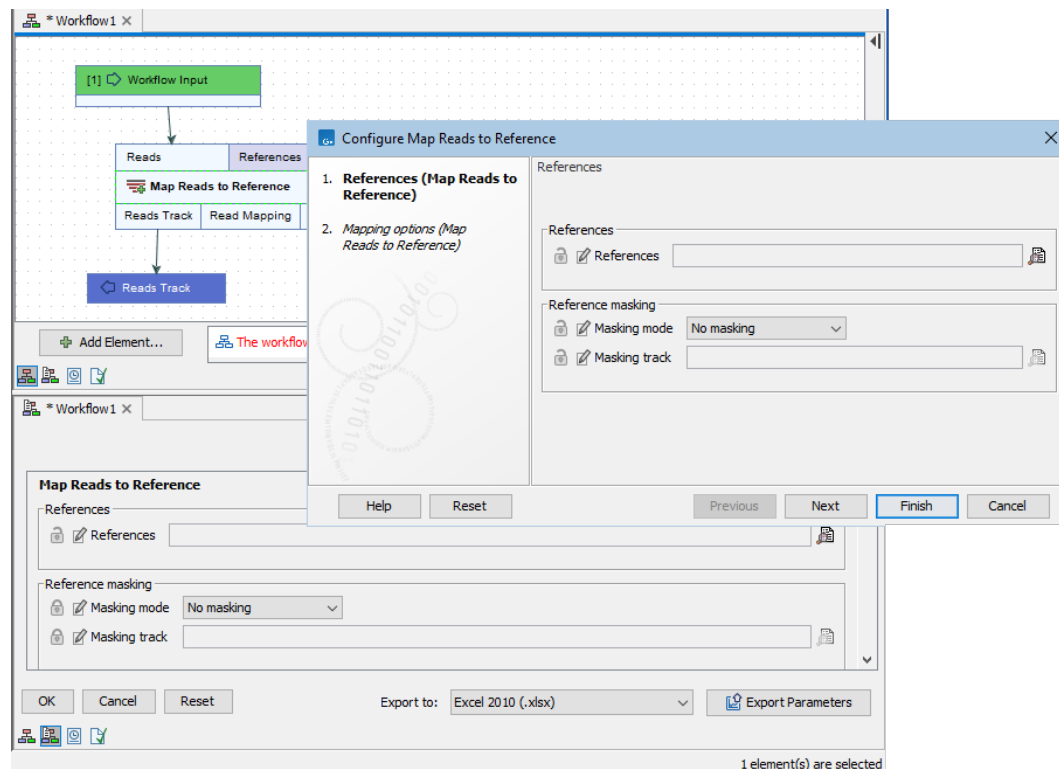


Figure 14.35: The Workflow view (top) and Workflow Configuration view (bottom) have been opened as linked views. The Map Reads to Reference element has been opened for configuration in the Workflow view and the Masking mode and Masking track options have been unlocked. They will correspondingly appear unlocked in the Workflow Configuration view after the Finish button is clicked.

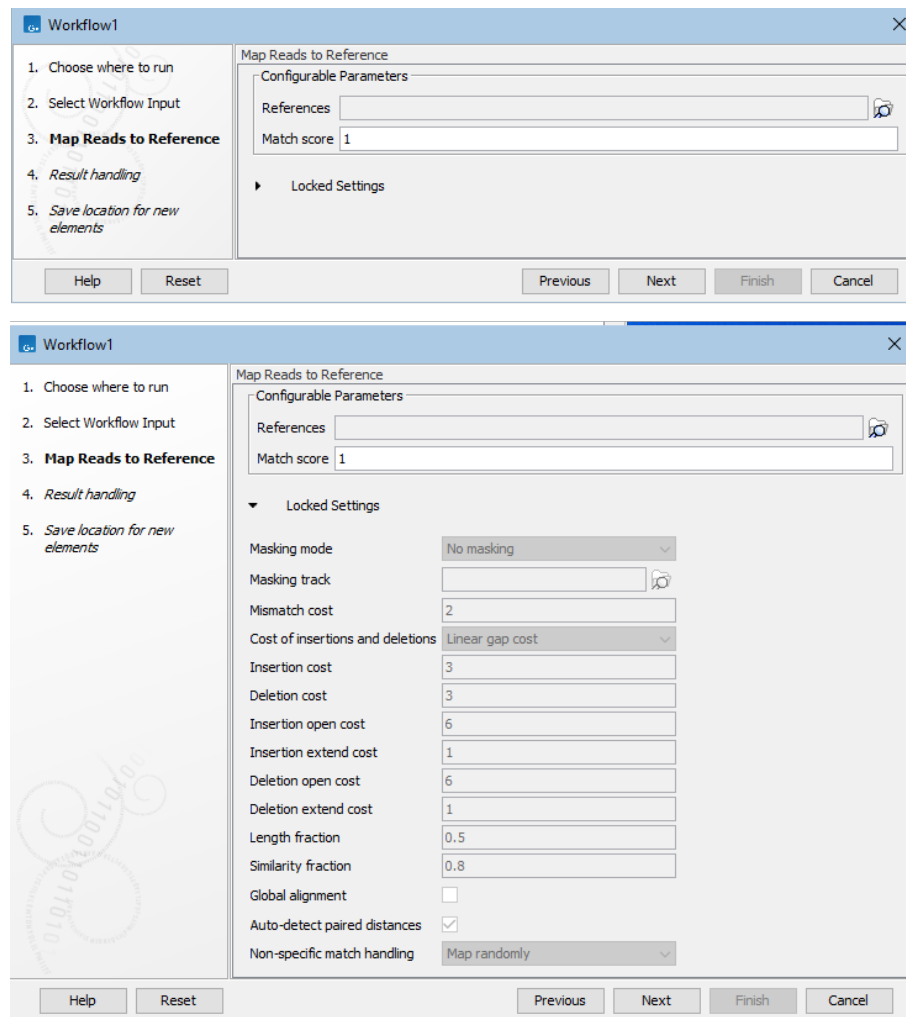


Figure 14.36: A workflow launch wizard step showing the configurable (unlocked) options at the top, with a heading for the locked settings (top). Clicking on the Locked Settings heading reveals a list of the locked options and their values (bottom)

2. Changing default values

When elements are added to a workflow, the default values assigned to options are the same as the defaults for the corresponding tool when run from the Tools menu. Values for options can be updated in the workflow design. When this is done, the new values become the defaults used when the workflow is run.

3. Changing option names

When an option name is changed, the new name is used in the workflow launch wizard.

Renaming options can be useful when only a few options can be configured when launching the workflow and their default names are quite generic.

Click on the edit icon (✎) beside an option to edit the name (figure 14.37).

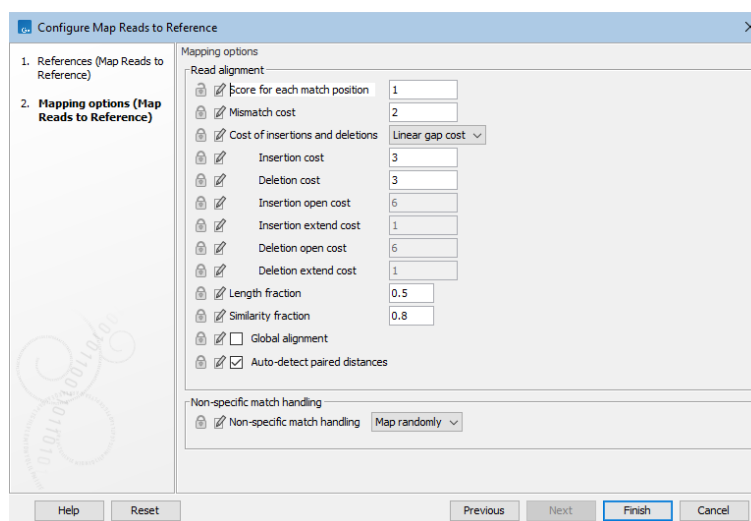


Figure 14.37: An option originally called "Match score" has been renamed "Score for each match position" in the element configuration dialog. It has also been unlocked so the value for this option will be configurable when launching the workflow.

Note: Clicking on the **Reset** button in a workflow element configuration dialog will reset all changes in that particular configuration step to the defaults, including any updated option names.

14.2.3 Configuring Workflow Input elements

Workflow Input elements are used to provide data to a workflow. At least one such element is required in a workflow for supplying the data for analysis. Input elements can be connected to any input channel expecting data.

When there is more than one Input element in a workflow, the order that data is prompted for in the launch wizard can be configured (see Ordering inputs (section 14.1.3)).

For information about configuring reference data inputs in workflows, see section 14.2.3. The use of the "Workflow role" option, available in tool input channels and in Input elements where reference data is expected is described there. Information about launching workflows where workflow roles have been defined is described in section 14.3.4.

Configuring Input element options

By default, Input elements are configured to allow data to be selected from CLC locations listed in the Navigation Area, or from elsewhere, in which case, the data will be imported using on-the-fly import. The workflow author can limit these options (figure 14.38), as well as configure import options with non-default values. Settings can be locked if they should not be configurable when launching the workflow (see Basic configuration of workflow elements (section 14.2.2)).

On-the-fly import options that can be configured in Input elements are:

- **Allow any compatible importer** All compatible importers will be available when launching the workflow and all the options for each importer will be configurable.
- **Allow selected importers** Specify particular importers to be available when launching the workflow. Options for each selected importer can be configured by clicking on the **Configure Parameters** button.

Note: To specify CLC data not stored in a CLC location as input to a workflow, on-the-fly import must be allowed, and CLC Format must be one of the allowed importers. See also Launching workflows individually and in batches (section 14.3).

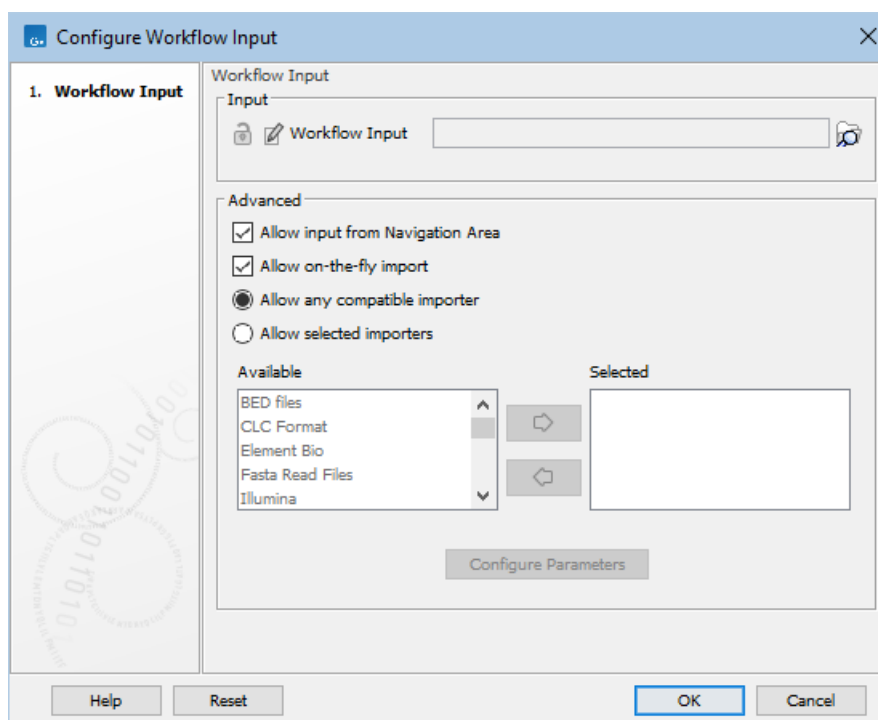


Figure 14.38: The allowed data sources are configured in Input elements. By default, both checkboxes in the Advanced section are enabled, allowing data to be selected from a CLC location (input from the Navigation Area), or from another location (on-the-fly import). For on-the-fly import, any available importer can be used by default. When only specific importers are allowed, those importers can be configured by selecting each in turn and clicking on the "Configure Parameters" button. In this case, only the data types supported by the specified importers, can be selected as input for on-the-fly import when launching the workflow.

On-the-fly import versus using workflow elements for specific importers

For importing data as the first step of a workflow, on-the-fly importer, as described above, is the most flexible and commonly used option. However, workflow input elements for specific NGS importers are also available.

Examples using each of these options are shown in figure 14.39. How these translate when launching the workflow is shown in figures 14.40 and 14.41. The relative merits of each option are outlined in table 14.1.

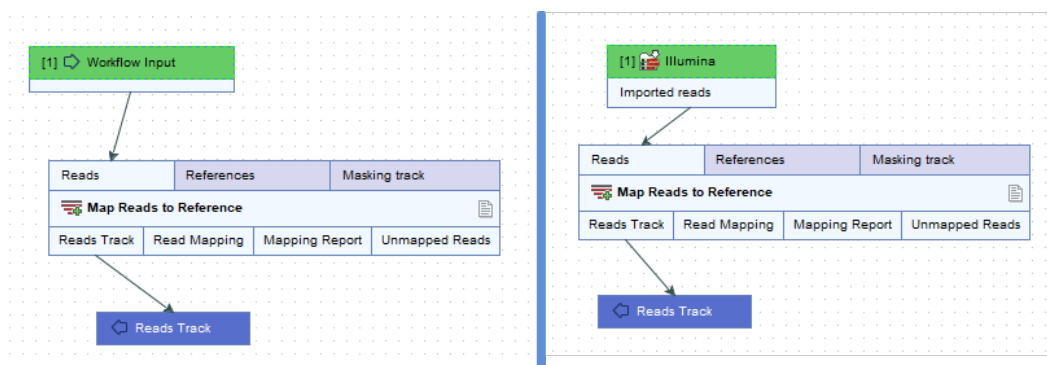


Figure 14.39: Raw data can be imported as part of a workflow run in 2 ways. Left: Include an Input element, and use on-the-fly import. Right: Use a specific Import element. Here, the Illumina import element was included.

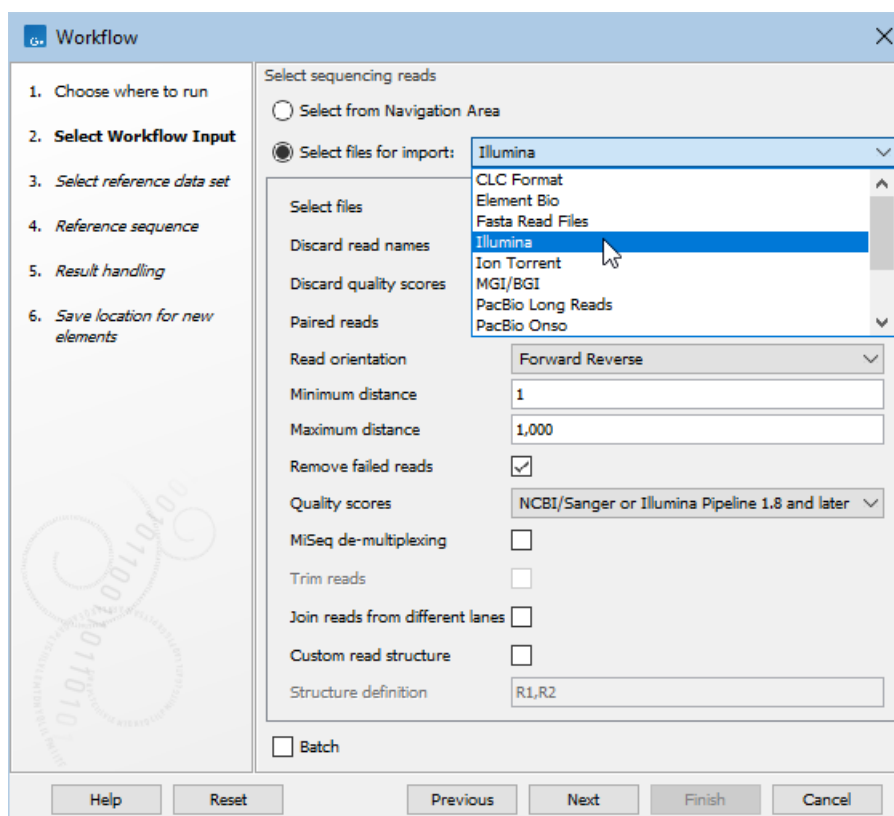


Figure 14.40: Launching a workflow with an Input element and choosing to select files to import on-the-fly.

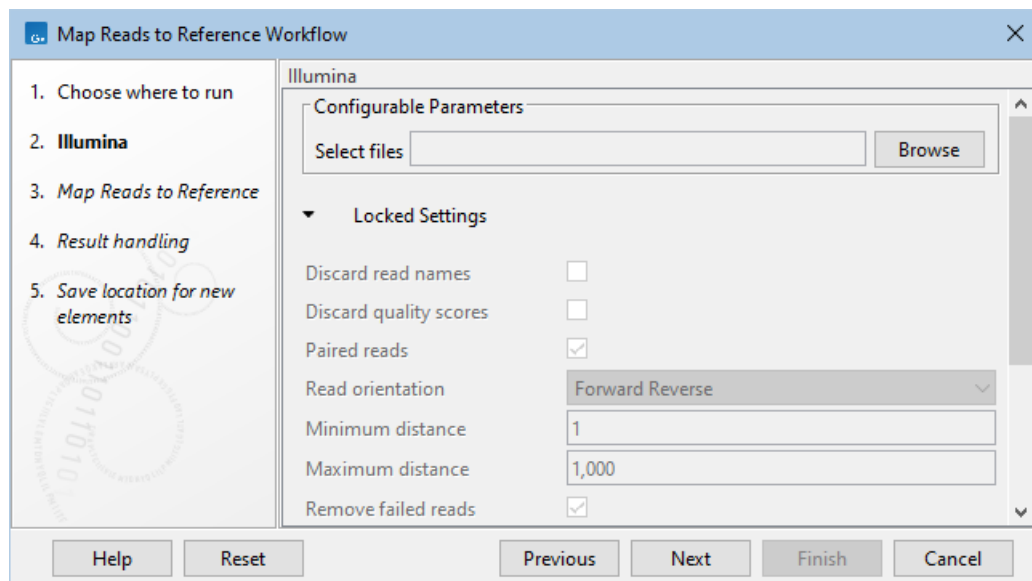


Figure 14.41: Launching a workflow with a dedicated import element, in this case, an Illumina import element.

Functionality	Input element	Dedicated import element
<i>Running in batch mode</i>	Supported. Check the Batch option in the launch wizard.	Not supported. (The Batch option is not visible in the launch wizard).
<i>Iterate elements</i>	Supported.	Supported.
<i>Choosing an importer when launching</i>	Any available importer can be selected when launching. Use of already-imported data is also supported. Workflow authors can specify the importers available when launching.	Only data formats relevant for the specific importer can be selected for use.
<i>Configuring import options</i>	Options for all importers allowed by the workflow author can be configured, and set to be unlocked or locked.	Import options for the specific importer can be configured, and set to be unlocked or locked.
<i>Saving imported elements</i>	Supported. To save the imported elements, connect a Save On-the-fly Imports element to the Input element, and connect its output to an Output element.	Supported. To save the imported elements, connect an Output element to the Import element.

Table 14.1: Workflow import methods compared

Notes:

- The use of Iterate elements to run all or part of a workflow in batches is described

in section 14.3.3.

- Configuration options for NGS importers are described in section 7.3.

Saving data imported on-the-fly

To save data elements created using on-the-fly import, connect a **Save On-the-Fly Imports** (📁) element to the relevant Input element and connect an Output element to it (figure 14.42).

The **Save On-the-Fly Imports** element only acts on data imported using on-the-fly import. If input data is selected from a CLC File Location, this element has no effect.

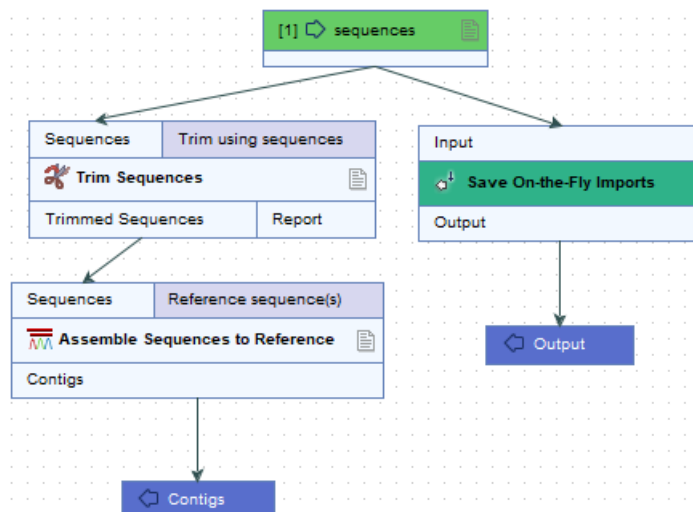


Figure 14.42: Using this workflow, data imported on-the-fly would be saved as an output from the Save On-the-Fly Imports element.

A **Save On-the-Fly Imports** element is not needed when an Iterate element is connected to the Input element being used for on-the-fly import. In this situation, an Output element can be connected directly to the Iterate element (figure 14.43).

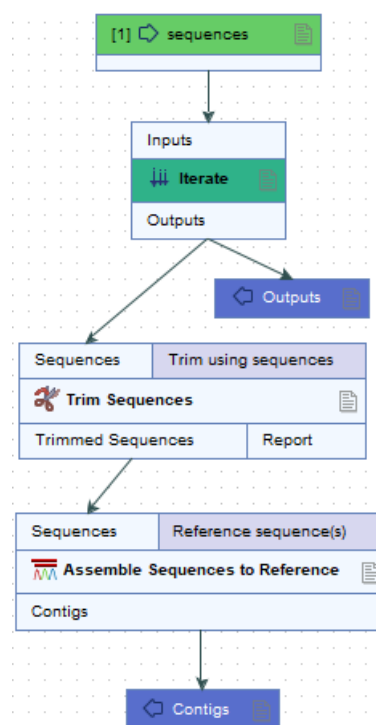


Figure 14.43: When an Iterate element is connected to an Input element, data imported on-the-fly can be saved by connecting an Output element to the Iterate element.

Reference data and workflows

Many analyses require reference data. For example, when mapping reads, the reference genome to map the reads to must be specified. The workflow design and configuration determines what choices, if any, will be presented for selecting reference data when the workflow is launched.

The choices available to the workflow designer are:

1. Force the use of specific reference data. No choice is offered when launching the workflow.

To configure a workflow element for this, select the data element to use as input in the relevant workflow element, and then lock that option. Do not enter any term into the "Workflow role" field (figure 14.44).

2. Allow the choice of any compatible data element when launching the workflow.

To configure a workflow element for this, unlock the relevant input option. Each unlocked option will be presented in the workflow launch wizard. If a data element is pre-selected, that element will be pre-selected in the launch wizard, but it can be replaced by a different element (figure 14.45).

If a term is entered in the "Workflow role" field, an additional launch wizard step will be presented for specifying the reference data handling for that workflow run. In that "Specify reference data handling" step, selecting the option to use specified data elements results in later wizards steps being presented that allow the selection of the relevant data elements (figure 14.46).

Note: If neither a *workflow role* nor a data element is configured, the workflow input must remain unlocked so that a data element can be selected when launching the workflow.

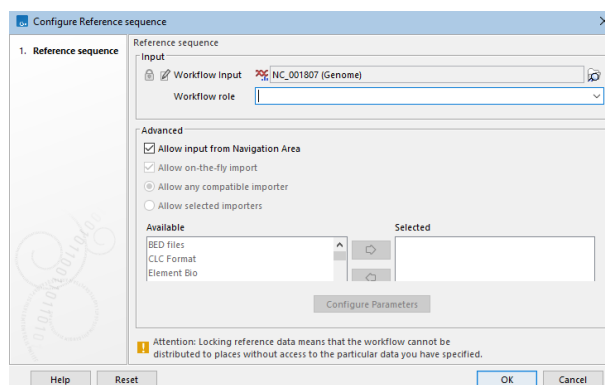


Figure 14.44: A data element has been selected in the Workflow Input field and the option is locked. When the workflow is launched, there will be no launch wizard step allowing selection of a different element for this input.

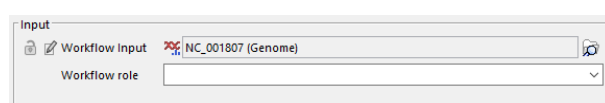


Figure 14.45: A workflow input expecting reference data has been configured with a specific data element and left unlocked. This data element will be pre-selected in the corresponding wizard step when the workflow is launched, and a different data element could be selected for use at that time.

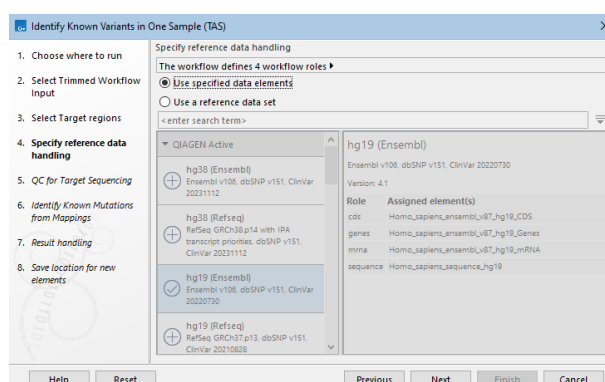


Figure 14.46: The "Specify reference data handling" wizard step is presented because at least one workflow input has been configured with a workflow role. The option to use specified data elements has been selected, so subsequent wizard steps will be presented for the selection of those elements.

3. Allow the selection of a Reference Data Set when launching the workflow.

If the "Workflow role" field for at least one workflow input has been configured, a wizard step allowing the selection of a Reference Data Set will be shown when launching a workflow (figure 14.47).

If all workflow inputs configured with a *workflow role* are unlocked, then when the workflow is launched, a Reference Data Set or individual reference data elements can be selected.

If any workflow input with just a *workflow role* configured is locked, a Reference Data Set must be selected when launching the workflow.

For further information about selecting a Reference Data Set in a workflow launch wizard,

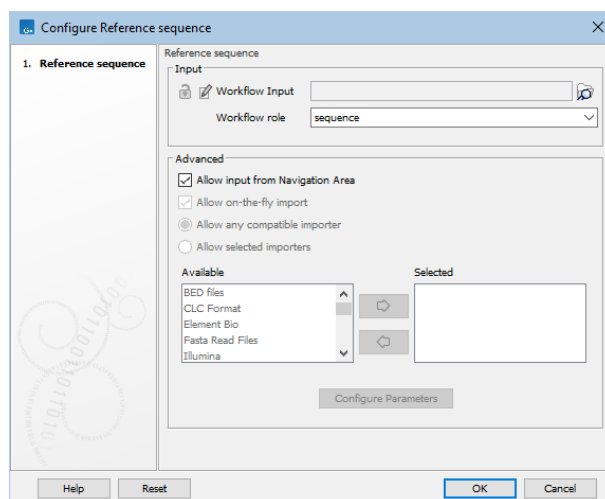


Figure 14.47: The workflow role, "sequence", has been specified for this input. When this workflow is launched, a workflow wizard step will be presented, allowing the selection of a Reference Data Set as the source of the reference data.

see section 14.3.4. For further information about Reference Data Sets, see section 11.3.

14.2.4 Configuring Workflow Output and Export elements

Results generated by a workflow are only saved if the relevant output channel of a workflow element is connected to a Workflow Output element or an Export element. *Data sent to output channels without an Output or Export element attached are not saved.*

Terminal workflow elements with output channels must have at least one Workflow Output element or Export element connected.

Configuring custom names for workflow results

The names to assign to outputs and exported files from workflows can be configured to include specific text as well as information taken from a workflow run, for example, the names of inputs to the analysis, dates and times the results were generated, etc.

To configure the naming pattern for a Output or Export workflow element, double-click on it, or right-click on it and then select the option **Configure...** from the menu. The naming pattern in Output elements is defined in the **Custom output name** field (figure 14.48). In Export elements, it is defined in the **Custom file name** field.

Hover the mouse cursor over the configuration field to reveal a tooltip containing a list of available *placeholders* (figure 14.49). Placeholders are terms within curly brackets used to indicate that particular information from a workflow run should be included in the output name or exported file name. Terms in placeholders are not case specific.

Note: Placeholders used by export tools run directly (not via a workflow) are described in section 8.1.3. Other settings relating to export, relevant both for exports run directly or in a workflow context, are described in section 8.1.2.

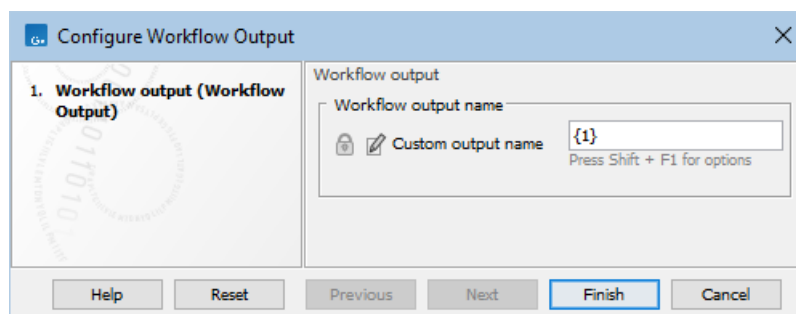


Figure 14.48: Defining the name to assign to an output from a workflow. The default naming pattern for Output elements uses the placeholder `{1}`, which is a synonym for the placeholder `{name}`.

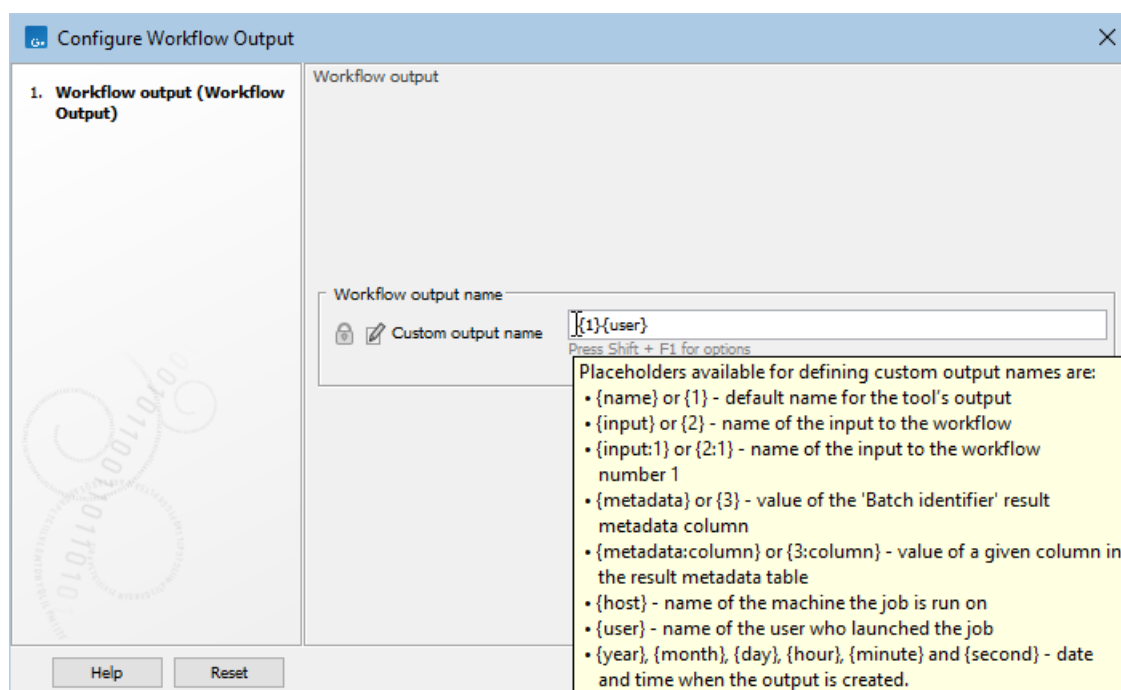


Figure 14.49: Hover the mouse cursor over the field where a custom name can be configured to reveal a tooltip with a list of available placeholders.

Placeholders available for Output and Export workflow elements are:

- **{name}** or **{1}** The default name for that output from that tool, i.e. the name that would be used if the tool was run outside a workflow context.
- **{input}** or **{2}** The name of the primary workflow input(s) for the path of the workflow being traversed.

"Primary workflow input" generally refers to the data being analyzed, i.e. inputs expecting sample data, as opposed to inputs expecting reference data.

For a workflow with multiple primary inputs to an arm of the workflow, **{input}**, or its equivalent **{2}**, would result in the name of each of these primary inputs being included in the names of the outputs from that workflow arm (figure 14.50).

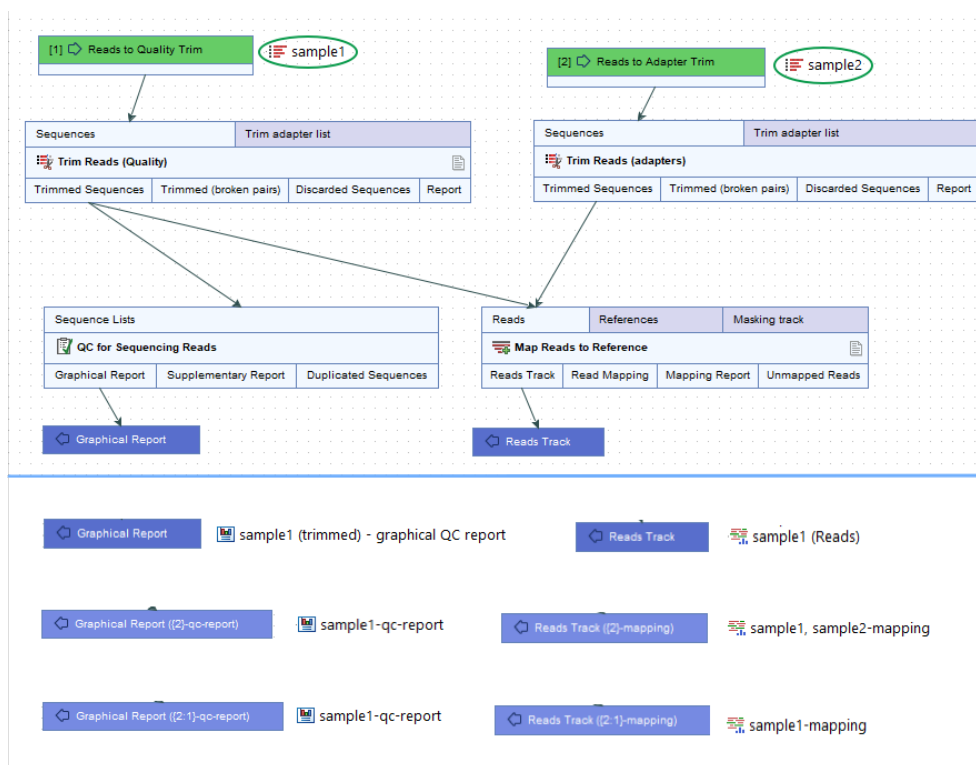


Figure 14.50: Top: A contrived workflow with two primary inputs (green boxes). The QC for Sequencing Reads step receives data only from the first input, "Reads to Quality Trim". The Map Reads to Reference step receives data originating from both primary inputs. Bottom: The effect of different naming patterns on result names when a sequence list called "sample1" was supplied for the first input and a sequence list called "sample2" was supplied for second input. The first row shows the Output elements and results using the default naming pattern, {1}. The middle row shows the Output elements and results when the naming pattern included the placeholder {2}, and the last row shows them when the naming pattern included the placeholder {2:1}.

- **{input:N}** or **{2:N}** The name of the Nth input to the workflow. E.g. **{2:1}** specifies the first input to the workflow, while **{2:2}** specifies the second input (figure 14.50).

Unlike the general form described above, i.e. **{input}** or **{2}**, reference data inputs can be included in names using this placeholder form (figure 14.51).

For a workflow with only one primary input, **{input}** or **{2}** is equivalent to the more specific form **{input:1}** or **{2:1}**.

For workflows containing control flow elements, the specific placeholder form, **{2:N}**, is recommended.

See section 14.1.3 for information about workflow input ordering, and section 14.2.5 for information about control flow elements.

- **{metadata}** or **{3}** The batch unit identifier for workflows executed in batch mode. Depending on how the workflow was configured at launch, this value may be obtained from metadata. For workflows not executed in batch mode or without Iterate elements, the value will be identical to that substituted using **{input}** or **{2}**.

Note: For workflows containing control flow elements, the more specific form of place-

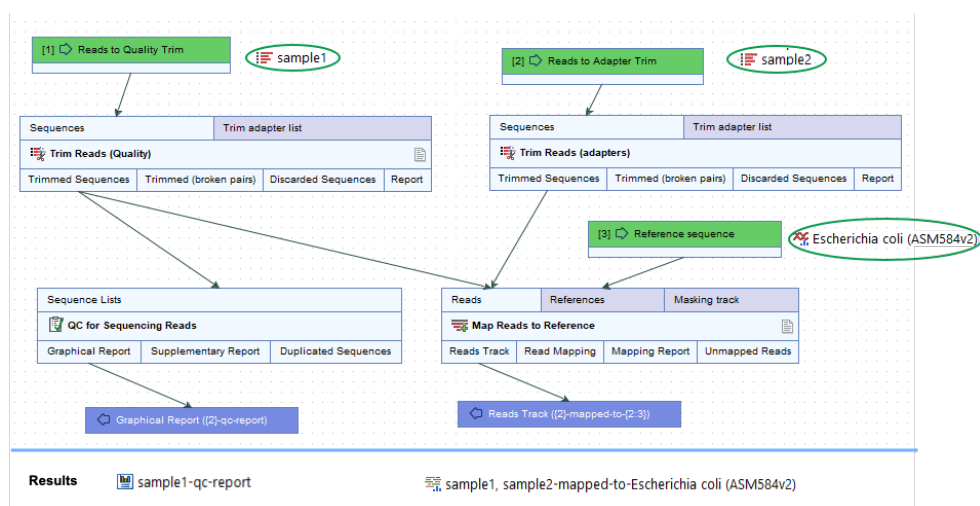


Figure 14.51: Top: A contrived workflow with two primary inputs and a reference data input (green boxes). Bottom: The names of the results generated in a given workflow run. The naming pattern for the Reads Track output includes {2}, which adds the names of all primary inputs to that analysis step, (sample1, sample2), and {2:3}, which adds the name of the third input, whatever the role that input has. In this case, it is a reference data input and an element called *Escherichia coli* (ASM584v2) was supplied.

holder, i.e. the `metadata:columnname` or `{3:columnname}` form, described below, is recommended.

- **{metadata:columnname}** or **{3:columnname}** The value for the batch unit in the column named "columnname" of the metadata selected when launching the workflow. Pertinent for workflows executed in batch mode or workflows that contain Iterate elements. If a column of this name is not found, or a metadata table was not provided when launching the workflow, then the value will be identical to that substituted using **{input}** or **{2}**.
- **{user}** The username of the person who launched the job
- **{host}** The name of the machine the job is run on
- **{year}**, **{month}**, **{day}**, **{hour}**, **{minute}**, and **{second}** Timestamp information based on the time an output is created. Using these placeholders, items generated by a workflow at different times can have different file names.

In addition to the placeholders above, the placeholder **{extension}** is available for exported file names. This is replaced by the default file extension for the exported file's format, e.g. .pdf, .txt.

Saving results to subfolders

Workflow outputs and exported files can be saved into subfolders by adding a forward slash / at the start of the custom name definition.

For example, with an Output element configured with `/variants/{name}`, the resulting output would be saved to a subfolder called `variants`, placed within the folder selected for outputs when the workflow is launched. If a specified subfolder does not already exist, it is created when the outputs are saved.

When defining subfolders for outputs or exported files, terms between all forward slash characters are interpreted as subfolders. For example, a name like `/variants/level2/level3/myoutput` would put the data item called `myoutput` into a folder called `level3` within a folder called `level2`, which itself is inside a folder called `variants`. The `variants` folder would be placed under the location selected for storing the workflow outputs.

Temporary, intermediate workflow results

During a workflow run, temporary, intermediate results may be generated, including for output channels that aren't connected to an Output or Export element.

Such intermediate results are normally deleted automatically after the workflow run completes. If a problem arises such that the workflow does not complete normally, intermediate results may not be deleted and will be in a folder named after the workflow with the word "intermediate" in its name.



14.2.5 Control flow elements

Control flow elements control the flow of data through a workflow. They can be found in the Control Flow folder of the Add Elements wizard (figure 14.52).

Elements are available for:

1. Controlling how data is grouped for analysis. These include **Iterate** and **Collect and Distribute**, described in section 14.2.5.
2. Controlling the flow of the workflow based on its configuration when launched. These include **Fork**, described in section 14.2.5, and **Save On-the-Fly Imports**, described in section 14.2.3.
3. Controlling the flow through the workflow based on aspects of the data. There are several such branching elements, described in section 14.2.5.

Iterate and Collect and Distribute elements

Iterate () and **Collect and Distribute** () elements are used to control how data is grouped for analysis.

- Iterate elements are placed at the top of a branch of a workflow that should be run multiple times, using different inputs in each run. The sets of data to use in each run are referred to as "batch units" or, sometimes, "iteration units".
- Collect and Distribute elements are, optionally, placed downstream of an Iterate element, where they collect outputs from the upstream iteration block (see below) and distribute them as inputs to downstream analyses.

The **RNA-Seq and Differential Gene Expression Analysis** template workflow, distributed with the *CLC Genomics Workbench* (https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=RNA_Seq_Differential_Gene_Expression_Analysis_workflow.html) is an example of a workflow that includes each of these control flow elements.

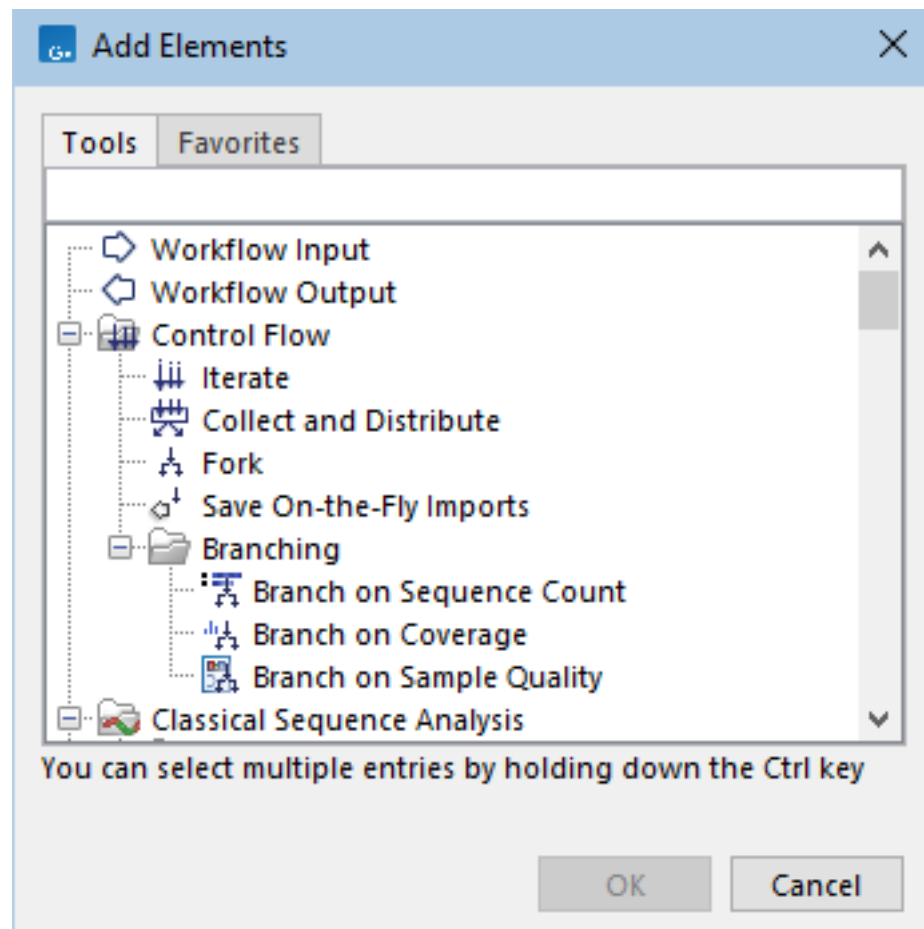


Figure 14.52: Control flow elements are found under the Control Flow folder in the Add Elements wizard.

The steps between an Iterate element and a Collect and Distribute element are referred to as an "iteration block". The workflow in figure 14.53 contains a single iteration block (shaded in turquoise), where steps within that block are run once per batch unit. The Collect and Distribute element (renamed to Collect Expressions) collects all the results from the iteration block and sends it as input to the next stage of the analysis (shaded in purple).



Figure 14.53: The roles of the Iterate and Collect and Distribute control flow elements are highlighted in the context of RNA-Seq and differential expression analyses. RNA-Seq Analysis lies downstream of an Iterate element, within an iteration block (shaded in turquoise). It will thus be run once per batch unit. Differential Expression for RNA-Seq lies immediately downstream of a Collect and Distribute element (renamed to Collect Expressions), and is sent all the expression results from the iteration block as input for a single analysis.

Defining batch units for workflows with Iterate elements

Workflow elements downstream of an Iterate element are run once for each batch unit. Details about defining batch units when launching workflows is described at section 14.3.2.

Running a workflow with a single Iterate element at the top of a workflow, no downstream Collect and Distribute element, and a single Input element is equivalent to running a similar workflow without the Iterate element in Batch mode. Setting up batch units in this situation is described in section 12.3.

Importing sequence data with sample information

A workflow containing just an Input, Output and Iterate element can be a useful tool to create a CLC Metadata Table with sample information and data elements associated with the relevant rows. This can then be used when launching tools and workflows requiring metadata. A template workflow with this design, Import with Metadata, is provided in the Preparing Raw Data template workflow folder, available under the Workflows menu, and is described in section 14.5.1.

Renaming Iterate elements

Providing meaningful names to Iterate elements can help at both the workflow design stage and also when launching the workflow.

The **Rename** option is available in the menu that appears when you right-click on a workflow element.

Iterate element names are included in the workflow launch wizard in the following steps:

- **Configure batching:** The name of Iterate elements are provided in association with the drop-down list of column names in the metadata provided. A meaningful Iterate element name can thus help guide the choice of relevant metadata to group the inputs into batch units (figure 14.54).
- **Batch overview:** There is a column for each Iterate element (figure 14.55). Meaningful names can thus make it easier to review batch unit organization critically when launching the workflow.

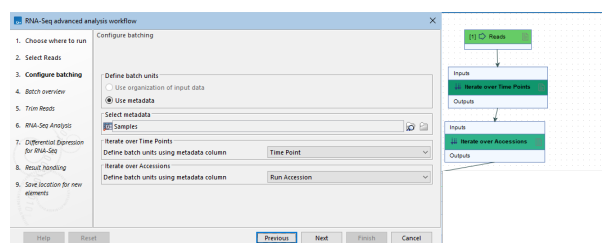


Figure 14.54: The two Iterate elements in this workflow (right) have been renamed. Their names are included in the "Configure batching" wizard step in the launch wizard (left).

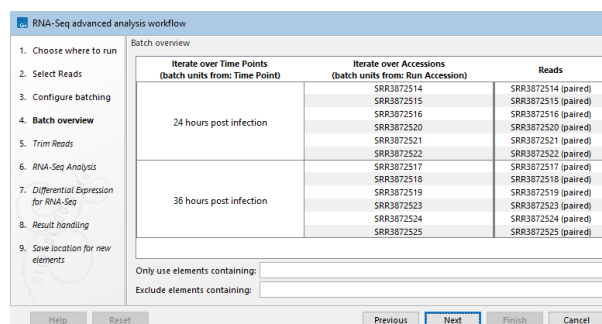


Figure 14.55: The batch overview for a workflow with two Iterate elements. The names assigned to the two columns containing the batch unit organization are the names of the corresponding Iterate elements.

Further configuring Iterate elements

Double-clicking on an Iterate element opens the configuration dialog, which contains the options listed below (figure 14.56). The default settings are relevant for most uses of the Iterate element.

1. **Number of coupled inputs** The number of separate inputs for each given iteration. These inputs are "coupled" in the sense that, for a given iteration, particular inputs are used

together. For example, when sets of sample reads should be mapped in the same way, but each set should be mapped to a particular reference (figure 14.57).

2. **Error handling** Specify what should happen if an error is encountered. The default is that the workflow should stop on any error. The alternative is to continue running the workflow if possible, potentially allowing later batches to be analyzed even if an earlier one fails.
3. **Metadata table columns** If the workflow is always run with metadata tables that have the same column structure, then it can be useful to set the value of the column titles here, so the workflow wizard will preselect them. The column titles must be specified in the same order as shown in the workflow wizard when running the workflow. Locking this parameter to a fixed value (i.e. not blank) will require the definition of batch units to be based on metadata. Locking this parameter to a blank value requires the definition of batch units to be based on the organization of input data (and not metadata).
4. **Primary input** If the number of coupled inputs is two or more, then the primary input (used to define the batch units) can be configured using this parameter.

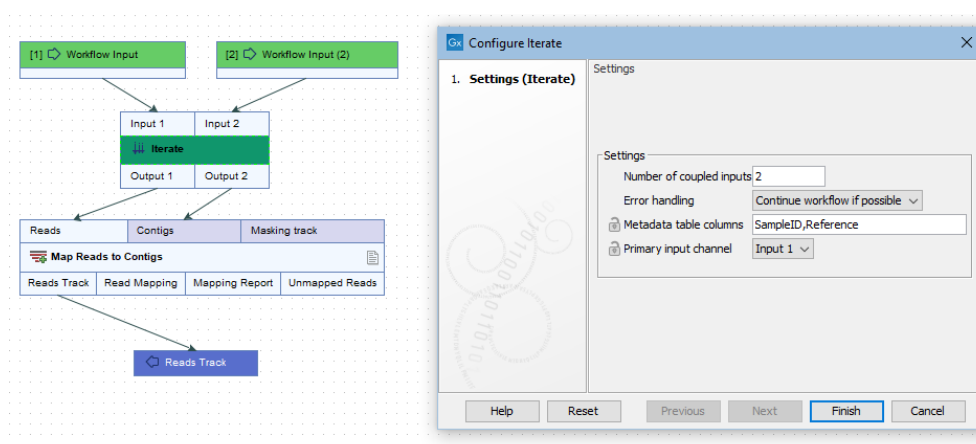


Figure 14.56: The number of coupled inputs in this simple example is 2, allowing each set of sample reads to be mapped to a particular reference, rather than using the same reference for all iterations.

Configuring Collect and Distribute elements

By default, a Collect and Distribute element has one output channel. In this case, all results from the iteration block are collected and passed to downstream steps of the workflow.

More than one output channel can be configured by entering terms in a comma separated list in the **Outputs** field (figure 14.58). The number of terms determines the number of output channels. Connections between these output channels and input channels of downstream elements determine how data should be distributed in the following stage of the workflow.

If the Collect and Distribute element has more than one output channel, the path taken by a given element is determined by the value in the metadata column specified when launching the workflow. This column can be preconfigured in the **Group by metadata column** setting.

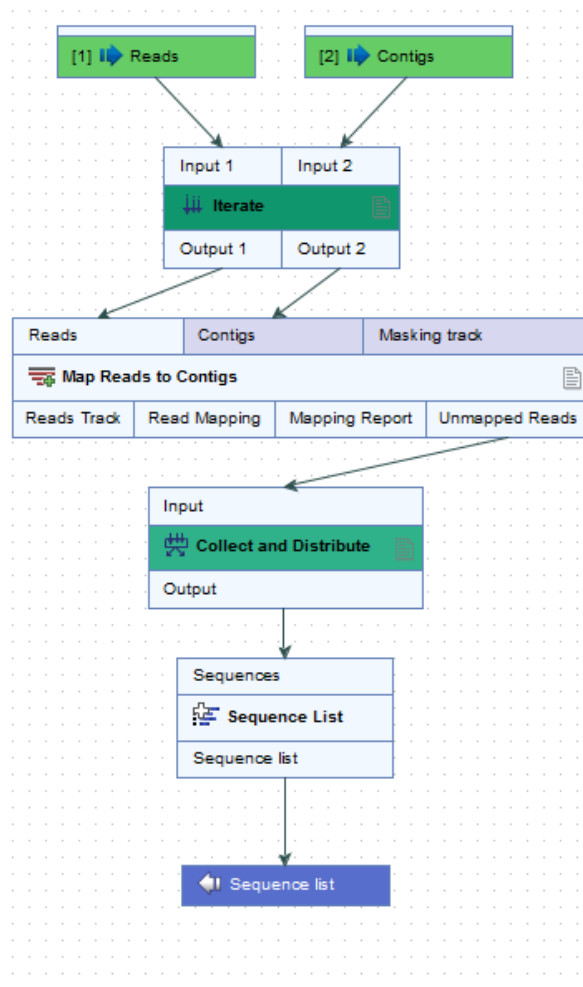


Figure 14.57: Reads can be mapped to specified contigs due to the 2 input channels of the Iterate element. Using this design, a single sequence list containing all the unmapped reads from all the initial inputs is generated. That would not be possible without the inclusion of the Iterate and Collect and Distribute elements.

For example, when launching the workflow in figure 14.59, a metadata column called "Type" was specified for defining which samples were cases and which were controls. The iteration units were defined by the contents of the "ID" column (figure 14.60).

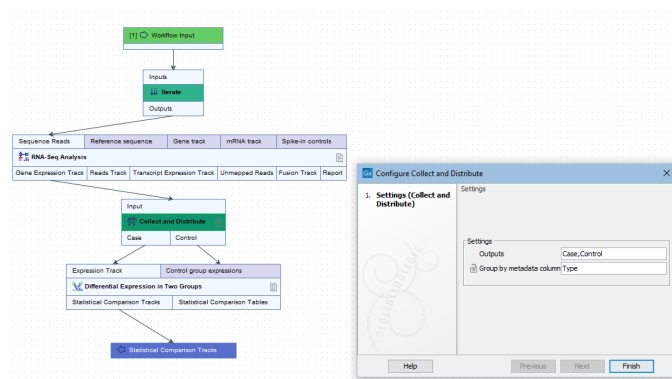


Figure 14.58: A comma separated list of terms in the Outputs field of the Collect and Distribute element defines the number of output channels and their names.

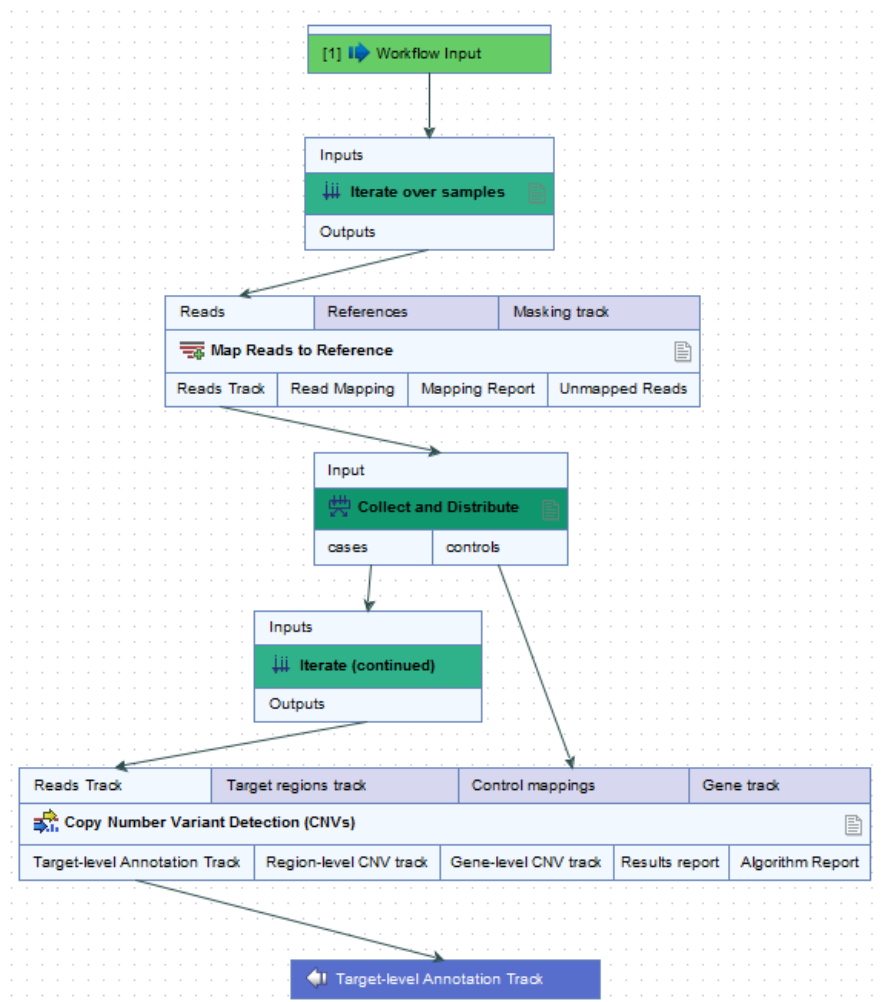


Figure 14.59: In this workflow, each case sample is analyzed against all of the control samples.

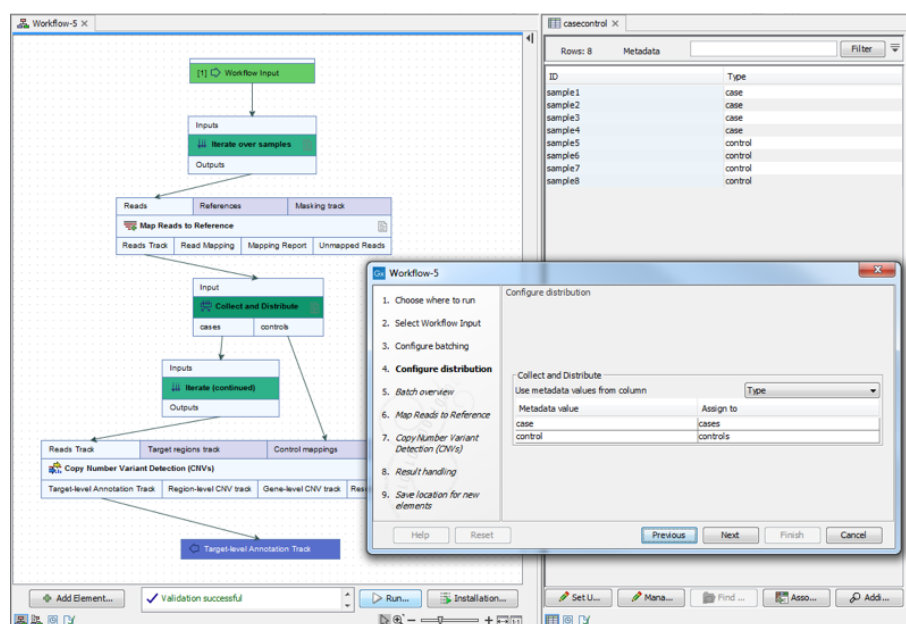



Figure 14.60: Contents of the metadata column "Type" define which samples are cases and which are controls. Iteration units are defined by the contents of the "ID" column.

Fork

A choice between running particular parts of an analysis can be offered by including one or more **Fork** () elements in a workflow.

Including Fork elements in workflows can also help decrease the number of workflows that need to be maintained, as multiple analysis paths can be included in a single workflow, with only one or some of those paths being taken when the workflow is run.

For example, when the workflow shown in figure 14.61 is launched, a choice between "Quality" and "Quality and Vector" is offered in the launch wizard. Choosing "Quality" means the data will flow down the path containing the "Trim on Quality" element, while choosing "Quality and Vector" means the data will flow down the path containing the "Trim Vector and on Quality" element.

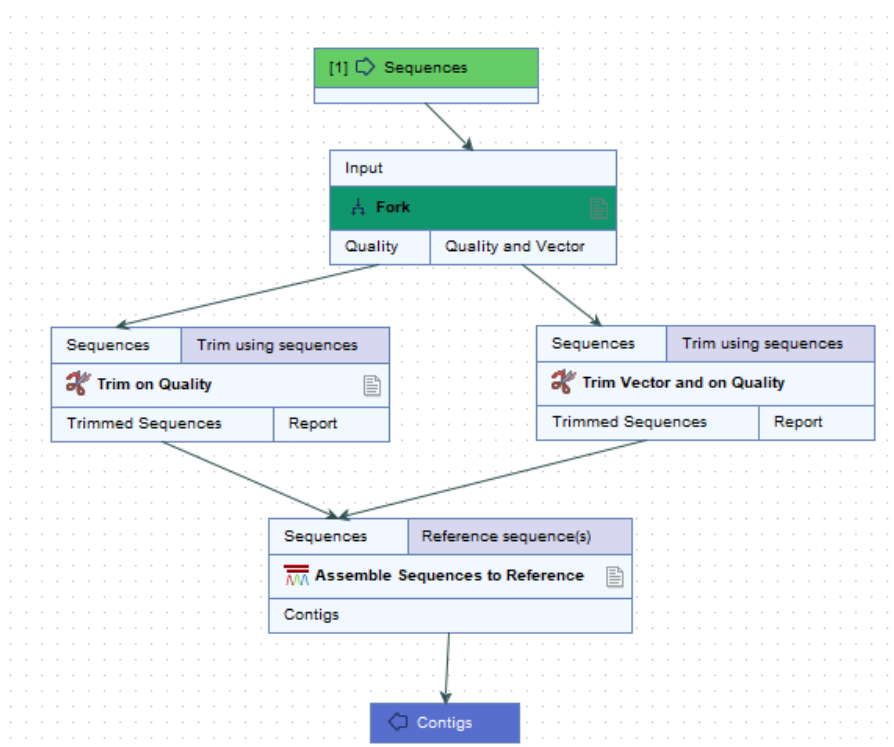


Figure 14.61: A simple workflow with a Fork element. When launched, a choice is offered in the launch wizard for which path the analysis should follow.

Configuring Fork elements

The configuration of Fork elements determines what is shown in the "Specify workflow paths" launch wizard step (figure 14.62 and figure 14.63). Specifically:

- **The name of the Fork element** The element name is used in the launch wizard to describe the choice being made.
- **The configurable fields in the Fork element** The configurable fields are used to define the available options and, if desired, specify a default:

- **Path names** The list of choices of downstream paths from this Fork element, entered as a comma delimited, text list. These are presented in a drop-down list in the launch wizard. An output channel is added to the Fork element for each path name.
- **Selected path** One of the configured path names, which then will be used as the default. If this field is left blank, the first of the paths in the "Path names" list will be pre-selected in the wizard when launching the workflow from a *CLC Workbench*.

Renaming and configuring workflow elements is described in section 14.2.2.

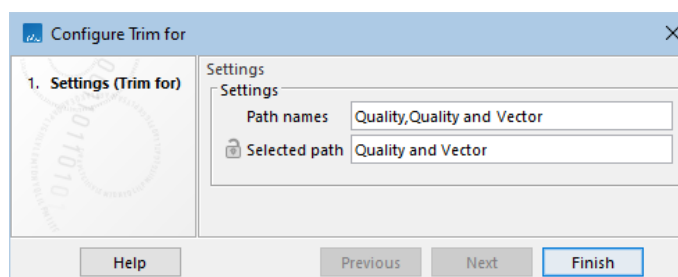


Figure 14.62: A Fork element renamed as "Trim for" is open for configuration. The wizard step seen when launching a workflow with this Fork element is shown in figure 14.63.

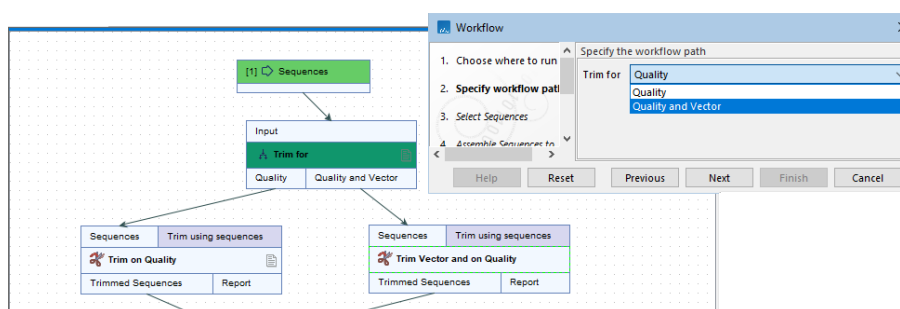


Figure 14.63: This workflow has a single Fork element, renamed as "Trim for". The path names, "Quality" and "Quality and Vector", (see figure 14.62) are listed in the "Specify workflow path" launch wizard step (top right)

Flexibility potential using Fork elements

Fork elements provide a great deal of flexibility in the types of choices that can be offered when launching workflows. Examples include:

- Providing the choice of following one of several possible analysis paths (figure 14.63).
- Providing the choice of following one of several possible analysis paths or following multiple analysis paths (figure 14.64).
- Providing the choice of whether or not to run a particular part of the analysis (figure 14.65).

When a workflow contains multiple Fork elements, all the corresponding choices are presented in a single "Specify workflow paths" launch wizard step (figure 14.65).

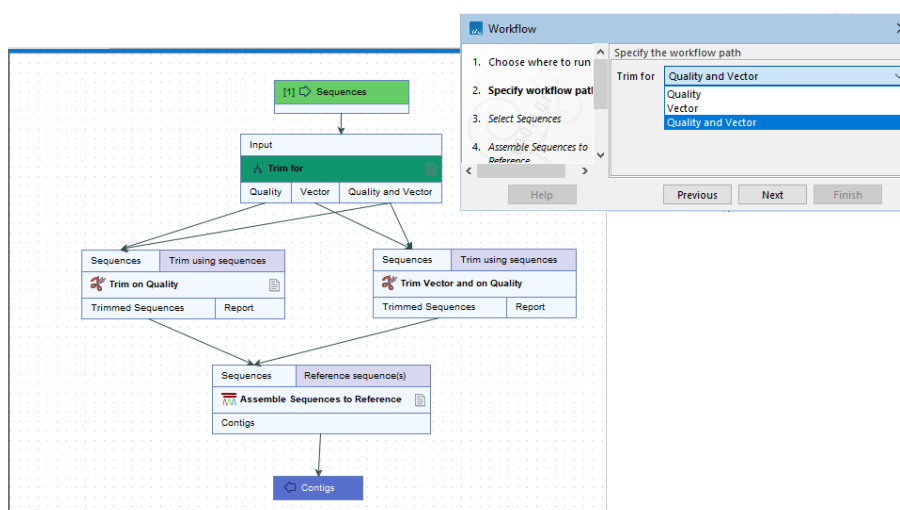


Figure 14.64: When launching this workflow, the choice can be made to trim sequences based on quality, trim for vector sequence, or trim for both.

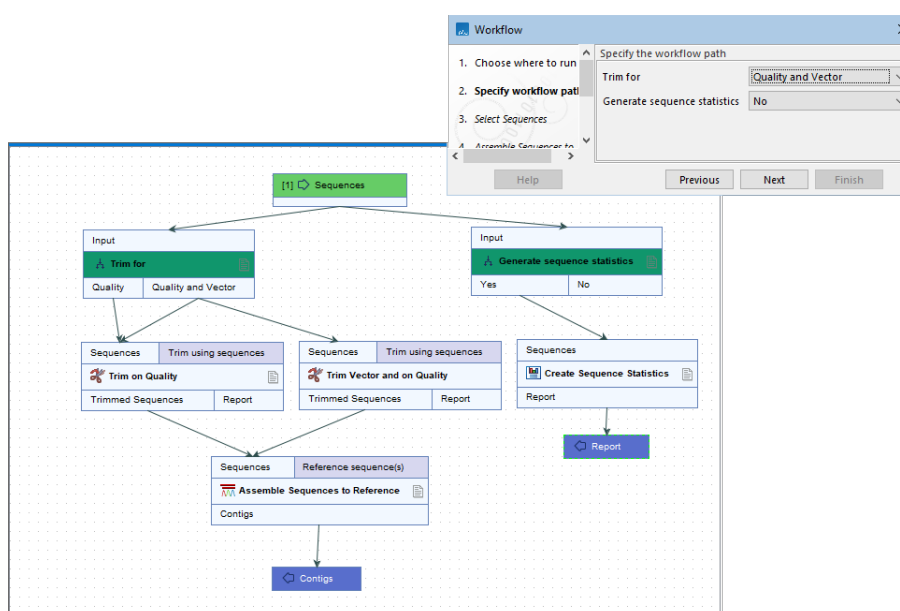


Figure 14.65: There are two Fork elements in this workflow, and thus two choices will need to be made when launching it. Both are presented in the "Specify workflow path" launch wizard step. The Yes/No choice for the "Generate sequence statistics" option determines whether or not the Create Sequence Statistics analysis will be run.

Branching elements

Branching elements control the path that data takes through a workflow.

Branch on Sequence Count

Branch on Sequence Count (🔍) elements are used when downstream handling of sequence lists should depend on the number of sequences in the list.

The sequence list provided as input will flow through the Pass or the Fail output channel depending on whether the number of sequences meets the condition specified in the branching element (figure 14.66).

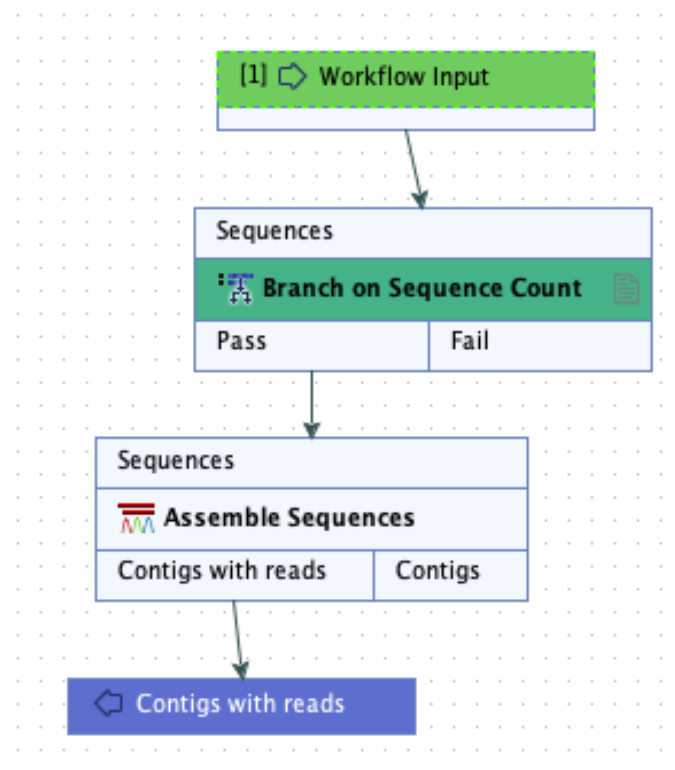


Figure 14.66: If the sequence list provided as input meets the condition specified in a Branch on Sequence Count element, it will flow through the Pass output channel and be used in the Assemble Sequences step. Otherwise, it will flow through the Fail output channel, where here, it would not be processed further.

In the Branch on Sequence Count configuration dialog (figure 14.67), the configuration options are:

- **Comparison** The operator to use for the comparison: \geq , $=$ or \leq , offered in a drop-down list.
- **Number of sequences** The number of sequences to use in the comparison.

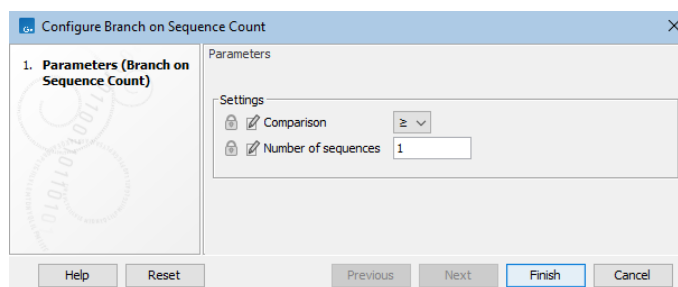


Figure 14.67: Configuration of a Branch on Sequence Count element

Branch on Coverage

Branch on Coverage (🔗) elements are used when downstream handling of read mappings should depend on whether coverage in that mapping meets a specified threshold. A read mapping and a corresponding report containing coverage information are supplied as input. The read mapping flows through the Pass or the Fail output channel depending on whether or not the coverage level in the report meets the condition specified in the branching element (figure 14.68).

Reports generated by the following tools are supported for use with **Branch on Coverage** elements:

- **QC for Targeted Sequencing**
- **QC for Read Mapping** Note: Zero coverage regions in these reports are ignored.

In the Branch on Coverage configuration dialog (figure 14.67), the configuration options are:

- **Type** The type of coverage value, Minimum, Median, Average or Maximum, offered in a drop-down list.
- **Comparison** The operator to use for the comparison: \geq , $=$ or \leq , offered in a drop-down list.
- **Value** The coverage value for the comparison.

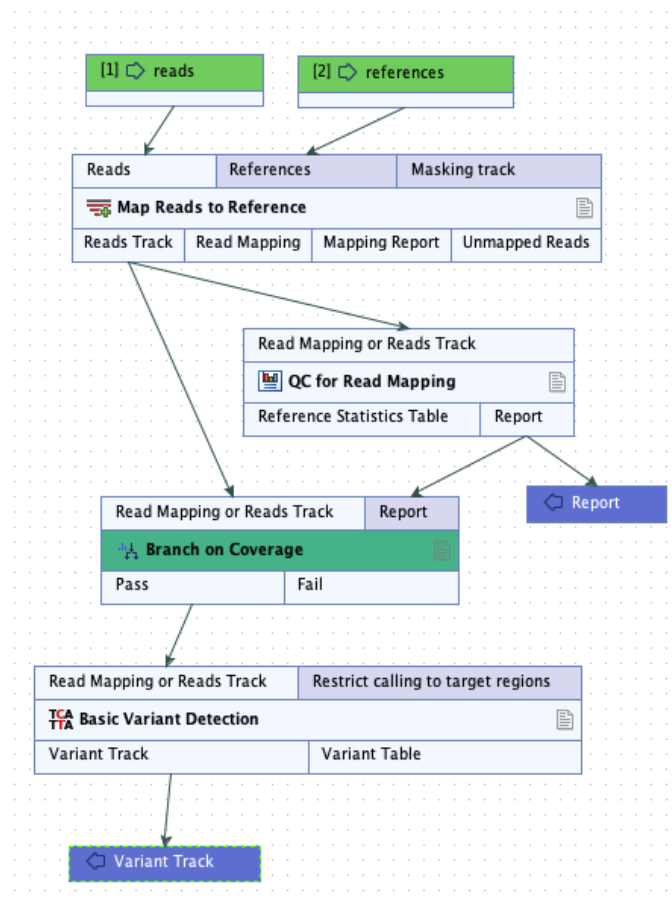


Figure 14.68: A read mapping and a report flow into the Branch on Coverage element. If the coverage according to the report meets the condition configured in the branching element, the mapping will flow through the Pass output channel and be used in the Basic Variant Detection step. Otherwise, it will flow through the Fail channel, where here, it would not be processed further.

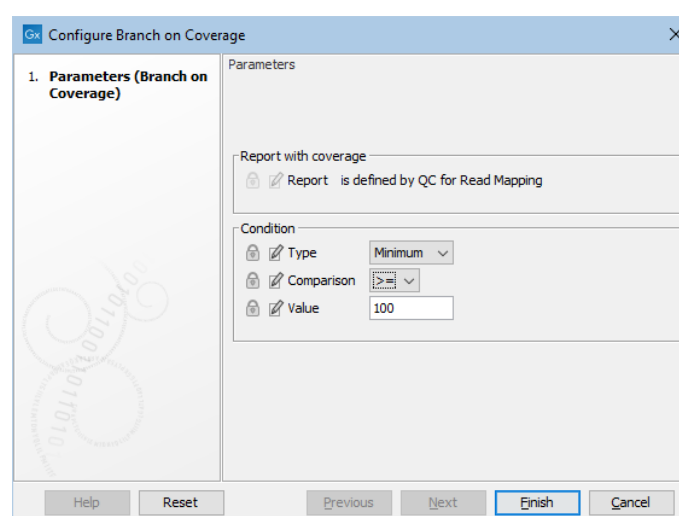


Figure 14.69: Configuration of a Branch on Coverage element

Branch on Sample Quality

Branch on Sample Quality (🔍) elements are used where downstream handling of data should depend on quality aspects of that data. Quality information is read from a corresponding sample report or combined report. Based on that information, the input data flows out of the branching element into one of three downstream workflow paths (figure 14.70).

The flexibility of the **Branch on Sample Quality** element is due to its dependence on information in the "Quality control" subsection of sample reports. The information in that subsection is configured in Create Sample Report (see section 37.6). Combine Reports containing sample reports with quality information can also be used as input to the Branch on Sample Quality element (see section 37.5).

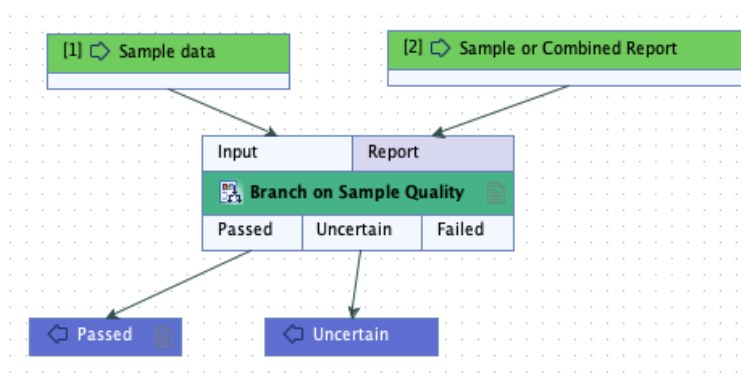


Figure 14.70: *Branch on Sample Quality* has an input channel for the data and an input channel for a corresponding report containing quality information. Data flows through the *Passed*, *Uncertain*, or *Failed* output channel, depending on quality information in the report.

For each quality condition configured in the **Create Sample Report** tool, a color, yellow or red, can be specified to be assigned when the condition is not met. This information is then used by this branching element as follows:

1. If a condition assigned **red** is not met, the data flows through the **Failed** channel. Where all such conditions are met, then:
2. If a condition assigned **yellow** is not met, the data flows through the **Uncertain** channel. Where all such conditions are met, then:
3. The data element flows through the **Passed** channel.

Consequently, if a report without a "Quality control" section is provided, the data element will flow through this channel.

The branching elements **Branch on Coverage** and **Branch on Sequence Count** also control the flow of data through a workflow based on quality measures, but they do not have the flexibility of the Branch on Sample Quality element. Specifically, they allow a single quality condition to be specified for a single data type, with data flowing into one of two downstream paths through the workflow.

14.2.6 Track lists as workflow outputs

Track lists can be made and saved in workflows by doing the following (see figure 14.71:

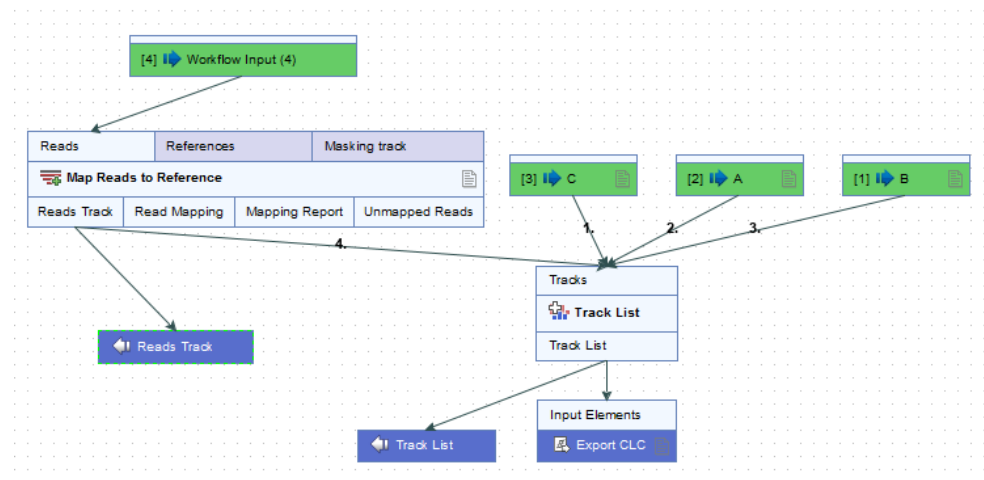


Figure 14.71: This workflow will save and export a track list containing pre-existing data stored in a CLC data area, as well as data generated by the workflow. For workflows containing the Track List element to work, it is mandatory that the data generated by the workflow and included in the Track list is also saved as an independent track.

- Add a Track List element to the workflow.
- Connect an Output or Export element to the Track List element.
- If you wish to include tracks that are available from within the workflow, add Input elements for each of these (here A, B and C) and connect them to the Tracks input channel. Any tracks based on a compatible genome can be added to a given track list.
- To include tracks generated within the workflow, connect the Output channels of elements producing these tracks to the Tracks input channel of the Track List element (in this example, the Reads Track generated by the Map Reads to Reference element).
- Remember to save all tracks generated by the workflow and used in the Track List: connect an Output element to each output channel connected to the Tracks input channel. This is needed because to view a track list, all the tracks referred to by it must be saved to a CLC data area. When this is not the case, a warning message "Track List: all tracks must also be workflow outputs" is displayed, and the **Run** and **Create Installer** buttons are disabled
- Optionally, configure the order the tracks should be taken into the Track List element. This order is the order the tracks will be shown in the track list that is generated (see section 14.1.2). Note that if a track list contains a variant track, then when the track list is opened, the variant table opens in a split view with the track list. If a track list contains several variant tracks, then the one highest in the list is the one that will have its table opened in the split view.

14.2.7 Input modifying tools

A few tools in the *CLC Workbench* are input modifying, meaning that when the tool is run from the Tools menu, it directly manipulates the input provided, instead of generating a new element

as output. However, when such tools are used in a workflow context, they do generate a new element as output¹.

Examples of such tools are:


- Find Open Reading Frames
- Gaussian Statistical Analysis
- Pfam Domain Search
- Proportion-based Statistical Analysis

14.3 Launching workflows individually and in batches

When a workflow is launched, a wizard opens that will take takes you step by step through launching the workflow, including supplying the data to be analyzed, configuring any available options, specifying where outputs should be saved, etc.

Workflows stored in a CLC data location, i.e. available from the Navigation Area, can be launched by opening them, and then clicking on the **Run Workflow** button at the bottom, right hand side of the Workflow Editor.

Installed workflows and template workflows can be launched in the following ways:

- Double click on the workflow name in the Workflows tab in the Toolbox panel, which is in the bottom, left side of the Workbench.
- Select the workflow from the Workflows menu at the top of the Workbench.
- Use the Quick Launch  tool (see section 12.1).

Workflow inputs

Data to be used in a workflow analysis can be selected from the Navigation Area or can be imported on-the-fly from files stored elsewhere. The specific options available depend on how the workflow was configured by the author. Using on-the-fly import, the first action taken when the workflow is run is to import the specified data.

When **Select files for on-the-fly import** is selected, the format of the data files must be specified using the drop-down menu beside this option. If configuration options are available for the selected importer, they will be shown in the lower part of the dialog.

If remote locations are available, such as *CLC Server* import/export directories, or AWS S3 buckets, a **Location** drop-down menu will be visible above the file selection area, either in the launch wizard (figure 14.72) or in the **Select files** dialog that opens when a **Browse** button is clicked on.

Note:

¹Prior to version 22.0, input modifying tools behaved the same when run directly or in a workflow context. Workflow elements for such tools were marked in the Input channel and affected output channel by an M in a circle. There were restrictions for such tools in workflows in these older versions. Please see the manual for the version you are running for full details

- To use CLC data stored in an AWS S3 bucket in a workflow analysis, you must choose the option **Select files for on-the-fly import** and choose the format **CLC Format**.
- If you select data from an AWS S3 bucket for an analysis that will be run on your *CLC Workbench* or *CLC Server*, the data will be downloaded from AWS before the analysis begins. For large datasets, this may take some time. Downloading from AWS S3 to a local file system may incur charges from AWS. See [AWS S3 pricing](#).

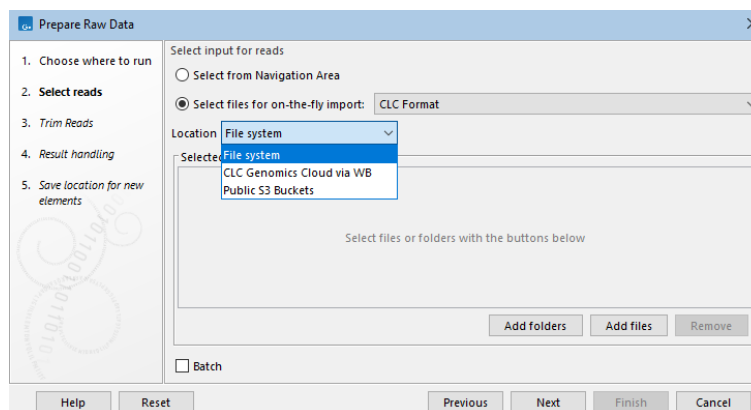


Figure 14.72: Input data is specified when launching a workflow. CLC data can be selected from the Navigation Area. Data stored elsewhere can be selected after choosing the option "Select files for on-the-fly import" and specifying the format of that data.

For information about configuring workflow Input elements when creating or editing a workflow, see section [14.2.3](#).

Workflow outputs

Output and Export elements in workflows specify the analysis results to be saved. In addition to analysis results, a Workflow Result Metadata table can be output. This contains a record of the workflow outputs, as described in section [14.3.1](#).

The history of data elements generated as workflow outputs contains the name and version the workflow that created it. When an installed workflow was used, the workflow build id is also included (see section [2.5](#)).

Launching batches of analyses

When multiple inputs are provided, you can choose to run the workflow multiple times, one time for each input, or for defined sets of inputs that should be analyzed together. This is referred to as running in **Batch mode**, with the inputs to be analyzed together being referred to as **batch units**. This is described in section [14.3.2](#).

In addition, using Iterate and Collect and Distribute control flow elements within a workflow allows for part of a workflow being run once per batch unit, while other parts are run on all the data together. This is described in section [14.3.3](#).

Workflow intermediate results

Workflow intermediate results are generated during the workflow execution. These data elements are needed for use by downstream steps but are then deleted when the analysis successfully completes. While they exist, workflow intermediate results are stored in a dedicated subfolder of the folder that is selected to store outputs when the workflow is launched.

Where intermediate results are stored for jobs run on a CLC Server depends on settings on the server.

14.3.1 Workflow Result Metadata tables

A Workflow Result Metadata table contains a row describing each output generated by the workflow. Each element generated by the workflow has an association to the relevant row.

Creating a Workflow Result Metadata table is enabled by default (figure 14.73).

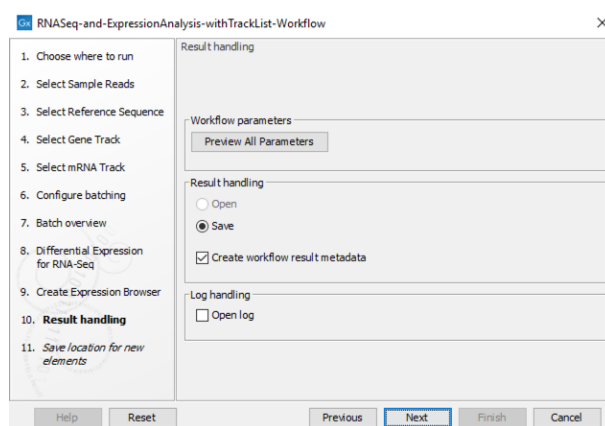


Figure 14.73: The final step when launching a workflow includes an option to create a workflow result metadata table.

The **Import with Metadata** template workflow takes advantage of the Workflow Result Metadata output to make importing data and metadata together simple (see section 14.5.1).

See section 13.3.1 for information on finding and working with data associated with metadata rows.

Workflow Result Metadata tables from batch runs

Workflow Result Metadata tables can be useful when finding results of workflows run in Batch mode due to the large number of outputs that can be produced.

A single table is generated per batch run, containing information about all the results².

A *Batch identifier* column is included for any outputs that were generated as part of a batch run. This includes outputs from workflows run in Batch mode, and also outputs from steps in an iteration loop, i.e. downstream of an **Iterate** control flow element (figure 14.74).

Where batch units are defined using metadata, the Workflow Result Metadata table includes

²There is one exception to this. Where batch units have been defined by the organization of the input data and the outputs are to be saved in the same folders as the inputs, one workflow result metadata table is generated per analysis.

original metadata information, where possible.

See section 14.3.2 for more information on launching workflows in batch mode.

See section 14.2.5 for details about control flow elements and section 14.3.3 for more information on launching workflows containing control flow elements.

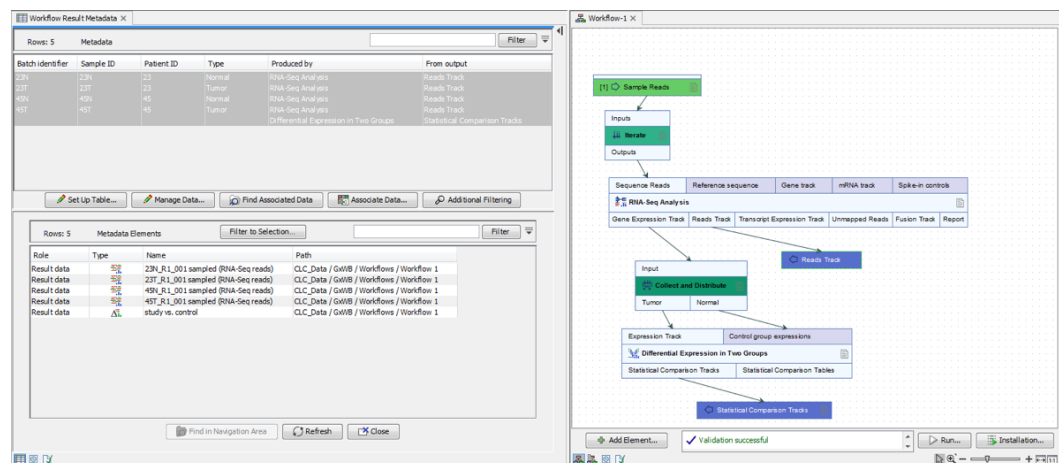


Figure 14.74: The Workflow Result Metadata table, top left, was generated from a run of the workflow on the right. Here, 4 RNA-Seq Analysis runs occurred within the iteration loop (between the Iterate and the Collect and Distribute elements). Those results were then supplied to Differential Expression in Two Groups, which was run once. There are thus 5 rows in the Workflow Metadata Result table. The RNA-Seq Analysis results each have a batch identifier, while the statistical comparison output does not.

14.3.2 Running workflows in batch mode

Running analyses in batches occurs when:

- The Batch checkbox at the bottom of input steps in the launch wizard has been checked, and/or
- The workflow contains one or more **Iterate** control flow elements. Steps downstream of **Iterate** elements and upstream of **Collect and Distribute** elements, if present, are run once for each batch unit (see section 14.2.5).

A batch unit consists of the data that should be analyzed together. The grouping of data into batch units is defined after the inputs for analysis have been selected.

Defining batch units based on the organization of the input data

For simple workflows, batch units can be defined based on how the input elements or files are organized. This is identical to defining batch units when launching a tool, as described in section 12.3.

Here "simple workflows" means workflows with just one analysis input that changes per batch, for example, the sets of sequencing reads to be mapped, where the same reference sequence

is used for every mapping. This equates to a workflow without any Iterate elements being run in Batch mode, or a workflow with just one Iterate element being run (not in Batch mode).

Defining batch units based on metadata

When launching any workflow, batch units can be defined using metadata. For more complex scenarios, this will be the only option. Such scenarios include:

- Where there is more than one level of batch units. This could be:
 - A workflow with more than one Input element, where the inputs to both of these should be grouped into batch units. An example of such a workflow is described in section 14.4.
 - A workflow containing more than one Iterate element.
 - A workflow containing containing an Iterate element that will be run in Batch mode. An example of this is described in the "RNA-Seq and differential gene expression analysis" tutorial, available from <https://resources.qiagenbioinformatics.com/tutorials/RNASeq-DGE-analysis.pdf>.
- Where Iterate or Collect and Distribute elements in the workflow have been configured to require metadata.

Note: When launching a workflow containing analysis steps that require metadata, the metadata provided to define batch units is also used for those analysis steps. For example, in the **RNA-Seq and Differential Gene Expression Analysis** template workflow, metadata provided to define batch units is also used for the Differential Expression for RNA-Seq step.

There are two ways metadata defining batch units can be provided:

1. **Using a CLC Metadata Table** In this case, the data elements selected as inputs must already have associations to this CLC Metadata Table.
If a CLC Metadata Table with data associated to it has been selected in the "Select Workflow Input" step of a workflow, that table will be pre-selected in the "Configure batching" step of the launch wizard. You can specify the column that batch units will be based on. Data associated with the table rows for each unique value in that column make up the contents of the batch units. The contents can be refined using the fields below the preview pane (figure 14.75).
Outputs from the workflow that can be unambiguously identified with a single row of the CLC Metadata Table will have an association to that row added. Outputs derived from two or more inputs with different metadata associations will not have associations to this CLC Metadata Table.
2. **Using an Excel, CSV or TSV format file.** The metadata in the file is imported into the CLC software at the start of the workflow run. Requirements for this file are:
 - The first row must contain column headers.
 - The first column must contain either the exact names of the files selected or at least enough of the first part of the name to uniquely identify each file with the relevant row of the metadata file. If data is being imported on-the-fly, the file name can include file extensions, but not the path to the data.

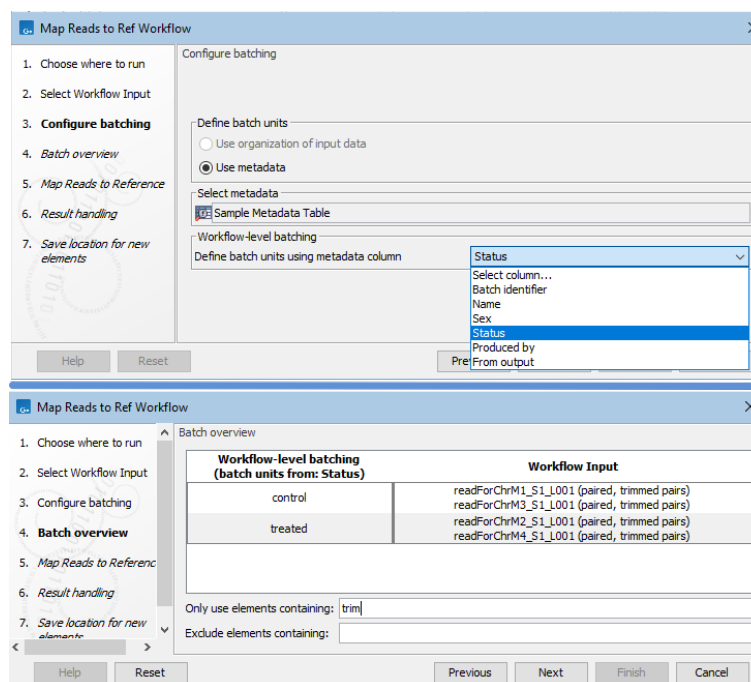


Figure 14.75: A CLC Metadata Table with data associated to it was selected as input to a workflow being launched in Batch mode. In the Configure batching wizard step, the metadata source is pre-configured. The column to base batch units on can be selected (top). The Batch overview step shows the data elements in each batch unit. Here "trim" has been entered in the "Only use elements containing" field, so only elements containing the term "trim" in their names are included in the batch units (bottom).

- A column containing information that defines how the data should be grouped for the analysis, i.e. the information that defines the batch units. In many cases, this column contains sample identifiers. This may be the first column if there are as many batch units as input files.

When the data being imported is paired sequence reads, the first column would contain the names of each input file, and another column would uniquely identify each pair of files (figure 14.76).

If there is a tool in the workflow that requires descriptive information, for example, factors for statistical testing in **Differential Expression for RNA-Seq**, then the file should also contain columns with this information.

For example, if a data element selected in the Navigation Area has the name Tumor_SRR719299_1 (paired) (Reads), then the first column could contain that name in full, or just enough of the first part of the name to uniquely identify it. This could be, for example, Tumor_SRR719299. Similarly, if a data file selected for on-the-fly import is at: C:\Users\username\My Data\Tumor_SRR719299_1.fastq, the first column of the Excel spreadsheet could contain Tumor_SRR719299_1.fastq, or a prefix long enough to uniquely identify the file, e.g. Tumor_SRR719299.

Example: On-the-fly import of single end reads based on metadata

In figure 14.77, a workflow with a single input is being launched in batch mode. The eight

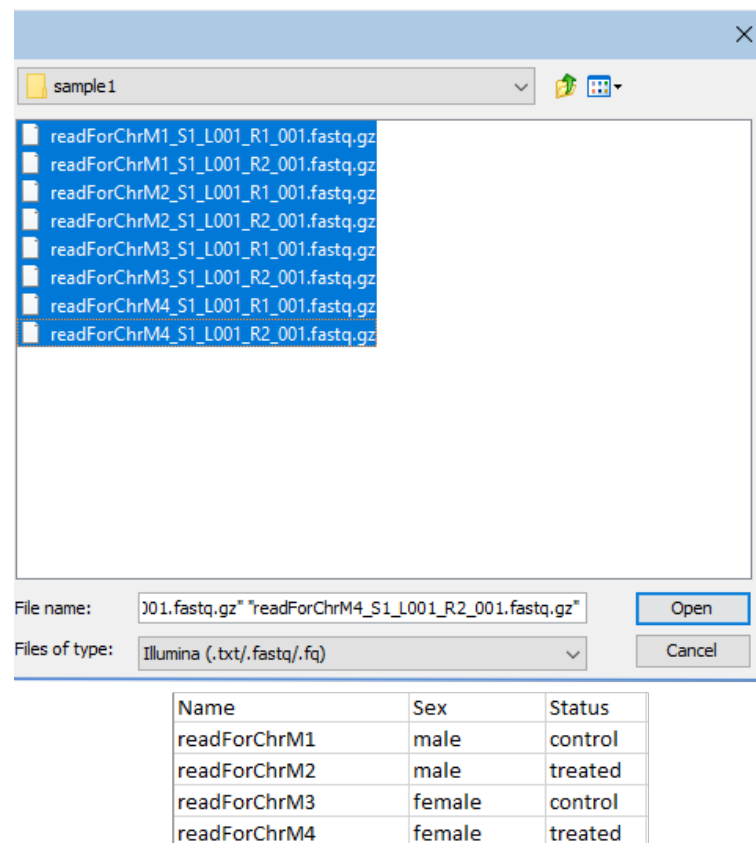


Figure 14.76: Paired fastq files from two samples were selected for import (top). The Excel file with information about this data set contains a header row and 4 rows of information, one row per input file. The contents of the first column contain enough of each file name to uniquely identify each input file. The second column contains sample IDs.

files selected contain Illumina single end reads. This raw data will be imported on the fly using metadata to define the batch units. The metadata column in the Excel file that contains information defining the batch units has been specified. Here, files with the same value in the SRR_ID column will be imported and analyzed together.

Each row in the SRR_ID column has a unique entry, so 8 batch units will be made, with one sequence file in each batch unit. If a column containing fewer unique values was selected, one or more batch units would consist of several files. This is illustrated in figure 14.77.

In the next step, a preview of the batch units is shown. The workflow will be run once for each entry in the left hand column, with the input data grouped as shown in the right hand column (figure 14.78).

Example: On-the-fly import of paired end reads based on metadata

When importing data on-the-fly and organizing batch units based on metadata, the metadata must have a row per file being imported. For paired data, this means at least 2 rows per sample. One column in the file must contain information about which files belong together. This allows the sequence list created to be associated with the relevant information.

The contents of an Excel file with information about 2 sets of paired files containing Sanger data

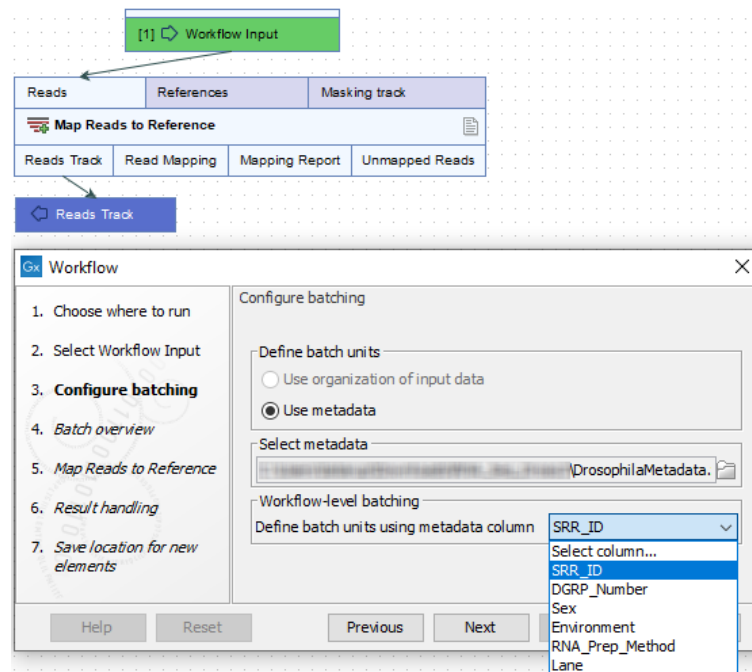


Figure 14.77: Batch units are defined according to the values in the `SRR_ID` column of the Excel file that was selected.

are shown in figure 14.79.

Each row for data that is in the same batch unit must contain the same descriptive information. Where there is conflicting information for a given batch unit, the value for that column will be ignored. If all entries for a given column are conflicting, the column will not appear in the resulting CLC Metadata Table.

Saving results from workflows run in batch mode

When a workflow is run in batch mode, options are presented in the last step of the wizard for specifying where to save results of individual batches (see section 12.3).

If the workflow contains **Export** elements, an additional option is presented, **Export to separate directories per batch unit** (figure 14.80). When that option is checked, files exported are placed into separate subfolders under the the export folder selected for each export step.

14.3.3 Running part of a workflow multiple times

To run a part of the workflow multiple times, once for each batch unit, add an *Iterate* control flow element to the workflow. All elements downstream of an *Iterate* element are run on each batch unit individually, until a *Collect and Distribute* element is encountered. The parts of the workflow downstream of the *Collect and Distribute* element are run on all the data together, or in subsets of the data, as configured in the *Collect and Distribute* element (see section 14.2.5).

For example, the workflow in figure 14.81 would run an RNA-Seq Analysis for each sample separately, and then create a single combined report for the set of samples.

When running on a CLC Server the iterating parts of the workflow is run as separate jobs. For

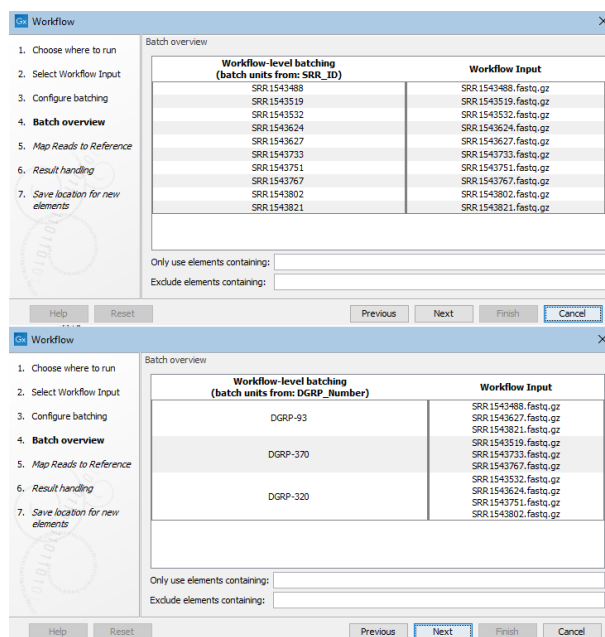


Figure 14.78: The Batch overview step allows you to review the batch units. In the top image, a column called `SRR_ID` had been selected, resulting in 8 batch units, so 8 workflow runs, with the data from one input file to be used in each batch. In the lower image, a different column was selected to define the batch units. There, the workflow would be run 3 times with the input data grouped as shown.

parallel execution of these iterations, job nodes or grid nodes must be available.

When using metadata table to specify the batch units, you are prompted to specify the column that defines how the samples should be grouped for execution. In figure 14.82, grouping by the column "ID" results in the RNA-Seq Analysis tool being run 8 times, once for each sample. Selecting the "Gender" column instead results in the RNA-Seq Analysis tool being run 2 times, once for each value in that column, male and female. The Combine Reports tool runs once, using information from all samples.

The name of workflow elements can be changed. This changes the text displayed in the wizard when the workflow is run. This can be useful if a workflow contains multiple identical elements (figure 14.83).

14.3.4 Specifying reference data in the launch wizard

In workflows where reference data can be selected during launch, you may see individual wizard steps for selecting particular reference data elements, or, where a workflow supports the use of a Reference Data Set and at least one compatible set is available, a launch wizard step called "Specify reference data handling" will be presented.

Where individual data elements will be selected, that data must already be available locally. When a Reference Data Set supplied by QIAGEN (i.e. a QIAGEN Set) is selected, data from that set not already available locally can be downloaded as part of the workflow launch. This is described further below.

The "Specify reference data handling" wizard step has three sections:

	A	B	C	
1	Filename	Sample Name	Status	Color
2	sample1_F	sample1	treated	blue
3	sample1_R	sample1	treated	blue
4	sample2_F	sample2	control	red
5	sample2_R	sample2	control	red

Gx Import with Metadata

1. Choose where to run
2. Select Workflow Input
3. Configure batching
4. **Batch overview**
5. Result handling
6. Save location for new elements

Batch overview

Iterate (batch units from: Sample Name)
sample1
sample2

Only use elements containing:

Exclude elements containing:

Help

Reset

Previous

N

Figure 14.79: An Excel file at the top describes 4 Sanger files that make up two pairs of reads. The "Sample Name" column was identified as the one indicating the group the file belongs to. Information about the relevant sample appears in each row. At the Batch overview step, shown at the bottom, you can check the batch units are as intended.

- **Top** Information about *workflow roles* configured in workflow inputs that expect reference data.

Clicking on the arrow next to the statement about how many workflow roles have been configured reveals details about them, including any data elements defined as defaults (figure 14.85). Where no data element was pre-configured in the workflow for the corresponding input, it will say "Not set" in the "Assigned element(s)" column.

Workflow roles are described further in section 14.2.3.

- **Middle** The options to specify data elements individually or to use a Reference Data Set as the source of reference data for the analysis:

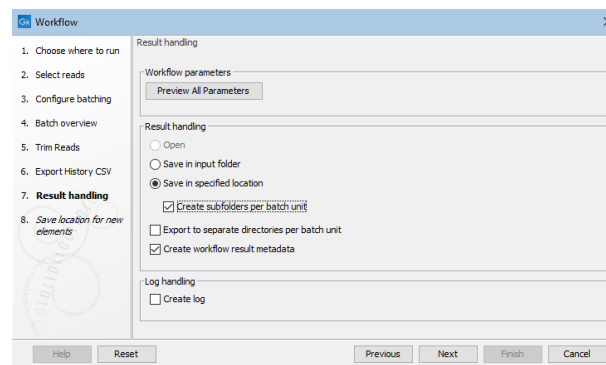


Figure 14.80: Options are presented in the final wizard step for configuring where outputs and exported files from each batch run should be saved.

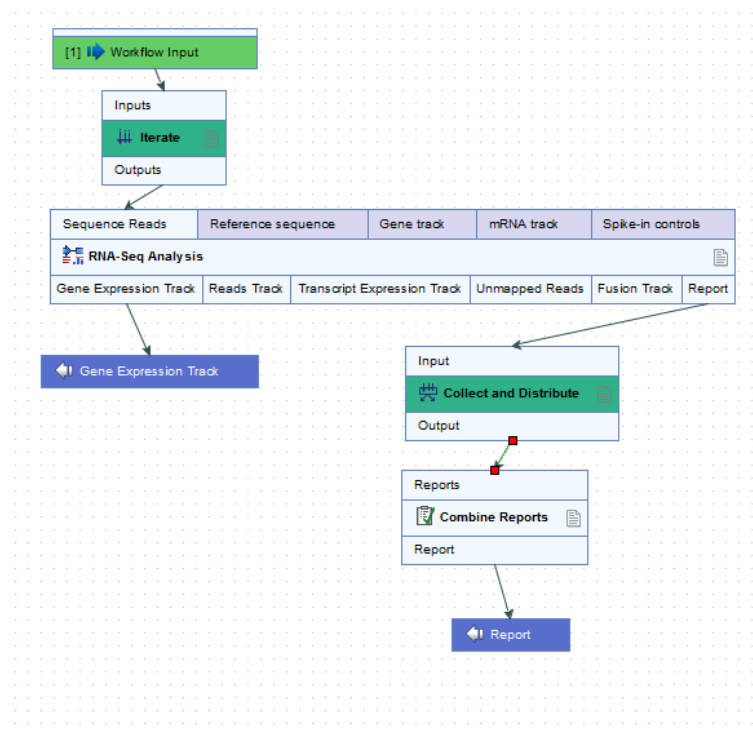


Figure 14.81: The RNA-Seq analysis tool is run once per sample and a single combined report is then generated for the full set of samples.

- **Use specified data elements** Select this option to specify the reference data elements to use individually. This can be useful when there is no appropriate Reference Data Set available.
 Hovering the mouse cursor over the "Use specified data elements" option reveals a tooltip. When the option is enabled, the tooltip contains a list of the data elements that will be used by default (figure 14.84). When the option is disabled, the tooltip contains information about why the option is disabled.
- **Select a reference data set to use** Select this option to use a Reference Data Set to define the reference data to use for the analysis.
- **Bottom** When a Reference Data Set is the source of reference data for the analysis, the

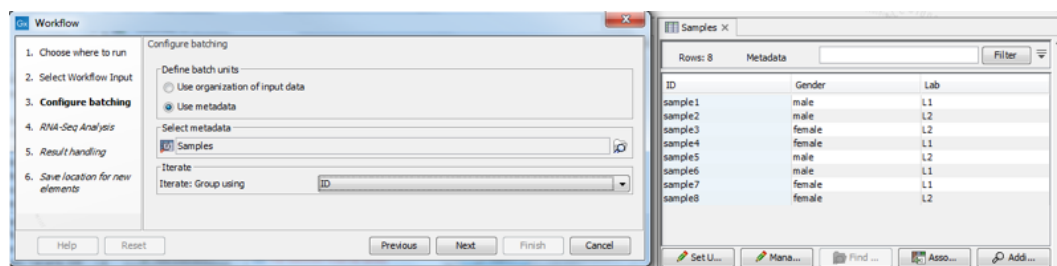


Figure 14.82: With the current selection in the wizard, the RNA-Seq Analysis tool will run 8 times, once for each sample. The Combine Reports tool will run once.

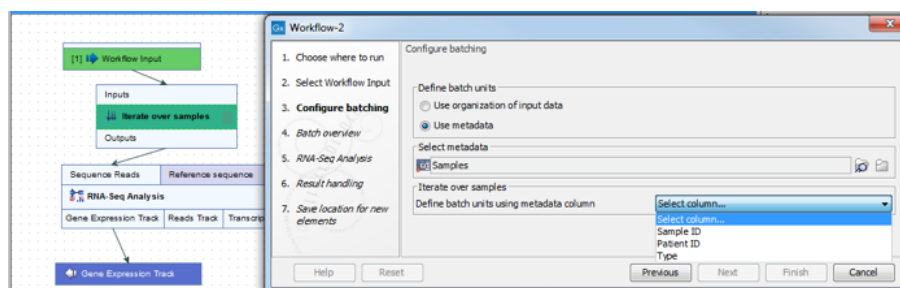


Figure 14.83: The Iterate element can be renamed to change the text that is displayed in the wizard when running the workflow.

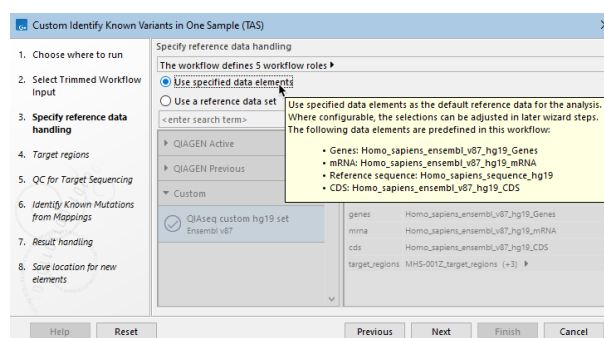


Figure 14.84: The option "Use specified data elements" is enabled in the workflow launch wizard. The mouse cursor has been hovered over this option, revealing a tooltip with information about the data elements configured as the defaults.

lower section of this wizard provides a list of the compatible Reference Data Sets. To be compatible, a Reference Data Set must have a *role* corresponding to each *workflow role* defined in the workflow.

When a Reference Data Set has been selected, information about it is displayed on the right hand side. If a particular *role* in that set has multiple data elements associated with it, clicking on the arrow to the right of that role reveals a list of those elements (figure 14.85).

To search for particular Reference Data Sets, enter terms in the search field just under the "Select a reference set to use" option (figure 14.86). To search for just an exact term, put it in quotes. To see a list of only the Reference Data Sets where all the required elements have already been downloaded, add the term "downloaded" to the search.

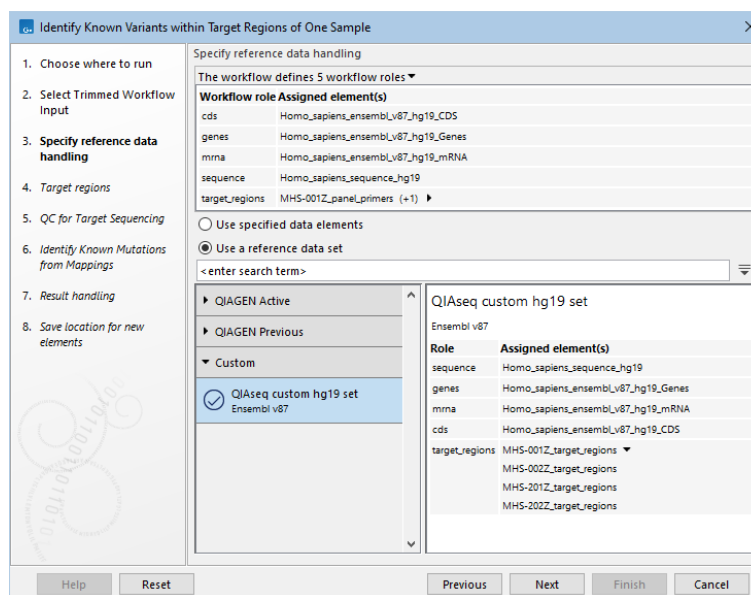


Figure 14.85: The arrow beside the text stating how many workflow roles are defined was clicked upon, revealing information about the roles defined in workflow inputs, and any specific data elements that have been pre-configured. The "Use a reference data set" option has been chosen, and a set to use has been selected. The roles defined in that Reference Data Set are shown on the right. The multiple data elements assigned to the `target_regions` role are revealed by clicking on the small arrow at the right side of that row. The relevant target region element for the analysis is specified in a later wizard step.

Downloading QIAGEN reference data via the workflow launch wizard

When a QIAGEN Set has been selected but one or more elements referred to by that set have not yet been downloaded, a button for downloading that data will be enabled (figure 14.87) . When logged into a CLC Server, an option to download the data to the CLC Server will also be available.

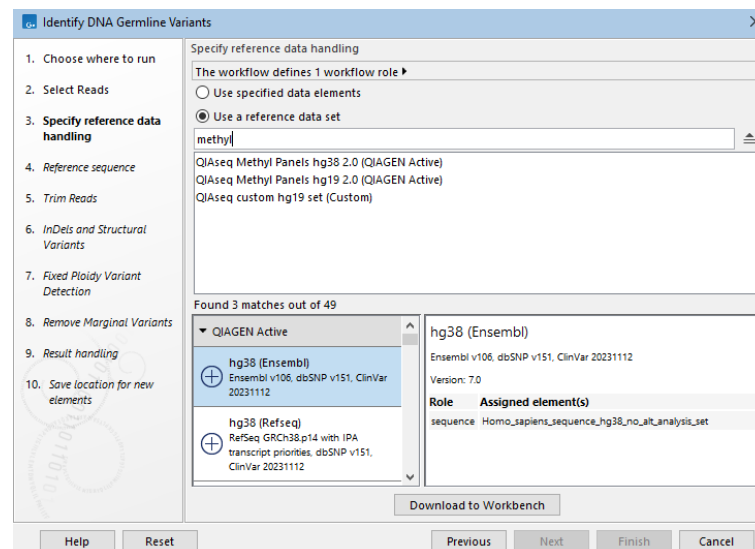


Figure 14.86: Search for Reference Data Sets from within the workflow launch wizard. The list returns all Reference Data Sets containing the terms entered.

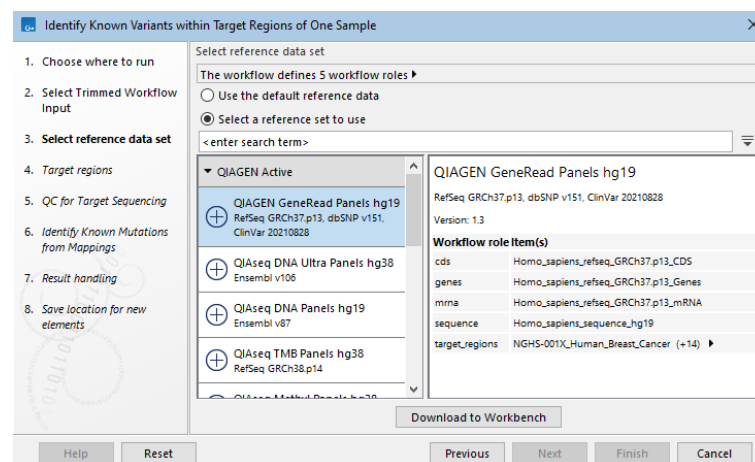


Figure 14.87: A QIAGEN Set that contains all the necessary roles has been selected. A + symbol to the left of the set name indicates that one or more element in that set is not yet downloaded. Such elements can be downloaded by clicking on the "Download to Workbench" button.

14.4 Advanced workflow batching

Fine-tuned control of the execution of whole workflows or sections of workflows can be achieved using metadata describing the relationships between particular samples and using control flow elements in a workflow design. Complex analysis goals that can be met in a straightforward manner include:

- **Grouping the data into different subsets to be analyzed together in particular sections of a workflow.** Groupings of data can be used in the following ways:
 - **Different groupings of data are used as inputs to different sections of the same workflow.** For example, an end-to-end RNA-Seq workflow can be drawn, where the RNA-Seq Analysis tool could be run once per sample and the expression results for

all samples could be used as input to a single downstream tool such as a statistical analysis tool. Or, given Illumina data originating from multiple lanes, QC could be run on the data from each lane individually, then the results for each sample could be merged and mapped to a relevant reference genome, and then a single QC report for the whole cohort could be created. For details, see section 14.3.3 and section 14.4.2.

- **Different workflow inputs follow different paths through parts of a workflow.** Based on metadata, samples can be distributed into groups to follow different analysis paths in some workflow sections, at the same time as processing them individually and identically through other sections of the same workflow. For example, a single workflow could be used to analyze sets of tumor-normal paired samples, where each sample is processed in an identical way up until the comparison step, where the matching tumor (case) and normal (control) samples are used together in an analysis tool.

Configuring Collect and Distribute elements is central to the design of this workflow. This is described in section 14.2.5. Running such workflows is described in section 14.3.3.

- **Matching particular workflow inputs for each workflow run.** Where more than one input to a workflow changes per run, the particular input data to use for each run can be defined using metadata. The simplest case is as described in section 14.4.1. However, more complex scenarios, such as when intermediate results should be merged or parts of the workflow should be run multiple times, can also be catered for using control flow elements (see section 14.2.5).

14.4.1 Batching workflows with more than one input changing per run

When a workflow contains multiple **Input** elements (multiple light green boxes),

A Batch checkbox is available in the launch wizard for each **Input** element attached to a main input channel.

Checking that box indicates that the data supplied for that input should change in each batch run.

By contrast, if multiple elements are selected, and the Batch option is not checked, all elements will be treated a single set, to be used in a single analysis run.

Most commonly, one input is changed per run. For example, in a batch analysis involving read mappings, usually each batch unit would include a different set of reads, but the same reference sequence.

However, it is possible to have two or more inputs that are different in each batch unit. For example, an analysis involving a read mapping, where each set of reads should be mapped to a different reference sequence. In cases like this, batch units must be defined using metadata.

Figure 14.88 shown an example where the aim is to do just this. The workflow contains a Map Reads to Contigs element and two workflow input elements, Sample Reads and Reference Sequences. The information to define the batch units is provided by two Excel files, one containing information about the Sample Reads input and the other with information about the Reference Sequences input. The contents of files that would work for this example are shown in figure 14.89.

Of particular note are:

- The column in the metadata for the primary input that specifies the group the data belongs to. Each group makes up a single batch unit.
- The column in both metadata files that together will be used to ensure that the correct data from each workflow input are included together in a given batch run. For example, a given set of sample reads will be mapped to the correct reference sequence. A column with this name must be present in each metadata file or table.

In figure 14.88, Sample Reads is the primary input: We wish to run the workflow once for each sample, which here, is once for each SRR_ID entry. The Reference sequence to use for each of these batch units is defined in a column called Reference, which is present in both the file containing information about the samples and the file containing information about the references.

14.4.2 Multiple levels of batching

Sometimes it can be useful to batch or iterate over multiple levels. For example, suppose we have Illumina data from 4 lanes on the flow cell, such that each sample has 4 associated reads list. We may wish to run QC for Sequencing Reads per reads list, but the RNA-Seq Analysis tool per sample. A workflow like the one drawn in figure 14.90 allows us to do this, by connecting Iterate elements directly to each other. The top-level Iterate element results in a subdivision (grouping) of the data, and the innermost Iterate results in a further subdivision (grouping) of each of those groups. Note that it is not necessary for the workflow to include a Collect and Distribute element.

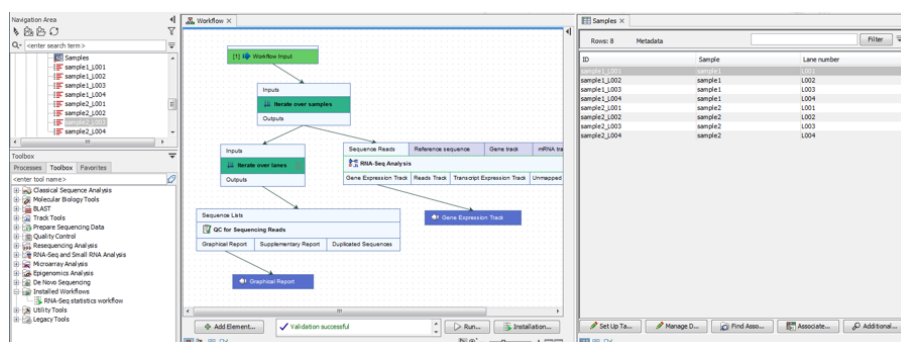


Figure 14.90: The top-level Iterate element results in a subdivision (grouping) of the data, and the innermost Iterate results in a further subdivision (grouping) of each of those groups.

When running the workflow, only metadata can be used to define the groups, because the workflow contains multiple levels of iterations (figure 14.91).

It is always possible to execute a third level of batching by selecting the Batch checkbox when launching the workflow: this will run the whole workflow, including the inner batching processes, several times with different sets of data.

Control flow elements are described in more detail in section 14.2.5.

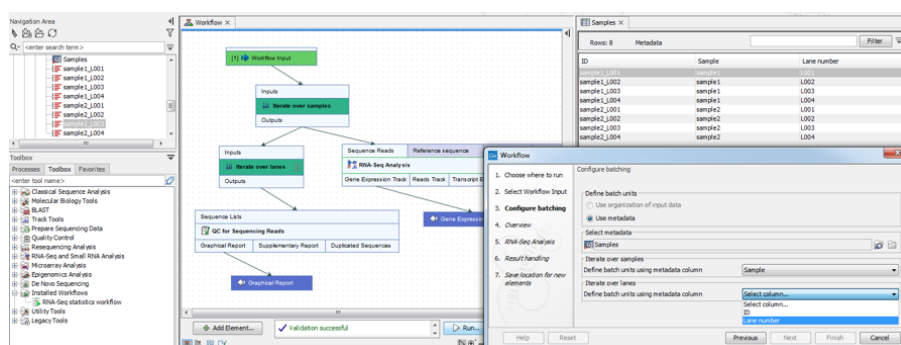


Figure 14.91: When the workflow contains multiple levels of iterations, only metadata can be used to define the groups.

14.5 Template workflows

Template workflows are provided as example workflows (figure 14.92). They can be launched as they are from the under the Workflows menu, or copies can be opened, as described below, allowing you to optimize the workflow to fit your specific application. After making a copy and editing it, you can run the workflow from the Workflow Editor, or you can create an installer and install the workflow to your *CLC Workbench* or *CLC Server*. Links to documentation about these activities are provided later in this section.

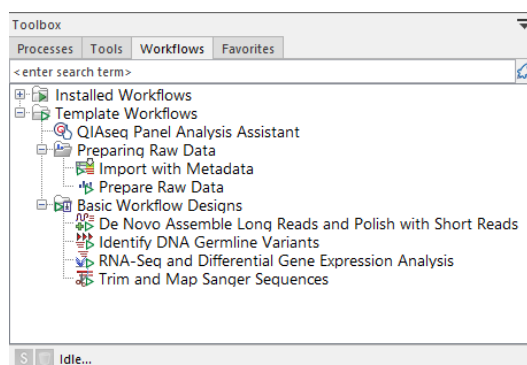


Figure 14.92: The Template Workflows folder in the Workflows tab of the Toolbox

Opening a template workflow for viewing or editing

You can open a copy of a template workflow in the Workflow Editor, where you can view or edit it:

- From under the Workflows tab in the Toolbox in the lower, left side of the Workbench: Right-click on the workflow name and select the option **Open Copy of Workflow** from the menu that appears.

or

- From the Workflow Manager: Open the Workflow Manager by clicking on the **Manage Workflows** button () in the toolbar, and choose the option **Manage Workflows**.

Click on the **Template Workflows** tab and then select the workflow of interest. Then click on the **Open Copy of Workflow** button.

You can specify which settings can be adjusted when launching a workflow, and which cannot, by unlocking or locking parameters in workflow elements. Unlocked parameters can be adjusted when launching the workflow. For locked parameters, the value specified in the design is always used when the workflow is run.

Installed workflows cannot be edited directly, so by locking settings, and installing the workflow, you create a customized version of a template workflow, validated for your purposes, where you know exactly the settings that will be used for each workflow run.

Related documentation

The following manual pages contain information relevant to working with copies of template workflows:

- Configuring workflow elements, including locking and unlocking parameters: section [14.2.2](#)
- Tips for configuring the view of workflows when editing them: section [14.1.9](#)
- General information about editing workflows: section [14.1](#)
- Installing a workflow: section [14.6.2](#)

The template workflows distributed with the *CLC Genomics Workbench* are described after this section. Template workflows distributed with plugins are described in the relevant plugin manual.

14.5.1 Import with Metadata

Using the **Import with Metadata** template workflow, sequence data is imported, and a CLC Metadata Table is created containing the metadata provided in an Excel, CSV or TSV format file. Each imported sequence list is associated with the row in that CLC Metadata Table that contains metadata relating to it.

This is a very simple workflow, containing just an Iterate element connected to an Input element and an Output element (figure [14.93](#)).

The following features of workflows are key to how this works:

- **A CLC Metadata Table with a row recording each workflow output can be created.** This element is named "Workflow Result Metadata". Using this template workflow, it will contain one row per imported sequence list.
The Workflow Result Metadata element is described in more detail in section [14.3.1](#).
- **When batch units are defined using metadata, all the columns in the metadata file are included in the Workflow Result Metadata element.**

Together, these features allow this simple workflow to import data, create a CLC Metadata Table containing information about the data being imported, and establish associations between the imported data and the CLC Metadata Table.

This template workflow is configured to import only sequence data. However, import of other sorts of data can easily be configured, as described in section [14.5](#).

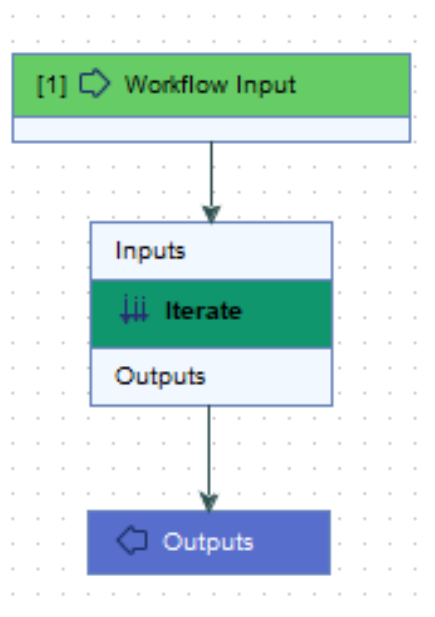


Figure 14.93: The Import with Metadata template workflow

Launching the workflow

The Import with Metadata template workflow is at:

Workflows | Template Workflows | Preparing Raw Data | Import with Metadata (📄)

In the first step, select the format of the data being imported and check the options are configured as needed.

Batch units should be defined using metadata stored in an Excel, CSV or TSV format file. Requirements for this file are described in section 14.3.2.

Once the metadata file has been selected, the column to use to define the batch units can be specified.

In the Batch overview wizard step, the organization of the input files into batch units can be reviewed (figure 14.94).

In the Result handling step, ensure the option **Create workflow result metadata** is checked so the "Workflow Result Metadata" element will be created.

An example of the results of this workflow is shown in figure 14.95).

14.5.2 Prepare Raw Data

The **Prepare Raw Data** workflow performs quality control and trimming of the sequencing reads.

Launching the workflow

The Prepare Raw Data workflow is available from the Toolbox at:

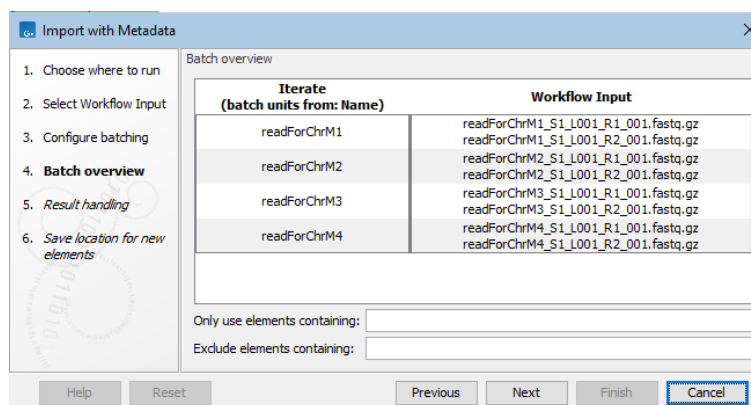


Figure 14.94: In the batch overview step, you can check that input data has been grouped into batch units as expected.

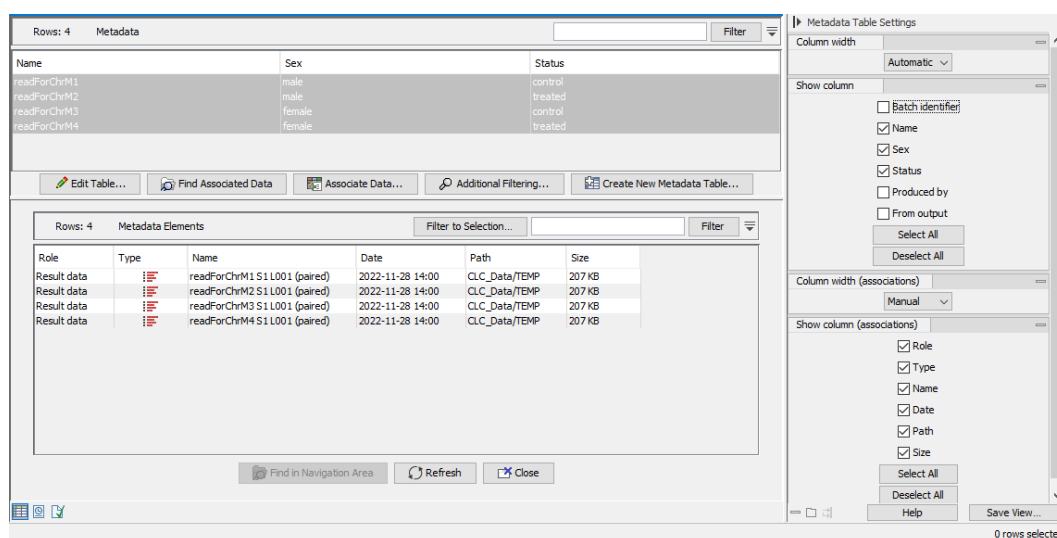


Figure 14.95: The CLC Metadata Table created using the Import with Metadata template workflow has been opened. There is a row per sequence list imported. In this view, some column names in the side panel have been unchecked so that only the sample-specific information is shown. The sequence lists associated with the metadata rows are listed in the bottom panel as a result of selecting all the rows and clicking on the Find Associated Data button.

Workflows | Template Workflows | Preparing Raw Data | Prepare Raw Data

The **Prepare Raw Data** workflow contains two tools, **QC for Sequencing Reads**, see section 28.1, and **Trim Reads**, see section 28.2.

Launch the workflow and step through the wizard.

1. Select the reads to be processed.

If more than one sequence list has been selected, check **Batch** to analyze each input separately. This will generate a trimmed sequence list and QC report for each sequence list provided. See **Running workflows in batch mode**, see section 14.3.2.

If multiple sequence lists are input and the workflow is not run in Batch mode, a single

sequence list and a single QC report are generated, based on all the reads from all the sequence lists.

2. In the next step, choose "Use organization of input data" to specify how to define the batch units.
3. Next, you can review the batch units resulting from your selections above.
4. In the next step, settings for the **Trim Reads**, see section 28.2 tool can be inspected and adjusted (figure 14.96).

If analyzing RNA-Seq data, we recommend trimming polyA tails by checking **Trim homopolymers from 3'**, **polyA**, and **polyT**.

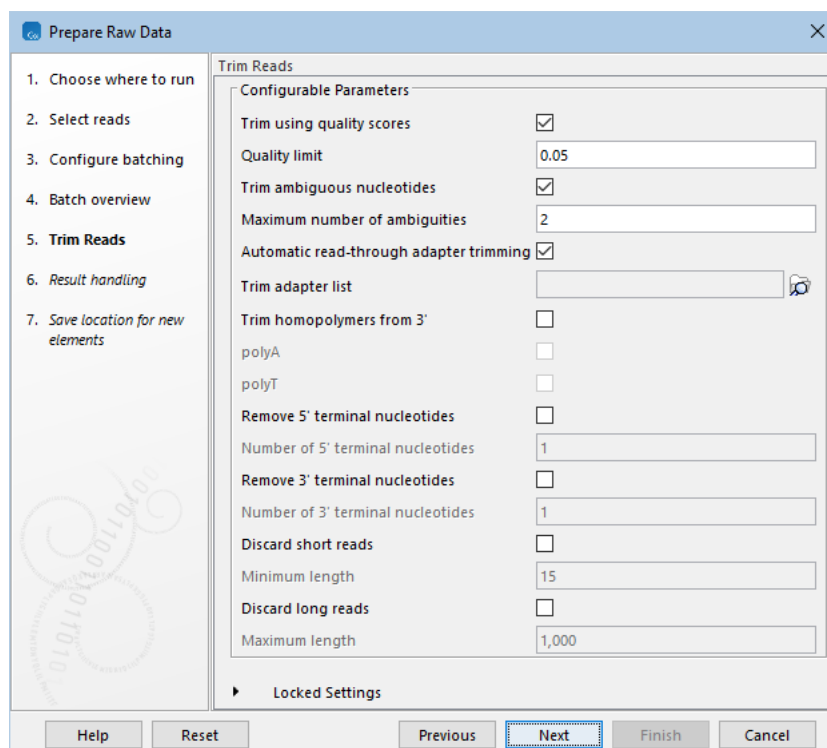


Figure 14.96: *Trim Reads* settings can be adjusted.

5. In the next step, you can click on **Preview All Parameters** to review your settings and specify how results should be handled.
6. In the final step, you can choose a location to save the results to.

Generated outputs

The outputs provided by this workflow are:

1. **Graphical report.** See **QC for Sequencing Reads**, see section 28.1 for details.
2. **Supplementary report.** See **QC for Sequencing Reads**, see section 28.1 for details.
3. **Trimmed sequences.** Sequence list containing trimmed paired reads. See **Trim output**, see section 28.2.7 for details.

4. **Trimmed (broken pairs).** Sequence list containing trimmed broken paired reads. See **Trim output**, see section 28.2.7 for details.
5. **Trim report.** See **Trim output**, see section 28.2.7 for details.

The reports generated should be inspected to determine whether the quality of the sequencing reads and the trimming is acceptable. If the quality is acceptable, the trimmed reads can be used in downstream analyses.

14.5.3 De Novo Assemble Long Reads and Polish with Short Reads

The **De Novo Assemble Long Reads and Polish with Short Reads** workflow performs de novo assembly of long reads and polishes the assembly with high quality short reads.

Launching the workflow

The De Novo Assemble Long Reads and Polish with Short Reads workflow is at:

Workflows | Template Workflows | Basic Workflow Designs  | **De Novo Assemble Long Reads and Polish with Short Reads** 

Launch the workflow and step through the wizard:

1. Select the sequence list containing the long uncorrected reads to be assembled (figure 14.97).
2. Select the short reads to be used for polishing (figure 14.98).
3. Choose assembly parameters. The Minimum contig length can be adjusted to control which contigs are being output. It is particularly important to lower this value if assembling a small microbial genome, as the default may be too high for viral genomes or plasmids. The genome size and ploidy settings become available by checking the PacBio HiFi box. Adjusting these can help improve results when assembling PacBio HiFi reads. (figure 14.99).
4. If the short reads have adapters that have not yet been trimmed, a Trim adapter list can be added (figure 14.100).
5. Select whether unpolished contigs should be included in the output contig list (figure 14.101).
6. In the final step, select a location to save outputs to.

Tools in the workflow and outputs generated

The **De Novo Assemble Long Reads and Polish with Short Reads** workflow contains five tools:

- **QC for Sequencing Reads.** Performs basic QC on the long sequencing reads (section 28.1).
- **Trim Reads.** Trims the short reads for low quality nucleotides (section 28.2).

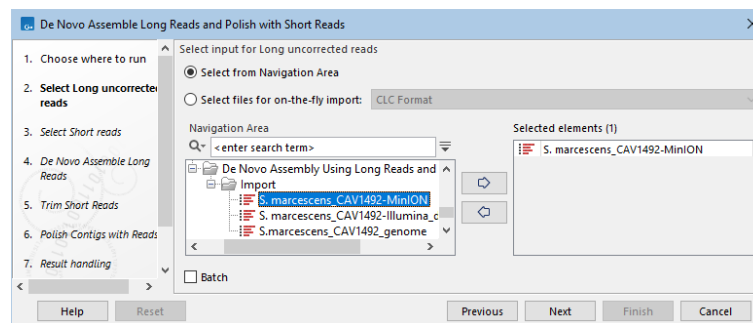


Figure 14.97: Select long uncorrected reads to assemble.

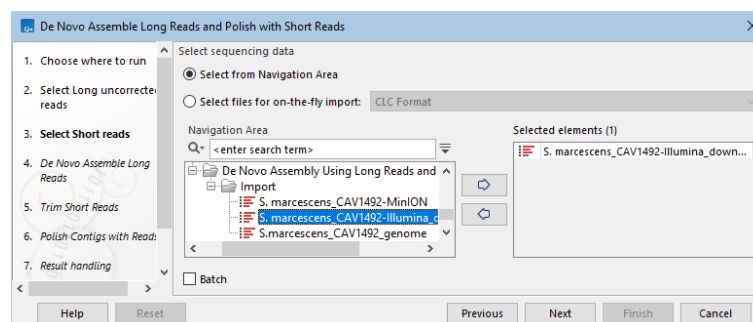


Figure 14.98: Select short reads used to polish the assembly.

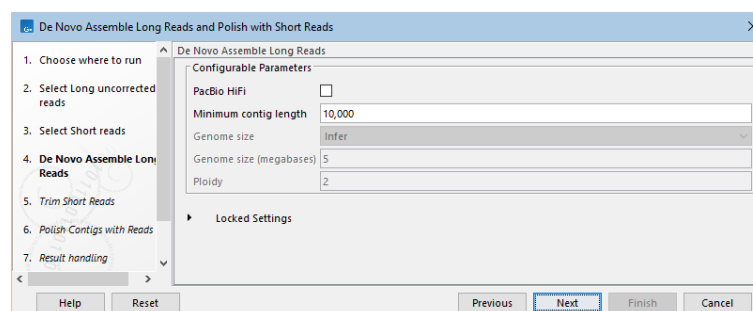


Figure 14.99: Adjust parameters for de novo assembly. The genome size and ploidy settings can only be set if the PacBio HiFi checkbox has been checked. Only do this, if reads are actually PacBio HiFi.

- **De Novo Assemble Long Reads.** Performs de novo assembly of the long input reads (section 35.3). The output contig list is used as input for the polishing step.
- **Polish Contigs with Reads.** Cleans up errors in the de novo assembled contigs using short, high quality reads (section 35.4). Outputs a sequence list containing the updated contigs.
- **Create Sample Report.** Collects reports from the individual tools into a single consolidated sample report (section 37.6). All stand-alone reports are also output in a "QC & Reports" folder.

Customizing the workflow

Template workflows can be easily edited to add, or change analysis steps. See section 14.5 for

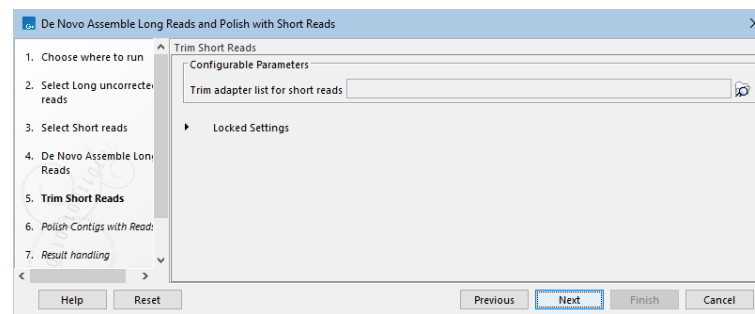


Figure 14.100: Optionally, add a Trim adapter list to use for removing adapters from the short reads.

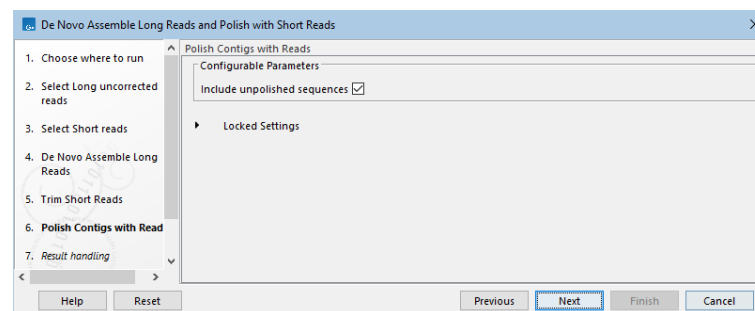


Figure 14.101: Select whether to keep unpolished contigs.

information about how to open a copy of a template workflow for editing.

- **Trim Reads.** Other than changing the default quality and optional adapter trimming, additional trimming options can be set to process the short reads before being used for polishing.
- **Polish Contigs with Reads.** Partial order alignment window size can be adjusted to optimize polishing results.

14.5.4 Identify DNA Germline Variants workflow

The **Identify DNA Germline Variants** workflow includes the minimum analysis steps necessary for variant detection when starting with a set of sequencing reads. This workflow also includes steps to generate reports and visualizations.

We recommend that samples with known variants are used to test and to optimize the workflow settings to fit your specific application. Suggestions for customizations of this workflow are provided at the end of this section.

The workflow requires reference data. Any available, relevant reference sequence can be chosen. Reference data for some organisms can be downloaded using the Reference Data Manager (section 11).

Launching the workflow

The **Identify DNA Germline Variants** workflow is at:

Workflows | Template Workflows | Basic Workflow Designs | Identify DNA Germline Variants

Key steps when launching the workflow include:

1. Selecting the sequence lists containing the reads to analyze.
2. Selecting relevant reference data (figure 14.102).
3. Configuring trimming options.
4. Optionally, restricting the InDels and Structural Variants tool to call variants only in target regions (figure 14.103).
5. Optionally, restricting Fixed Ploidy Variant Detection to call variants only in target regions.
6. Specifying thresholds for the variants to be reported (figure 14.104).
7. Selecting a location to save outputs to.

Tools in the workflow and outputs generated

The tools and outputs provided by this workflow are:

- QC for Sequencing Reads (section 28.1) performs basic QC on the sequencing reads and outputs a report that can be used to evaluate the quality of the sequencing reads. Here the report is included in a combined report, together with other reports, rather than being output as a separate report.
- Trim Reads (section 28.2) trims reads for adapter sequences and low quality nucleotides. The appropriate settings for Trim Reads depend on the protocol used to generate the reads. One of the outputs of this tool is a report, which here is included in a combined report, together with other reports, rather than being output as a separate report.
- Map Reads to Reference (section 30.1) maps reads to a reference sequence.
- InDels and Structural Variants (section 31.10) is used to predict InDels. Identified InDels are used to improve the read mapping during a realignment step. The identified InDels are also output to a track called **Indels-indirect_evidence**. With the default settings, only reads with 3 or fewer mismatches to the reference are considered when identifying potential breakpoints from unaligned ends. This may need to be adjusted if long and/or low quality reads are used as input.
- Local Realignment (section 30.4) uses the predicted InDels from **Indels and Structural Variants** to realign the reads and hence improve the read mapping. The resulting Reads Track is output with the name **Mapped_reads**.
- Fixed Ploidy Variant Detection (section 31.2) calls variants in the read mapping that are present at germline frequencies. In this workflow, the coverage threshold for variant detection is set to 10, meaning that no variants will be called in regions where the coverage is below 10. Similarly, a frequency threshold of 20 percent has been defined.

- Remove Marginal Variants (section 32.1.2) allows post-filtering of detected variants based on frequency, forward/reverse balance and minimum average base quality. The tool outputs the final variant list, called **Variants_passing_filters**. Note that decreasing the thresholds in this tool below the thresholds set in Fixed Ploidy Variant Detection will not result in detection of additional variants.
- Create Track List (section 27.2) outputs a track list **Genome_browser_view** containing the reference sequence, read mapping and identified variants.
- Create Sample Report (section 37.6) compiles the reports from other tools and outputs a **Combined_report**. It is possible to set QC thresholds in this tool, which results in an additional section in the report showing whether QC thresholds were met.

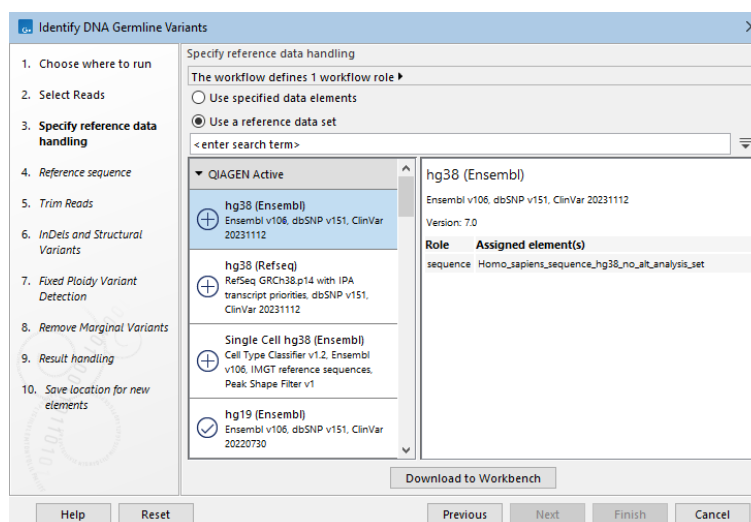


Figure 14.102: In the "Specify reference data handling" wizard step, you can select a Reference Data Set with the relevant reference sequence, as shown here, or choose the "Use specified data elements" option and then choose a particular reference sequence in the following wizard step.

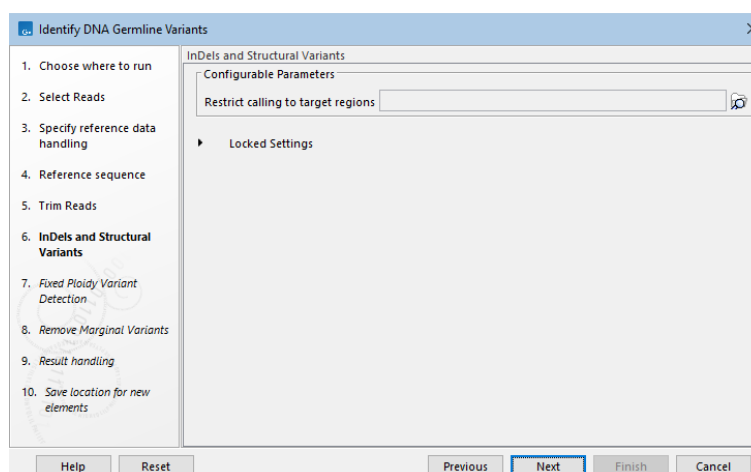


Figure 14.103: If a target region track is specified in this wizard step, the InDels and Structural Variants tool will only call variants within those target regions.

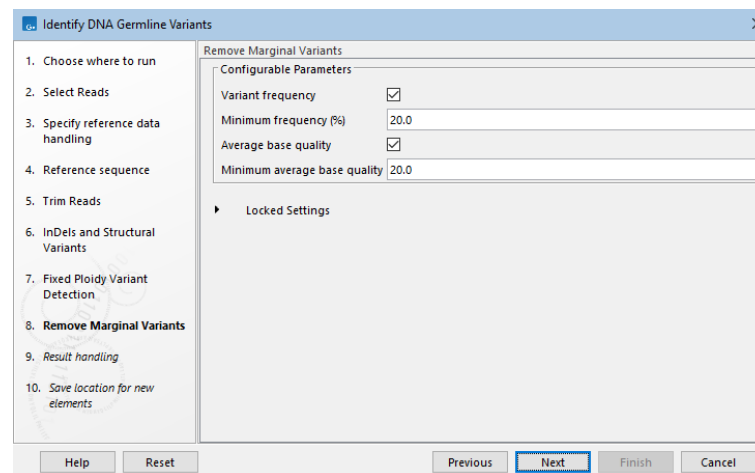


Figure 14.104: Minimum frequency and quality values can be configured to remove marginal variants from the results.

Customizing the Identify DNA Germline Variants workflow

Template workflows (section 14.5) can be easily edited to add or remove analysis steps, change parameter settings, etc.

Some changes that may be of particular interest when working with the **Identify DNA Germline Variants** workflow are:

- **Low frequency variant detection** To detect low frequency variants, the **Fixed Ploidy Variant Detection** workflow element should be replaced with Low Frequency Variant Detection (section 31.3).
- **Targeted sequencing**
 - Several tools in this workflow can be configured to consider only defined target regions, which is useful if a targeted protocol was used to generate the sequencing data. Making this adjustment typically reduces the runtime of the analysis substantially. This change can be made by editing the workflow, but you can also select target regions in the wizard when launching the template workflow.
 - Adding the tool QC for Targeted Sequencing (section 29.1) to the workflow can also be useful. This generates a report with useful metrics, such as the percentage of reads mapped in the target regions.
- **PCR duplicates** For protocols where PCR bias is expected, it can be useful to remove PCR duplicates from the read mapping. This can be achieved with the tool Remove Duplicate Mapped Reads (section 30.6). For inspiration, take a look at the workflow *Identify Variants (WES-HD)* distributed with the Biomedical Genomics Analysis plugin.
- **Annotation of variants** Variants can be annotated with different types of information, such as the gene they occur in and whether the variant changes the coding sequence of a protein. For inspiration, see the workflow *Annotate Variants (WGS)* distributed with the Biomedical Genomics Analysis plugin .

- **Filtering of variants** Variants can be filtered according to information in their annotations, such as their quality and their position relative to a gene. See section 32.1 for details.

For examples of variant filtering cascades, including for limiting the number of false positive variants reported, please refer to the template workflows distributed with the Biomedical Genomics Analysis plugin, such as the [Identify QIAseq DNA Variants](#) workflows.

14.5.5 RNA-Seq and Differential Gene Expression Analysis workflow

The **RNA-Seq and Differential Gene Expression Analysis** workflow calculates gene expression profiles per sample, and then performs differential expression analysis across samples. It also generates various reports and visualizations of the expression profiles and differential expression results. The ability to run parts of the workflow on a per-sample basis and other parts based on all samples, is possible due to the **Iterate** and **Collect and Distribute** workflow elements, see section 14.2.5.

Validation of results should be carried out. Some common workflow customizations are provided at the end of this section.

Inputs to the workflow

To run this workflow, you will need:

- **Trimmed reads** Reads can be trimmed using the **Trim Reads** tool (section 28.2) or the **Prepare Raw Data** template workflow (section 14.5.2).
- **Metadata** containing information about the samples. This can be an Excel, CSV or TSV format file (figure 14.105), or a CLC Metadata Table. The metadata provided should include the factors relevant for differential expression analysis (e.g. treatment level, sex, etc.). For details about providing metadata when launching a workflow, see section 14.3.2.

Sample	Group	Sex
Sample_1	Normal	M
Sample_2	Normal	M
Sample_3	Normal	F
Sample_4	Tumor	M
Sample_5	Tumor	F
Sample_6	Tumor	F

Figure 14.105: Metadata describing samples from a tumor-normal comparison experiment.

Launching the workflow

The **RNA-Seq and Differential Gene Expression Analysis** template workflow is at:

Workflows | Template Workflows | Basic Workflow Designs  **RNA-Seq and Differential Gene Expression Analysis** 

Launch the workflow and step through the wizard.

1. Select the trimmed reads to be processed.

2. In the next steps, select the reference sequence, genes, mRNA, and CDS tracks, and finally, the gene ontology. See the customizations at the end of this section if not all inputs are relevant for your species.
3. Next, choose "Use metadata" for defining the batch units. Select the CLC Metadata Table or the Excel, CSV or TSV format file containing information about the samples, and choose the column used for grouping the reads into batch units (figure 14.106). For further details see section 14.3.2.
4. In the next step, you can review the batch units resulting from your selections above.
5. You can configure next the strand specificity and library type used to produce your data.
6. Specify next the differential expression settings (figure 14.107).
7. In the next step, you can click on **Preview All Parameters** to review your settings.
8. In the final step, choose a location to save the results to.

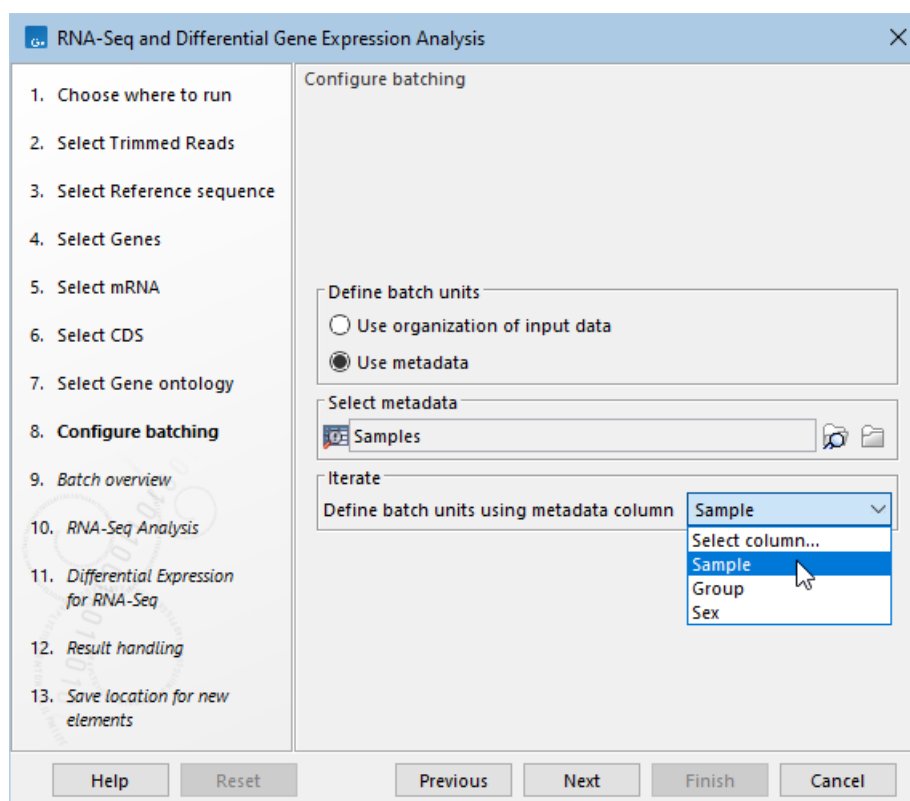


Figure 14.106: After selecting the metadata source, specify the column containing the information that groups the reads appropriately for the RNA-Seq analysis. Usually this would be a column containing a unique identifier per sample.

Tools in the workflow and generated outputs

The **RNA-Seq and Differential Gene Expression Analysis** workflow contains several tools and produces multiple outputs. The workflow has been configured to save many of the outputs to

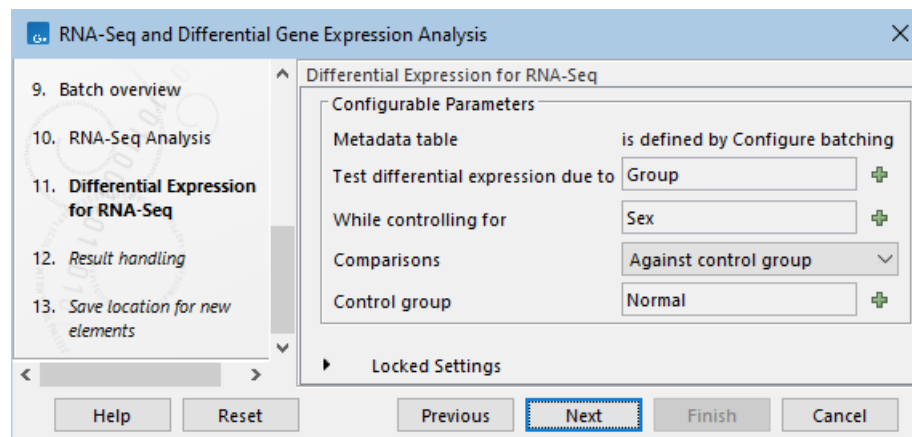


Figure 14.107: Specify the settings for the differential expression analysis. The columns from the metadata provided earlier will be available for selection in relevant options.

subfolders. These are created automatically within the folder that you selected to save results to when launching the workflow. See section 14.2.4 for details.

The following tools produce elements per sample:

- **QC for Sequencing Reads**, see section 28.1, outputs one report that is useful for validating the quality of the reads after trimming. It is saved to the subfolder `QC & Reports/<Batch unit>`.
- **RNA-Seq Analysis**, see section 33.4.1, outputs:
 - One Gene Expression Track containing the gene expression profile. It is saved to the subfolder `Gene Expression Tracks`.
 - One report summarizing the RNA-Seq analysis results. It is saved to the subfolder `QC & Reports/<Batch unit>`.
- **Create Sample Report**, see section 37.6, generates one report summarizing the QC for Sequencing Reads and RNA-Seq Analysis reports for that sample. The sample report is not saved as output. It is provided as input to the Combine Reports tool, in a downstream step of the workflow.

The following tools output elements across all samples. Their outputs are saved to the subfolder `Expression Analysis`:

- **Differential Expression for RNA-Seq**, see section 33.6.4, outputs Statistical Comparison Tracks containing the results of the performed tests. The number of output tracks depends on the settings used for the differential expression analysis (figure 14.107).
- **Gene Set Test**, see section 33.6.7, outputs a Gene Ontology enrichment analysis for each Statistical Comparison Track.
- **Create Expression Browser**, see section 33.2, outputs a single table containing all Gene Expression Tracks and Statistical Comparison Tracks.

- **Create Venn Diagram for RNA-Seq**, see section 33.6.6, outputs a Venn diagram comparing the overlap of differentially expressed genes from the Statistical Comparison Tracks.
- **PCA for RNA-Seq**, see section 33.5.1, outputs a plot containing the projection of the Gene Expression Tracks into two and three dimensions.
- **Create Heat Map for RNA-Seq**, see section 33.5.3, outputs a heat map of the most variable genes in across samples.

The following tools output elements across all samples. Their outputs are saved to the folder selected to save results to when launching the workflow:

- **Combine Reports**, see section 37.5, outputs one report. It takes the individual sample reports generated by Create Sample Report and generates a single report, useful for comparing the individual sample results.
- **Create Track List**, see section 27.2, outputs a track list containing the reference sequence, genes, mRNA, CDS, and the Statistical Comparison Tracks.

Customizing the workflow

Template workflows can be easily edited to add, remove or change analysis steps. See section 14.5 for information about how to open a copy of a template workflow for editing.

- **RNA-Seq Analysis**. The expression profile is at gene level and hence the differential expression is also reported at gene level. If you want to quantify transcript expression instead, use the "Transcript Expression Track" output instead of the "Gene Expression Track" output from the "RNA-Seq Analysis" workflow element.
- **Genes track** and **mRNA track**. RNA-Seq Analysis is configured to use of both genes and mRNA tracks. They are also added as inputs to the "Create Track List" workflow element. The "RNA-Seq Analysis" workflow element can be configured with different "Reference types" and any input workflow elements that are not needed can be removed.
- **CDS track**. A CDS track is added as input to the "Create Track List" workflow element. If you do not have CDS track for your reference genome, remove the "CDS" input workflow element.
- **Differential Expression for RNA-Seq**. The workflow element is configured for whole transcriptome RNA-Seq data. Options are available for other types of RNA data.
- **Gene Set Test**. The workflow element requires a GOA database. If you do not have this for your species, remove the "Gene Set Test" and "Gene ontology" workflow elements.
- **Create Heat Map for RNA-Seq**. The workflow element is configured to use the 25 genes that are most variable. Options are available for choosing other genes.

14.5.6 Trim and Map Sanger Sequences

The **Trim and Map Sanger Sequences** template workflow adds trim annotations to sequences and then maps them to a reference. This workflow generates a stand-alone read mapping and a trimming report.

This workflow is aimed at working with data generated using Sanger sequencing. It is not intended for the analysis of high volumes of data. A maximum of 100,000 sequences can be supplied as input.

Importing Sanger sequences (with trace information) is described in section 22.1.

Launching the workflow

To launch the **Trim and Map Sanger Sequences** template workflow, go to:

Workflows | Template Workflows | Basic Workflow Designs | Trim and Map Sanger Sequences (🔗)

Tools in the workflow and outputs generated

The **Trim and Map Sanger Sequences** template workflow contains two tools (figur 14.108):

- **Trim Sequences.** Adds Trim annotations to sequences. Trimming options can be configured when launching the workflow. A trimming report is generated. See section 22.2 for more information.
- **Assemble Sequences to Reference.** Maps the trimmed sequences to the specified reference sequence(s). Mapping options can be configured when launching the workflow. A stand-alone read mapping is generated. See section 22.4) for more information.

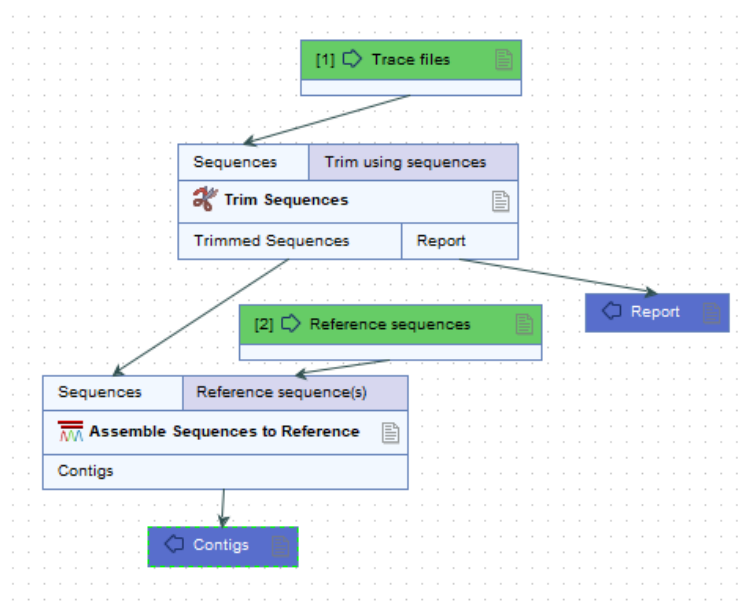


Figure 14.108: The Trim and Map Sanger Sequences template workflow

14.6 Managing workflows

Workflows can be managed from the Workflow Manager:

Utilities | Manage Workflows (⚙️)

or using the "Workflows" button (🔧) in the toolbar and then select "Manage Workflows..." (⚙️).

The Workflow Manager (figure 14.109) lists workflows installed on your system under the Installed Workflows tab. Workflows under the Installed Workflows tab can be configured, renamed and uninstalled, as described below.

Workflows provided by QIAGEN are listed under the Template Workflows tab.

Copies of workflows can be made by clicking on the "Open Copy of Workflow" button for the relevant workflow. This opens an editable copy of the workflow in the Workflow Editor. That workflow can then, if desired, be saved as an element in Navigation Area, and installed so it appears under the Installed Workflows folder of the Toolbox.

Note: Copies of installed and template workflows can also be opened from under the Workflows tab in the Toolbox at the bottom left side of the Workbench. Right-click on the workflow name and choose "Open Copy of Workflow" from the menu that appears.

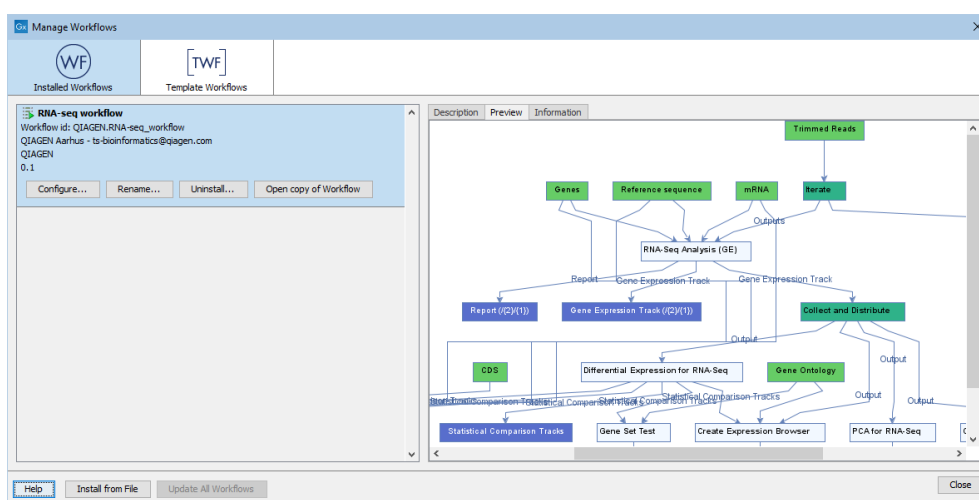


Figure 14.109: An installed workflow has been selected in the Workflow Manager. Some actions can be carried out on this workflow, and a preview pane showing the workflow design is open on the right hand side.

Configure

Clicking on the **Configure** button for an installed workflow will open a dialog where configurable steps in the workflow are shown (figure 14.110). Settings can be configured, and unlocked settings can be locked if desired.

Note: Parameters locked in the original workflow design cannot be unlocked. Those locked using the Configure functionality of the Workbench Manager can be unlocked again later in the same way, if desired.

Parameter locking is described further in section 14.2.2.

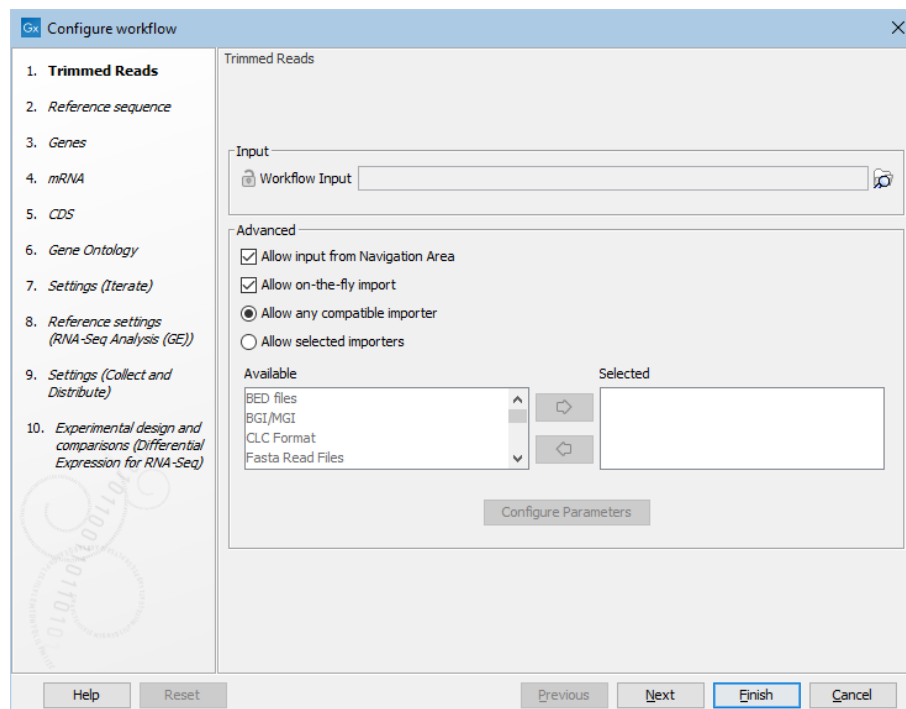


Figure 14.110: Configuring parameters for a workflow.

Note that parameters requiring the selection of data should only be locked if the workflow will only be installed in a setting where there is access to the same data, in the same location, as the system where the workflow was created, or if the parameter is optional and no data should be selected. If the workflow is intended to be executed on a *CLC Server*, it is important to select data that is located on the *CLC Server*.

Rename

Clicking on the **Rename** button for an installed workflow allows you to change the name. The workflow will then be listed with that new name in the "Installed Workflows" folder in the Workflows menu.

The workflow id will remain the same.

Uninstall Use this button to uninstall an installed workflow.

Description, Preview and Information In the right hand pane of the Workflow Manager, are three tabs.

- **Description** Contains the description that was entered when creating the workflow installer (figure 14.111). See section 14.6.2.
- **Preview** Contains a graphical representation of the workflow
- **Information** Contains general information about that workflow, such as the name, id, author, etc. (figure 14.112, and described in detail below).

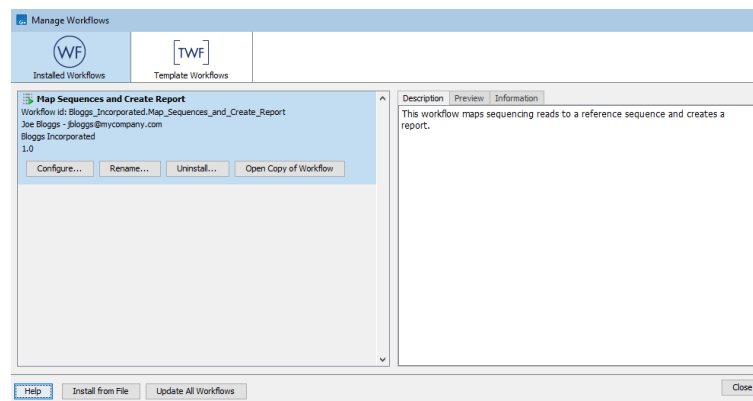


Figure 14.111: The description provided when creating the workflow installer is available in the Description tab in the Workflow Manager.

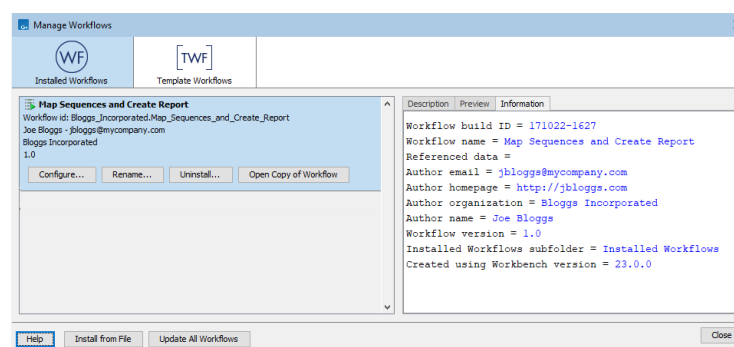


Figure 14.112: The Information tab contains the information provided when the workflow installer was created as well as the workflow build-id.

The Information tab (figure 14.112) contains the following:

- **Workflow build id** The date (day month year) followed by the time (based on a 24 hour time) when the workflow installer was created. If the workflow was installed locally without an installation file being explicitly created, the build ID will reflect the time of installation.
- **Workflow name** The name of workflow
- **Referenced data** If reference data was referred to by the workflow and the option Bundled or Reference was selected when the installer was made, the reference data referred to is listed in this field. See section 14.6.2 for further details about these options.
- **Author email** The email address the workflow author entered when creating the workflow installer.
- **Author homepage** The homepage the workflow author entered when creating the workflow installer.
- **Author organization** The organization the workflow author entered when creating the workflow installer.
- **Author name** The workflow author's name.

- **Workflow version** The version that the author assigned to the workflow when creating the installer.
- **Created using Workbench version** The version of the CLC Workbench used when the workflow installer was created.

14.6.1 Updating workflows

After installing a new version of a *CLC Workbench*, workflows may need to be updated before they can be used. Three situations are described in this section:

- Updating workflows stored in the Navigation area
- Updating installed and template workflows when using a upgraded Workbench in the *same* major version line
- Updating installed workflows when using software in a *higher* major version line

"Major version line" refers to the first digit in the version number. For example, versions 23.0.1 and 23.0.5 are part of the same major release line (23). Version 22.0 is part of a different major version line (22).

Updating workflows stored in the Navigation area

When you open a workflow stored in the Navigation Area that needs to be updated, an editor will open listing the tools that need to be updated, along with additional information about the changes to the tools (figure 14.113).

To update the workflow, click on the **OK** button at the bottom of the editor.



The updated workflow can be saved under a new name, leaving the original workflow unaffected.

Updating installed and template workflows when using an upgraded Workbench in the *same* major version line

When working on an upgraded *CLC Workbench* in the same major version line, installed and template workflows are updated using the Workflow Manager.

To start the Workflow Manager, go to:

Utilities | Manage Workflows ()

or click on the "Workflows" button () in the toolbar, and select "Manage Workflow..." () from the menu that appears.

A red message is displayed for each workflow that needs to be updated. An individual workflow can be updated by selecting it and then clicking on the **Update...** button. Alternatively, click on the **Update All Workflows** button to carry out all updates in a single action (figure 14.114).

When you update a workflow through the Workflow Manager, the old version is overwritten.

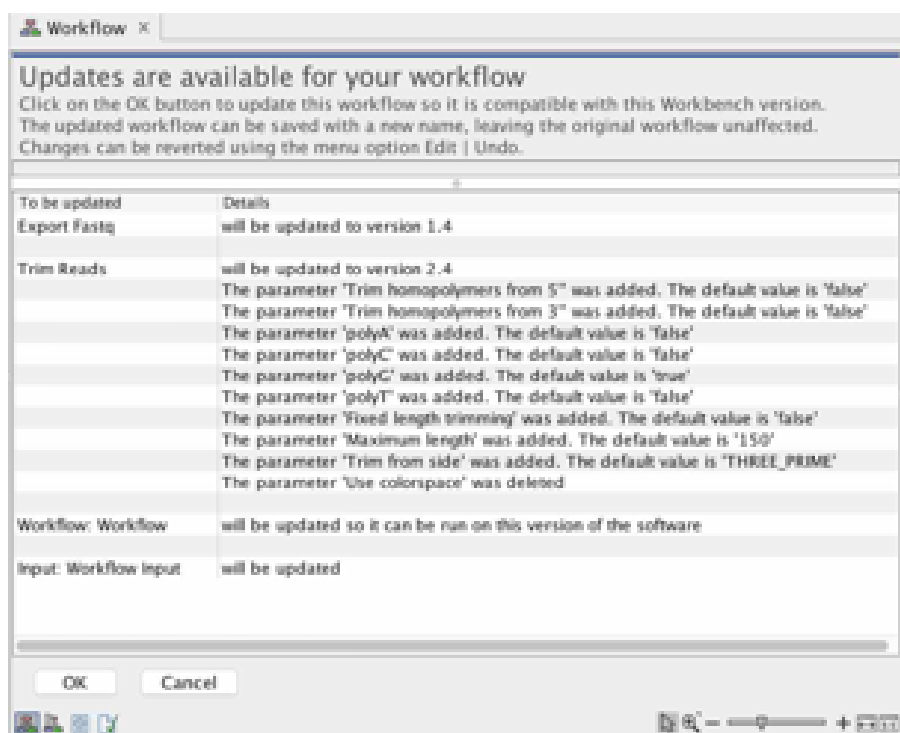


Figure 14.113: The workflow update editor lists tools and parameters that will be updated.

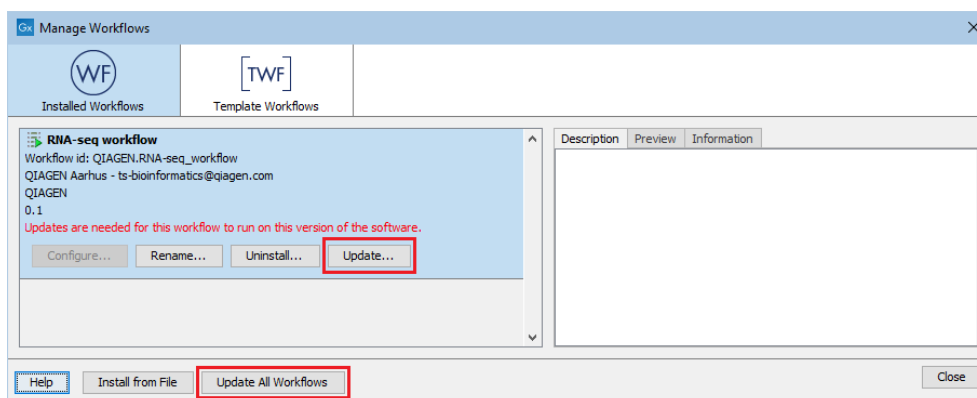


Figure 14.114: A message in red text indicates a workflow needs to be updated. The Update button can be used to update an individual workflow. Alternatively, update all workflows that need updating by clicking on the Update All Workflows button.

To update a workflow you must have permission to write to the area the workflow is stored in. Usually, you will not need special permissions to do this for workflows you installed. However, to update template workflows, distributed via plugins, the *CLC Workbench* will usually need to be run as an administrative user.

When one or more installed workflows or template workflows needs to be updated, you are informed when you start up the *CLC Workbench*. A dialog listing these workflows is presented, prompting you to open the Workflow Manager (figure 14.115).

Updating installed workflows when using software in a higher major version line

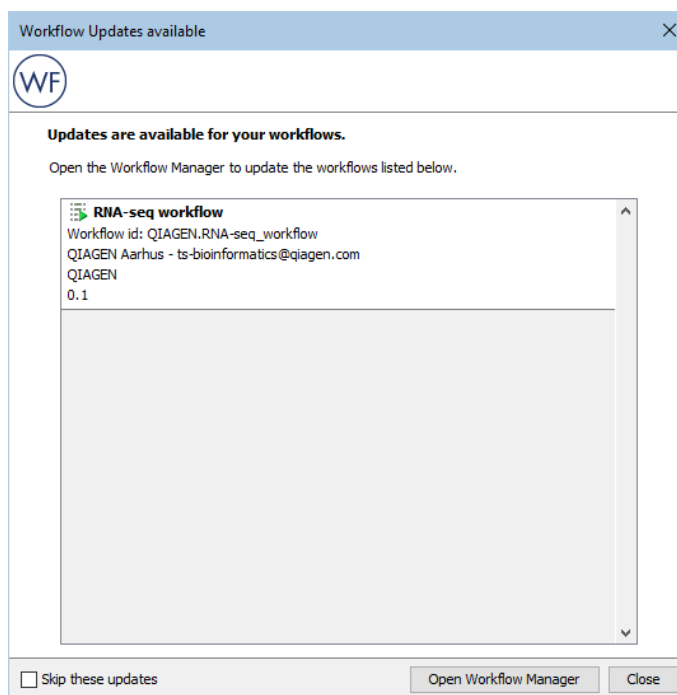


Figure 14.115: A dialog reporting that an installed workflow needs to be updated to be used on this version of the Workbench.

To update an installed workflow after upgrading to software in a higher major version line, you need a copy of the older Workbench version, which the installed workflow can be run on, as well as the latest version of the Workbench.

To start, open a copy of the installed workflow in a version of the Workbench it can be run on. To do this, right-click on the workflow's name in the **Installed Workflows** folder under the **Workflows** tab in the Toolbox panel in the bottom left side of the Workbench, and choose the option "Open Copy of Workflow" from the menu that appears (figure 14.116).

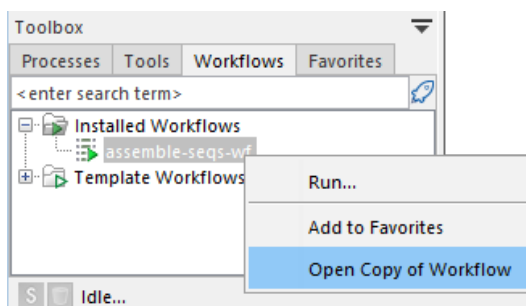


Figure 14.116: Open a copy of an installed workflow by right-clicking on its name in the Workflows tab in the Toolbox and choosing the "Open Copy of Workflow" option from the menu.

Save the copy of the workflow. One way to do this is to drag and drop the tab to the location of your choice in the Navigation Area.

Close the older Workbench and open the new Workbench version. In the new version, open the workflow you just saved. Click on the **OK** button if you are prompted to update the workflow.

After checking that the workflow has been updated correctly, including that any reference data is configured as expected, save the updated workflow. Finally, click the **Installation** button to install the workflow, if desired.

If the above process does not work when upgrading directly from a much older Workbench version, it may be necessary to upgrade step-wise by upgrading the workflow in sequentially higher major versions of the Workbench.

14.6.2 Workflow installation

Installing workflows allows you to provide and use a controlled version of the workflow. Only parameters left unlocked in such workflows can be configured when launching. Installed workflows are available for use from the Installed Workflows subfolder under the Workflows menu.

To create a workflow installer, or to directly install the workflow, click on the **Installation...** button at the bottom of the workflow editor (keyboard shortcut Shift + Alt + I).

At the end of this process, you will have the option to install the workflow directly to your Workbench or to create an installer file, which can be used to install the workflow on any compatible *CLC Workbench* or *CLC Server*. If you are logged into a *CLC Server* as a user with appropriate permissions, you will also have the option to install the workflow directly on the *CLC Server*.

Information about the workflow

In the Create Installer dialog, you are prompted for information about the workflow (figure 14.117). After the workflow is installed, this information will be visible in the Workflow Manager, as described in section 14.6.

The information requested when creating a workflow installer is described below. The workflow name and organization information are required- Other fields are optional.

Author name The name of the workflow author.

Author email The email address of workflow author.

Author homepage A URL.

Organization (Required) The organization name. This is used as part of the workflow id (section 14.6).

Workflow name (Required) The name of the workflow, as it should appear under the Workflows menu after installation. Changing this does not affect the name of the original workflow (as appears in your Navigation Area). This name is also used as part of the workflow id (section 14.6).

ID The workflow id. This is created using information provided in other fields. It cannot be directly edited.

Workflow icon An icon to use for this workflow in the Workflows menu once the workflow is installed. Icons use a 16 x 16 pixel gif or png file. If your icon file is larger, it will automatically be resized to fit 16 x 16 pixels.

Workflow version A major and minor version for this workflow. This version will be visible via the Workflow Manager after the workflow is installed, and will be included in the history of elements generated using this workflow. The version of the workflow open in the Workflow Editor, from which this installer is being created, will also be updated to the version specified here.

Workflow description A textual description of the workflow. After installation, this is shown in the Description tab of the Workflow Manager (section 14.6) and is also shown in a tooltip when the cursor is hovered over the installed workflow in the Workflows menu in the Toolbox panel at the bottom, left of the Workbench.

The screenshot shows a 'Create Installer' window with a 'Workflow information' tab. On the left, a sidebar lists '1. Workflow information' and '2. Install location'. The main area contains the following fields:

- Workflow data:**
 - Author name: Joe Bloggs
 - Author email: jbloggs@mycompany.com
 - Author homepage: http://jbloggs.com
 - Organization: Bloggs Incorporated
 - Workflow name: Map Sequences and Create Report
 - ID: Bloggs_Incorporated.Map_Sequences_and_Create_Report
- Workflow icon:** A button with a folder icon.
- Workflow version:** Two spinners showing '1' and '0'.
- Workflow description:** A text area with the text: 'This workflow maps sequencing reads to a reference sequence and creates a report.'

At the bottom are buttons for 'Help', 'Reset', 'Previous', 'Next', 'Finish', and 'Cancel'.

Figure 14.117: Provide information about the workflow that you are creating an installer for.

Choosing how reference data inputs should be handled

If data elements were preconfigured as inputs to any workflow element, then the next step when creating the workflow installer is to specify how those elements and associated data should be handled (figure 14.118).

In workflows where no data elements were preconfigured as inputs, this step is not shown.

The data handling options are:

Ignore The data elements selected as inputs in the original workflow are not included in the workflow installer.

Input options where *Ignore* is selected should generally be kept unlocked. If locked, the data element referred to must present in the exact relative location used on your system when creating the installer. If the option is locked, and the selected data element is not present in the expected location, an error message is shown in the Workflow Manager when the workflow is installed. It will not be possible to run that workflow until the relevant data element is present in the expected location.

If you have configured both a data element *and* a workflow role, then *Ignore* will usually be the best choice. In this case, when the workflow is installed, if the data element is part

of one or more QIAGEN or Custom Reference Sets and has been assigned the specified workflow role, those reference sets will be the ones shown by default to select from when launching the workflow.

See section 14.2.3 for information on workflow roles.

See section 14.2.2 for information on locking and unlocking parameters.

Bundle The data elements selected as inputs in the original workflow are included in the workflow installer. This is a good choice when sharing the workflow with others who may not have the relevant reference data on their system.

When installing a workflow with bundled data on a *CLC Workbench*, you are prompted where to save the bundled data elements. If the workflow is on a *CLC Server*, the data elements are saved automatically, as described in the *CLC Server* manual at:

http://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=Installing_configuring_workflows.html

Bundling is intended for use with **small** reference data elements. With larger elements, the workflow installer can quickly become very big. In addition, each time such a workflow is installed, a copy of the bundled data is included, even if the relevant data element is already present on the system elsewhere.

Reference This option is only presented if a data element located in a *CLC_References* location has been selected as input to the workflow. This is a good choice for those inputs where no workflow role is configured and where the workflow will be installed to a system with a *CLC_References* location where the relevant data is available, for example, a shared *CLC_References* location on a network system or a *CLC_References* file system location on a *CLC Server*. When selected, the workflow installer includes the relative location of the relevant data element, rather than including the reference data itself.

When working with large data elements, the Ignore or Reference option are recommended. To use the Reference option, the data elements must be present in a *CLC_References* area, as described below. (See section 11.4.1 for how to transfer data to a *CLC_References* folder.) If using the Ignore option, the relevant data elements should be shared using other means. For example, export the data and share this separately. The benefit with this, over bundling, is that the data can be shared once, rather than with every workflow installer that refers to it.

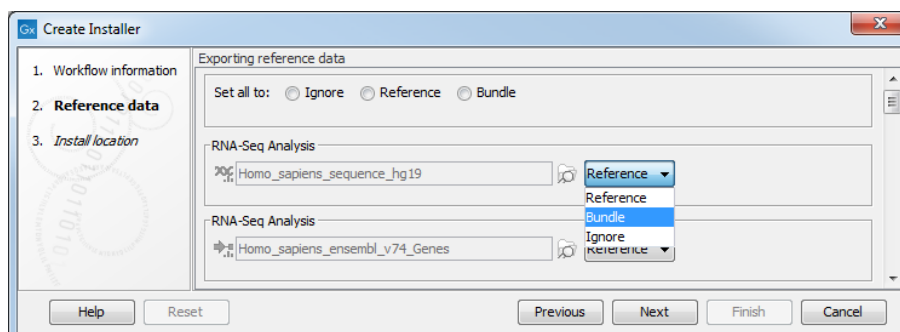


Figure 14.118: Bundling data with the workflow installer.

Installation options

The final step asks you to indicate whether to install the workflow directly to your Workbench or to create an installer file, which can be used to install the workflow on any compatible *CLC Genomics Workbench* or *CLC Server* (figure 14.119). If you are logged into a *CLC Server* as a user with appropriate permissions, you will also have the option to install the workflow directly on the *CLC Server*.

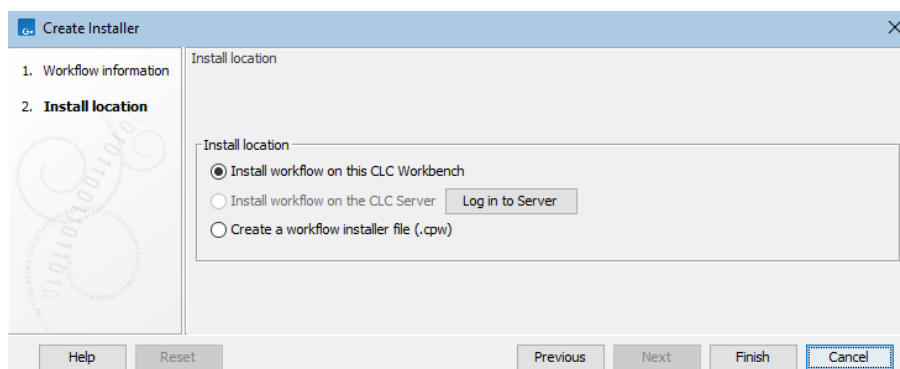


Figure 14.119: Select whether the workflow should be installed to your *CLC Workbench* or an installer file (.cpw) should be created. Installation to the *CLC Server* is only available if you are logged in as a user with permission to administer workflows on the *CLC Server*.

Workflows installed on a *CLC Workbench* cannot be updated in place. To update such a workflow, make a copy, modify it, and then create a new installer. We recommend that you increase the version number when creating the installer to help track your changes.

When you then install the updated copy of the workflow, a dialog will pop up with the message "Workflow is already installed" (figure 14.120). You have the option to force the installation. This will uninstall the existing workflow and install the modified version of the workflow. If you choose to do this, the configuration of the original workflow will be gone.

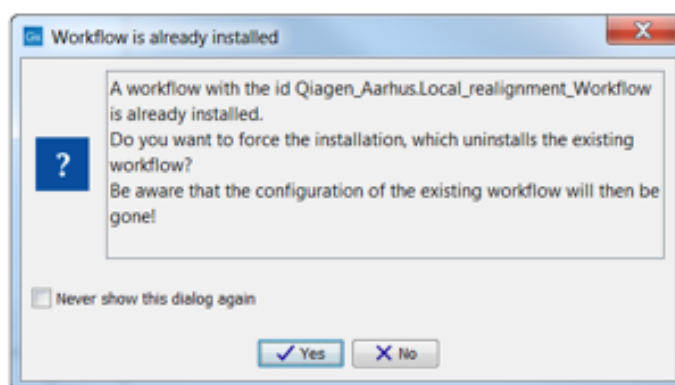


Figure 14.120: Select whether you wish to force the installation of the workflow or keep the original workflow.

14.6.3 Using workflow installation files

Workflow installation files (.cpw files) can be installed on a *CLC Workbench* using the Workflow Manager:

Utilities | Manage Workflows (⚙️)

or press the **Manage Workflows** button (⚙️) in the toolbar and then select "Manage Workflows..." (⚙️).

To install a workflow, click on Install from File and select a .cpw file. If the workflow has bundled data, you will be prompted for a location for that data. Once installed, the workflow will appear under the Installed Workflows tab (figure 14.121).

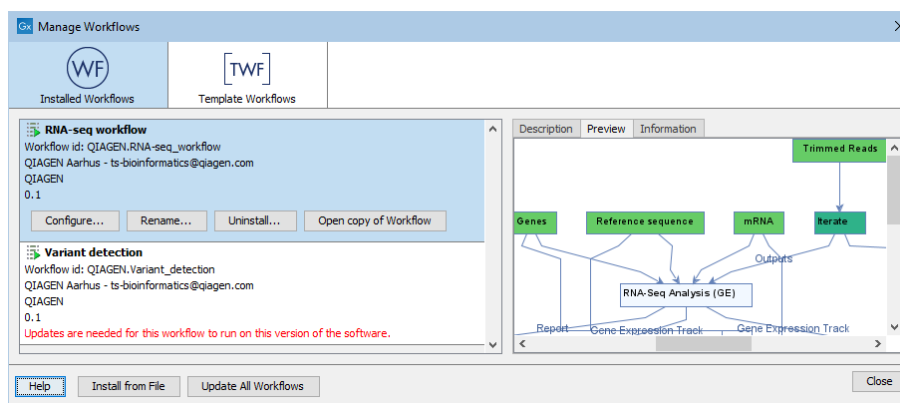


Figure 14.121: Workflows available in the workflow manager. The alert on the "Variant detection" workflow means that this workflow needs to be updated.

See section 14.6.2 for information about options for handling reference data inputs.

If the *References* option has been used, then the relevant data elements are expected to be present in the *CLC_References* location. If workflow roles have been used, then data with the relevant role is expected to be present in at least one Reference Set in the *CLC_References* location. In this case though, when launching the workflow, you will have the option to download relevant data sets if they are not already available.

Information about installing workflows on a *CLC Server* is provided in the *CLC Server* manual at:

http://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=Installing_configuring_workflows.html

14.7 QIAseq Panel Analysis Assistant

The QIAseq Panel Analysis Assistant provides an easy entrance point for working with data generated with QIAseq panels and kits. Using the QIAseq Panel Analysis Assistant, information about the panels and kits can be accessed, available analyses can be viewed and run.

Most analyses offered via the QIAseq Panel Analysis Assistant are based on template workflows, which are available under the Workflows menu. Analyses launched using the QIAseq Panel Analysis Assistant have the appropriate reference data preselected, see section 14.7.2. Additionally, some parameters are different to the template workflow, to account for the panel/kit design, see section 14.7.4 for details.

Validation of results should be carried out. See section 14.7.5 if adjustments to parameters are needed.

To start the QIAseq Panel Analysis Assistant, go to:

Workflows | Template Workflows | QIAseq Panel Analysis Assistant (🔍)

Adding analyses

Analyses of QIAseq data are delivered by plugins, see section 14.7.1 for details.

Viewing analyses

Starting the QIAseq Panel Analysis Assistant with at least one relevant plugin installed, opens a wizard listing different panel/kit categories on the left side, and analyses of panels/kits in the selected category on the right side (figure 14.122).

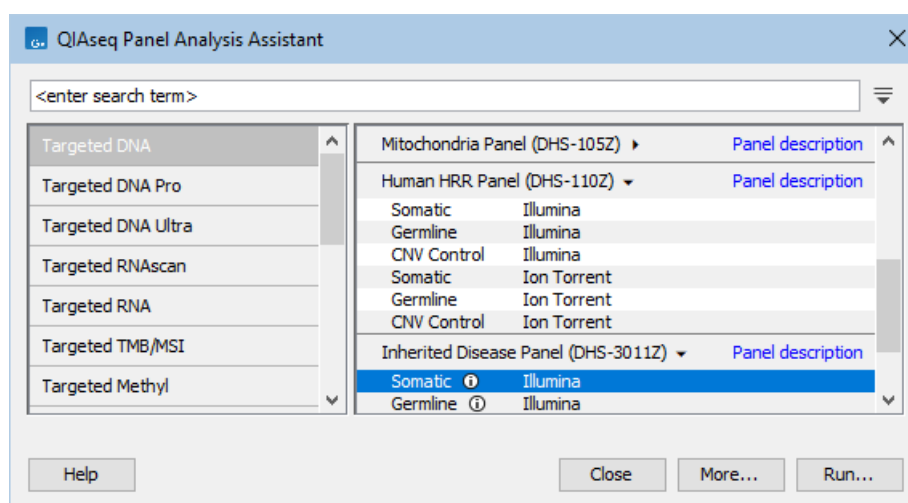


Figure 14.122: The QIAseq Panel Analysis Assistant when the **Biomedical Genomics Analysis** plugin is installed. Multiple analyses are available for the DHS-110Z and DHS-3011Z panels. The analyses for DHS-3011Z are marked with an info icon because they use different parameters relative to the underlying template workflow. The "Panel description" links to more information about the panel.

An analysis can be:

- A pre-configured template workflow, see section 14.5. Some analyses are marked with (🔍) (figure 14.122), see section 14.7.4 for details.
- A tool available from the Tools menu, with certain options pre-configured.
- A tool only available from within the QIAseq Panel Analysis Assistant.

The search field at the top (figure 14.122) can be used to search for terms in:

- Panel/kit name, for example *Mitochondria Panel*.
- Panel/kit identifier, for example *DHS-105Z*.
- Panel/kit analysis, for example *Somatic Illumina*.
- Template workflow, for example *Identify QIAseq DNA Somatic Variants*.

- Tool, for example *Call Methylation Levels*.

For exact searching, use

- Single quotes: matches terms containing exactly the search term.
- Double quotes: matches terms that are exactly the same as the search term.

Double-click on a search result to open it.

Once an analysis has been selected, it can be started using **Run**, see section 14.7.3 for details. Additional actions for the selected analysis are available under **More**:

- **View Reference Data.** Shows information about the reference data used, if relevant, see section 14.7.2 for details.
- **Open Copy of Workflow.** Opens a copy of the pre-configured template workflow, see section 14.7.5.
- **Find in Toolbox.** Navigates to the template workflow/tool in the Toolbox area. This requires that the Toolbox area in the bottom, left side of the *CLC Workbench* is visible. See section 2.3 for details.
- **Analysis Help.** Shows the relevant manual page.

14.7.1 Adding QIAseq analyses

Analyses of QIAseq data are delivered by the following plugins:

- *Biomedical Genomics Analysis*
- *CLC LightSpeed Module*
- *CLC Microbial Genomics Module*

When no relevant plugin is installed, starting the QIAseq Panel Analysis Assistant opens a wizard listing the relevant plugins (figure 14.123). The **Go to Plugins** button opens the "Manage Plugins" wizard with the corresponding plugin selected. See section 1.5 for information on installing plugins.

Once a plugin is installed, the analyses are added to the QIAseq Panel Analysis Assistant and the corresponding plugin is marked as being installed (figure 14.124).

14.7.2 Reference data for QIAseq analyses

Most analyses in the QIAseq Panel Analysis Assistant use QIAGEN Reference Data Sets for the corresponding panel/kit, see section 11.2.

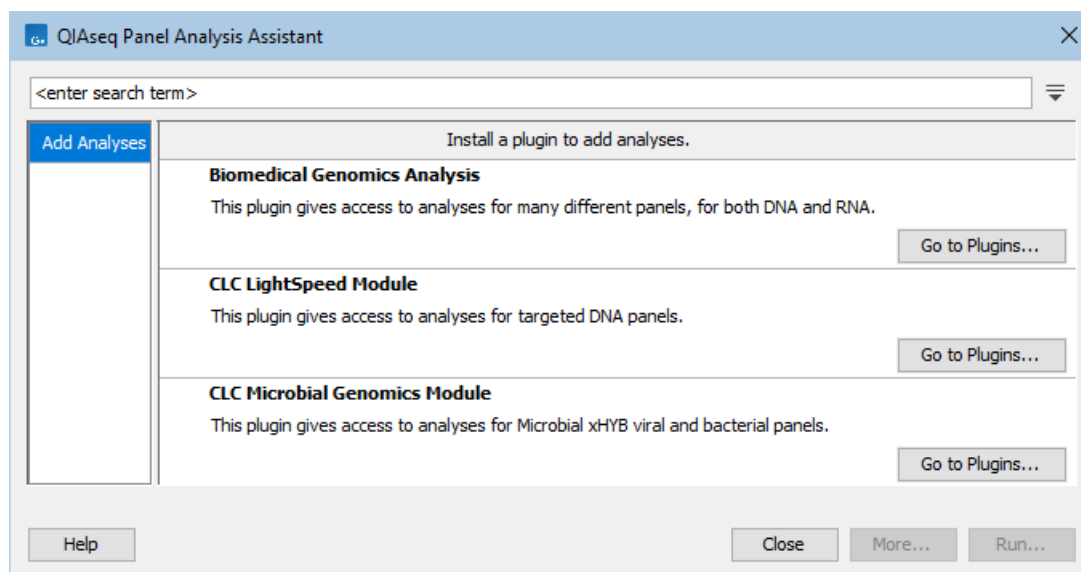


Figure 14.123: The QIAseq Panel Analysis Assistant when no relevant plugins are installed.

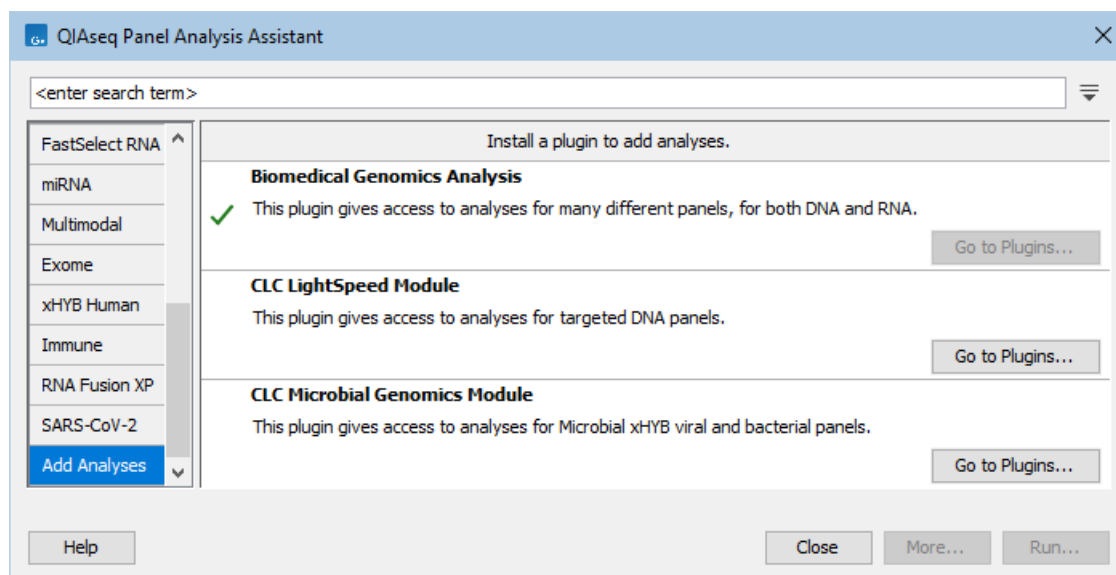



Figure 14.124: The QIAseq Panel Analysis Assistant when the **Biomedical Genomics Analysis** plugin is installed.

View reference data

To view the reference data used by an analysis, click **View Reference Data** under **More** in the QIAseq Panel Analysis Assistant wizard. The "View Reference Data" wizard opens, listing the Reference Data Elements used (figure 14.125). Clicking on the name of the Reference Data Set, at the top of the wizard (figure 14.125), opens the Reference Data Manager with the relevant Reference Data Set selected.

The Reference Data Elements that have previously been downloaded can be shown in the Navigation Area by clicking on the  button.

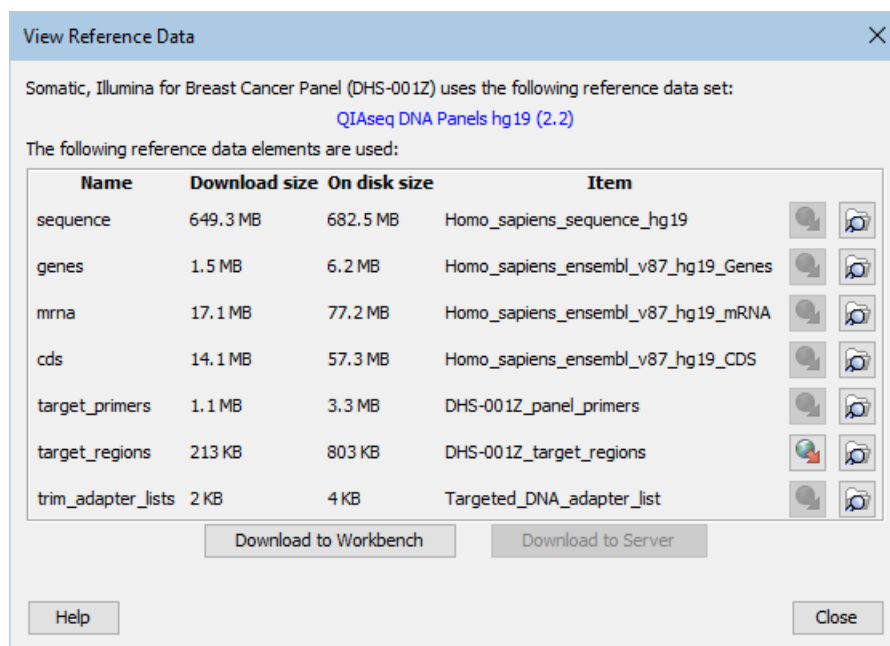


Figure 14.125: View reference data. The "Somatic, Illumina" analysis for the DHS-001Z panel uses the "QIaseq DNA Panels hg19" Reference Data Set. All but the target regions have been previously downloaded to the **CLC Workbench**. The "Download to Server" button is present because the **CLC Workbench** is logged into a **CLC Server**. However, all Reference Data Elements are already downloaded on the **CLC Server** and the button is disabled.

Download reference data prior to execution

All missing Reference Data Elements can be downloaded by clicking on the **Download to Workbench** button in the "View Reference Data" wizard (figure 14.125). Alternatively, click on (🌐) to download individual elements. If the *CLC Workbench* is logged into a *CLC Server*, there is also a **Download to Server** button.

Download reference data during execution

All missing Reference Data Elements can be downloaded during the execution of an analysis. After starting the analysis using the **Run** button, the "Acquire reference data" wizard step allows for downloading the missing Reference Data Elements by using the **Download to Workbench** button (figure 14.126). If the *CLC Workbench* is logged into a *CLC Server*, there is also a **Download to Server** button.

When executing the analysis on a *CLC Server*, the reference data must be available on the *CLC Server*. When executing the analysis on the *CLC Workbench* while logged into a *CLC Server*, reference data on the *CLC Server* can be used, but for performance reasons it is recommended to copy the reference data to the *CLC Workbench* (figure 14.126).

Use different reference data

To use a different Reference Data Set, see section 14.7.5.

A different Reference Data Set is usually needed for analyzing data generated using QIaseq

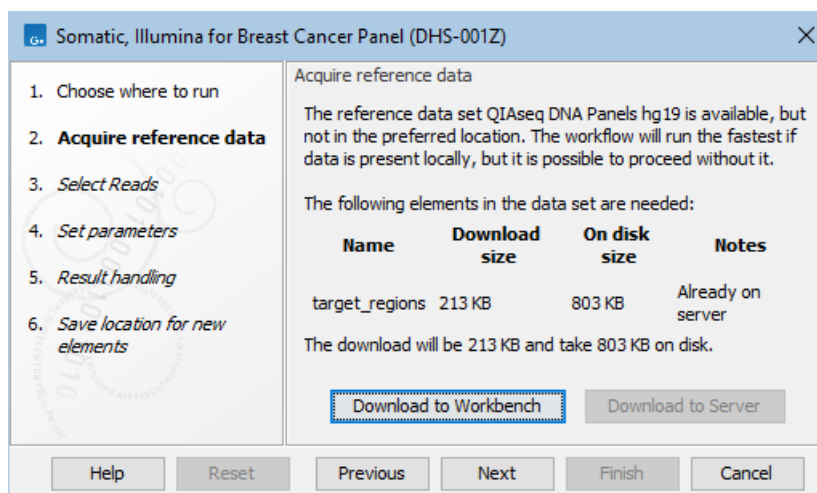


Figure 14.126: Download missing Reference Data Elements during execution. The "Download to Server" button is present because the **CLC Workbench** is logged into a **CLC Server**. However, all Reference Data Elements are already downloaded on the **CLC Server** and the button is disabled.

custom panels/kits, see section 14.7.6.

See section 11.3 for information about Reference Data Sets.

14.7.3 Running a QIAseq analysis

After starting an analysis using the **Run** button, a wizard shows, containing multiple steps:

- **Choose where to run.** If the *CLC Workbench* is logged into a *CLC Server*, the first step is to select where to run the analysis. See section 12.1.1 for details.
- **Acquire reference data.** If the reference data required for the analysis has not been downloaded previously, it must be downloaded before proceeding. See section 14.7.2 for details.
- **Select Input(s).** Select the data from the relevant QIAseq panel/kit. Typically, this would be the raw reads, but certain analyses require as input outputs produced by other analyses. Depending on the analysis, there might be multiple steps for selecting the inputs.
- **Set parameters.** Some analyses require that a few parameters are set (figure 14.127).
- **Result handling.** Specify if results should be opened or saved. Depending on the type of the analysis, different options are available:

- For workflows, results can only be saved.

The **Preview All Parameters** button allows to review all parameters used for the workflow execution. Clicking the button opens a new wizard similar to the Configuration Editor view, see section 14.1.7.

Checking **Create workflow result metadata** creates a Workflow Result Metadata table, see section 14.3.1 for details.

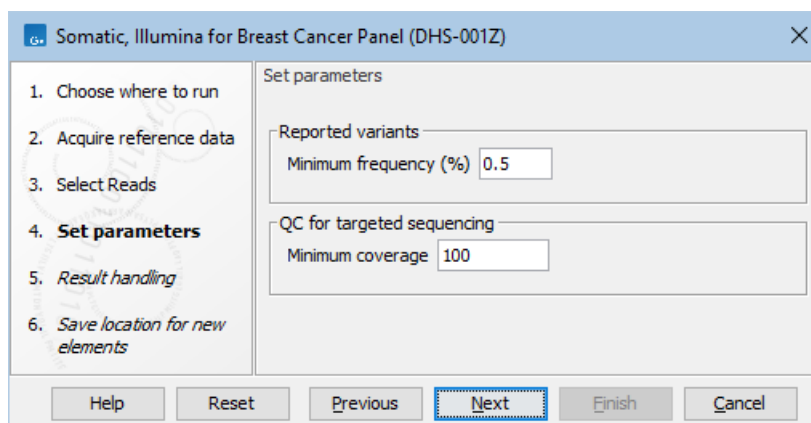


Figure 14.127: The "Somatic, Illumina" analysis for the DHS-001Z panel allows adjusting the minimum variant frequency and read coverage.

- For tools, there might be additional tool specific options.
- **Save location for new elements.** Specify where to save the results when selecting "Save" in the previous wizard step. See section 12.2 for details.

Batching

Analyses can be run individually or in batches by checking the **Batch** checkbox on the **Select Input(s)** wizard step. See section 12.3 for details.

When the analysis is a workflow, the batch units in the QIAseq Panel Analysis Assistant are always defined by the organization of input data, see section 14.3.2 for details. To define batch units based on metadata, see section 14.7.5.

14.7.4 Non-standard parameters in QIAseq analyses

Most analyses offered via the QIAseq Panel Analysis Assistant are based on template workflows, which are available under the Workflows menu. Analyses launched using the QIAseq Panel Analysis Assistant have the appropriate reference data preselected, see section 14.7.2. Additionally, some parameters are different to the template workflow, to account for the panel/kit design. Such analyses are marked with (ⓘ).

To view which parameters are set differently, click on (ⓘ). The "Changes to Workflow" wizard opens (figure 14.128), listing:

- **Element.** The element, typically a tool, for which a parameter is set to a different value.
- **Parameter.** The parameter that is set to a different value.
- **Value.** The value of the parameter.

14.7.5 Configuring QIAseq analyses

Most of the analyses provided in the QIAseq Panel Analysis Assistant can be run in different ways:

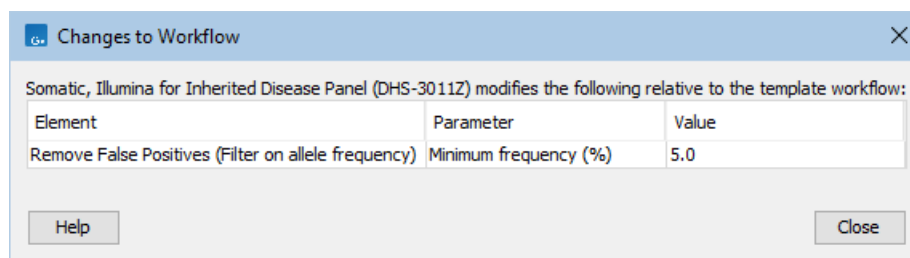


Figure 14.128: The "Somatic, Illumina" analysis for the DHS-3011Z panel removes false positive variants using a minimum frequency of 5%.

From the QIAseq Panel Analysis Assistant.

- How to do this: Click **Run** in the QIAseq Panel Analysis Assistant wizard. See section 14.7.3 for details.
- Considerations: Launching analyses using the QIAseq Panel Analysis Assistant is simple because most parameters are preconfigured.

From the Tools or Workflows menu.

- How to do this: Click **Find in Toolbox** under **More** in the QIAseq Panel Analysis Assistant wizard.
- Considerations: More parameters are available for configuration than when launching using the QIAseq Panel Analysis Assistant.

From a copy.

- How to do this: Click **Open Copy of Workflow** under **More** in the QIAseq Panel Analysis Assistant wizard.
- Considerations:
 - Only possible for analyses that use template workflows.
 - All parameters are available for configuration.
 - Needed if the analysis in the QIAseq Panel Analysis Assistant applies different parameters compared to the template workflow. Such analyses are marked using (ⓘ), see section 14.7.4 for details.
 - The workflow copy can further be adapted to better suit a specific panel/kit.

Starting the analysis using a template workflow from under the Workflows menu, or from a workflow copy, also allows you to:

- Use a different Reference Data Set.
- Define batch units based on metadata.

Opening and configuring a workflow copy

To open a copy of the pre-configured workflow used by an analysis, click **Open Copy of Workflow** under **More** in the QIAseq Panel Analysis Assistant wizard. The workflow copy is configured with the corresponding reference data (section 14.7.2) and has the same parameters as those used when running the workflow directly from the QIAseq Panel Analysis Assistant.

When opening a copy of the workflow,

- If all needed Reference Data Elements have been previously downloaded to the *CLC Workbench*, or, if relevant, to the *CLC Server* that the *CLC Workbench* is logged into, the copy opens in the background.
- If there are missing Reference Data Elements, a "Reference data" wizard offers to download them (figure 14.129). The download can be skipped by clicking **Finish**.

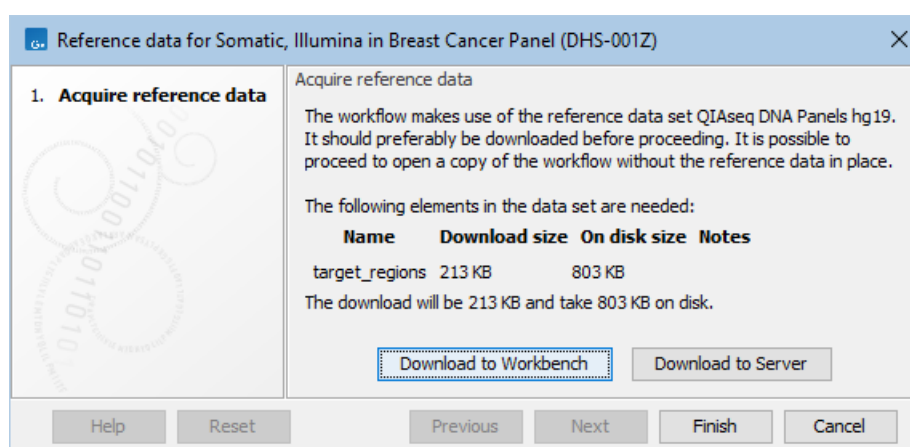


Figure 14.129: Download missing Reference Data Elements when opening a workflow copy. The "Download to Server" button is present because the **CLC Workbench** is logged into a **CLC Server** and the Reference Data Element is also missing from the **CLC Server**.

When there are missing Reference Data Elements and the download is skipped in the "Reference data" wizard (figure 14.129), the workflow fails to validate (see section 14.1.4) due to the missing data (figure 14.130). To run the workflow, either download the missing Reference Data Element or configure the workflow to use different data. See `internalrefsec:configuringinputelementssectionConfiguring Workflow Input elements` for information on how to design workflows using reference data.

See section 14.1 for information about editing workflows, section 14.2.2 and section 14.1.7 for information about changing the parameters of the tools. Once the workflow is fully configured and saved to the Navigation Area, it can be installed, see section 14.6.2 for details.

14.7.6 QIAseq custom panels

The QIAseq Panel Analysis Assistant only supports standard QIAseq panels/kits, for which reference data is already available in the *CLC Genomics Workbench*, see section 14.7.2.

QIAseq custom panels are usually built from standard QIAseq panels and the analyses available for the standard panel are typically suitable for analyzing the data generated using the custom

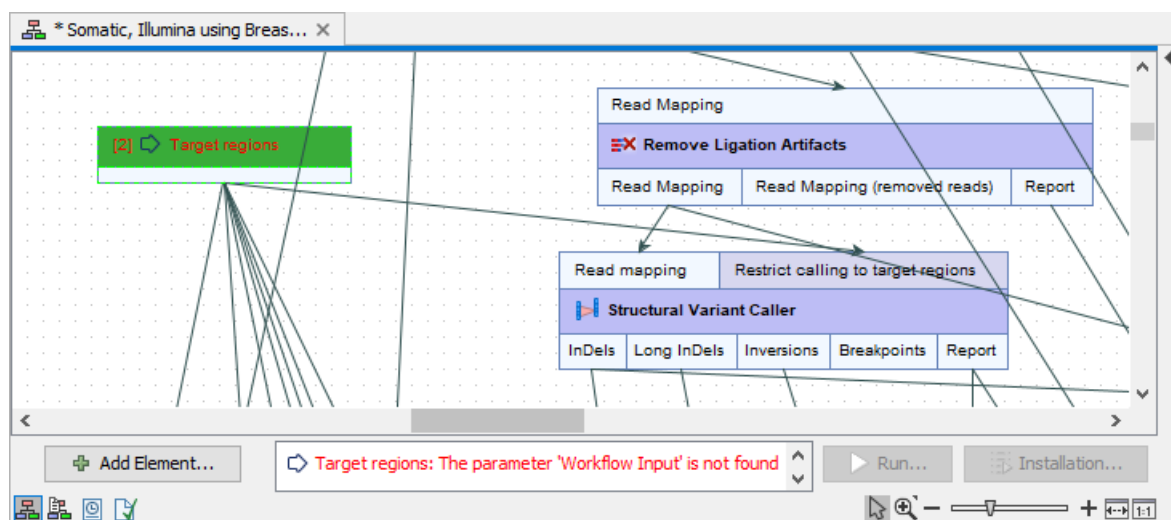


Figure 14.130: A copy of the template workflow for the "Somatic, Illumina" analysis for the DHS-001Z panel opened from the QIAseq Panel Analysis Assistant. The step to download missing reference data was skipped and thus the workflow does not validate due to the missing target regions.

panel.

For custom panels, specific files are made available upon the purchase. These need to be imported into the *CLC Genomics Workbench* and used instead of the standard Reference Data Elements. Such a file could for example contain target regions specified in BED files, see section 7.2 for how to import them.

Using an identified suitable analysis from the QIAseq Panel Analysis Assistant, data generated using QIAseq custom panels/kits can be analyzed by:

- **Using a Custom Set** containing the imported elements.
- **Using a workflow copy** configured to use the imported elements.

Using a Custom Set

To create a Custom Set:

- Select the suitable analysis from the QIAseq Panel Analysis Assistant.
- Use **View Reference Data** under **More** to see the reference data for the relevant analysis.
- Click on the name of the Reference Data Set. This opens the Reference Data Manager with the relevant Reference Data Set selected.
- Click on the **Create Custom Set** button to create a custom set based on the Reference Data Set.
- Update the custom set to use the imported elements as relevant.

See section 11.3 for details on creating custom Reference Data Sets.

Once the Custom Set is created, it can be used when configuring analyses, see section 14.7.5.

Using a workflow copy

See section 14.7.5 for details on how to open a workflow copy.

A workflow copy can be updated to use other Reference Data Elements. See section 14.2.3 for details.

To quickly find the input elements that need to be changed, use the **Find** text field in the Side Panel of the workflow (figure 14.131), see section 14.1.9 for details.

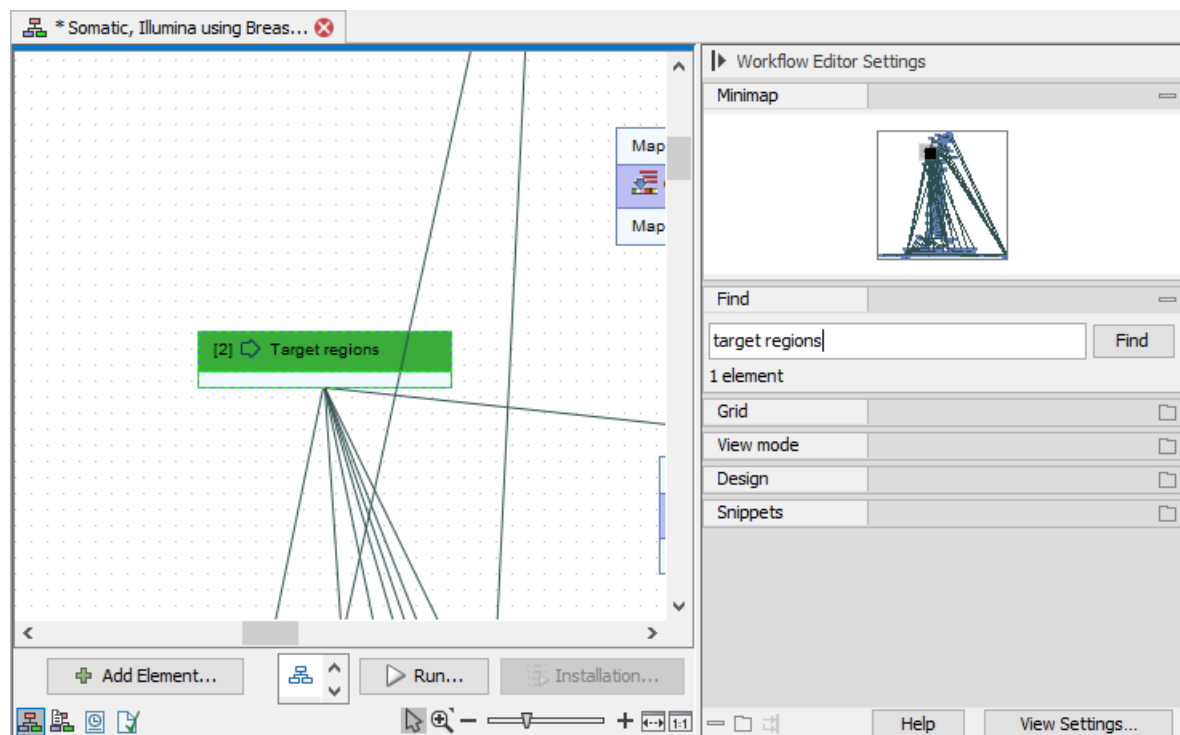


Figure 14.131: A copy of the template workflow for the "Somatic, Illumina" analysis for the DHS-001Z panel opened from the QIAseq Panel Analysis Assistant. The target regions input is quickly identified by searching for "target regions".

Part III

Basic sequence analysis

Chapter 15

Viewing and editing sequences

Contents

15.1 Sequence Lists	381
15.1.1 Creating sequence lists	381
15.1.2 Graphical view of sequence lists	382
15.1.3 Table view of sequence lists	384
15.1.4 Annotation Table view of sequence lists	386
15.1.5 Working with paired sequences in lists	386
15.2 View sequences	387
15.2.1 Sequence settings in Side Panel	387
15.2.2 Selecting parts of the sequence	394
15.2.3 Editing the sequence	395
15.2.4 Sequence region types	395
15.2.5 Circular DNA	395
15.3 Working with annotations	399
15.3.1 Viewing annotations	400
15.3.2 Adding annotations	404
15.3.3 Editing annotations	406
15.3.4 Export annotations to a gff3 format file	407
15.3.5 Removing annotations	408
15.4 Element information	408
15.5 View as text	409

Sequence information is stored in sequence elements or sequence lists. This chapter describes basic functionality for creating and working with these types of elements. Most functionality available for sequence elements is also available for sequence lists.

When you import multiple sequences, they are generally put into a sequence list, and this is the element type in use for most types of work. See chapter 7.

15.1 Sequence Lists

Sequence List elements contain one or more nucleotide sequences *or* one or more peptide sequences. They are used as input to many tools, and are generated as output by many tools. Sequence lists can contain single end sequences or paired end sequences, but not a mixture of both. Handling paired data is described at the end of this section.

When open in the viewing area, there are 5 icons at the bottom providing access to different views of the data. Side panel settings on views allow customization of what is shown. These settings can be saved, as described in section 4.6.

In figure 15.1, the graphical view and Table view of a sequence list are open in linked views (section 2.1). With linked views, clicking in one view can lead to updates in the other view. For example, selecting a row in the table view will cause the graphical view to update so the focus is on that sequence.

Much functionality is contained in right-click menus, with options differing depending on which view is open and where you click in it. Some actions are unique to a particular view, while others are available from more than one view. For example, sequence attributes can only be edited in the Table view but sequences can be added from the graphical view and Table view.




Figure 15.1: Two views of a sequence list are open in linked views, graphical at the top, and tabular at the bottom. Each view can be customized individually using settings in its side panel on the right.

15.1.1 Creating sequence lists

Sequence lists are created in various ways, including:

- When sequences are imported. See chapter 7.
- As outputs of analysis tools.
- When sequences are downloaded, for example using tools under the Download menu.
- Using **Create Sequence List**, found at:

Tools | Utility Tools () | Sequence Lists () | Create Sequence List ()

or launched via a button at the bottom sequence lists in Table  view.

This tool can also be accessed at File | New | Sequence List.




15.1.2 Graphical view of sequence lists

In the graphical view of sequence lists, you can see sequences at the residue level, or zoomed right out to get an overview of the full sequence. Options in the Side Panel on the right allow detailed customization, including aspects like viewing trace data and quality scores, where available (figure 15.1).

Much of the functionality available when working with sequence lists is also available when working with individual sequence elements. Shared functionality is described in section 15.2. This section describes options relevant only to sequence lists.

Menus available when you right-click give access to much functionality for manipulating sequence lists.

Right-click on a blank area to access the following options (figure 15.2):

- **Extract Sequences** Extracts all sequences from the sequence list. If your aim is to extract a subset of the sequences, this can be done from the Table  view (see section 15.1.3) or using Split Sequence List  (see section 37.12).
- **Add Sequences** Add sequences from sequence elements or sequence lists to this sequence list.
- **Delete All Annotations from All Sequences** Deleting all annotations on all sequences can be done with this option for sequence lists with 1000 or fewer sequences. In other cases, or for more control over the annotations to delete, use the Annotation Table  view, described further below.

Right-click on a sequence or sequence name to access options (figure 15.3) to:

- **Rename, select, open, or delete** that sequence.
- **Sort the sequence list** alphabetically by sequence name, by length or by marked status. These options are only available for sequence lists with 1000 or fewer sequences.
- **Delete sequences that have been marked** This option is enabled when at least one sequence has been marked. Marking sequences is described below.

Tips for working with larger sequence lists are given later in this section.

Marking sequences:

Sequences in a list can be marked. Once marked, those sequences can be deleted, or the sequence list can be sorted based on whether sequences are marked or not. It is easy to adjust markings on many sequences using the options in the right-click menu on selection boxes (figure 15.4).

To mark sequences:

1. Check the **Show selection boxes** option in the "Sequence List Settings" section of the side panel settings on the right hand side.

This makes checkboxes visible to the right of each sequence name.

2. Click in the checkbox beside a sequence name to mark the sequence.

Clicking in the box for a marked sequence will remove the mark.

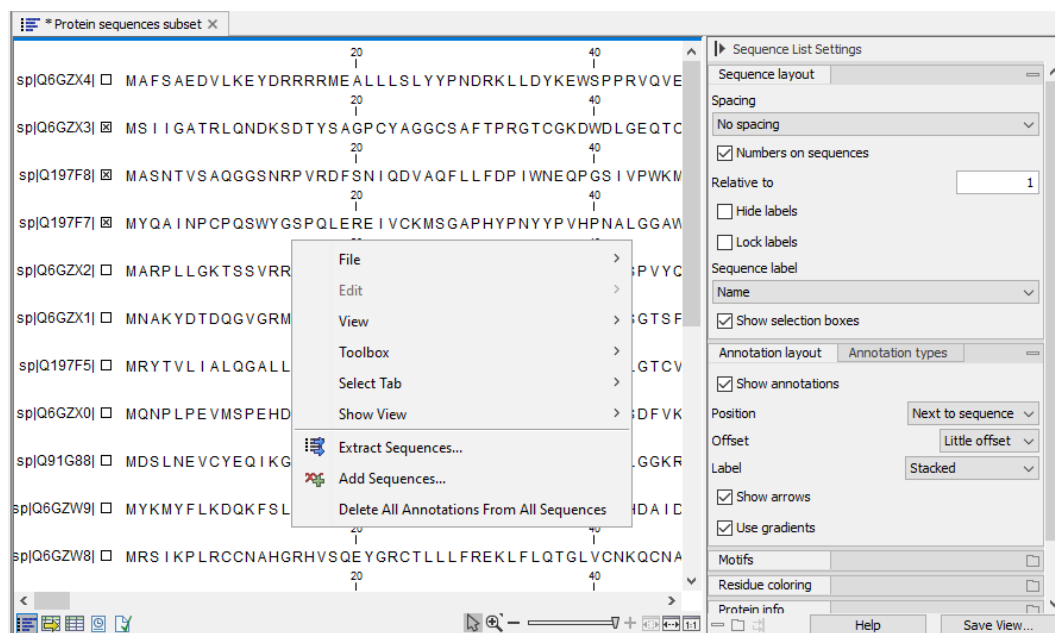


Figure 15.2: Options to extract the sequences in the list, add sequences to the list, and to delete all annotations on all sequences are available when you right-click on a blank area of the graphical view of a sequence list.

Tips when working with large sequence lists

- **Sorting long lists** can be done in Table (📊) view. For example, to sort on length, ensure the Size column is enabled in the side panel to the right of the table, and then click on the Size column header to sort the list. If a Table view is open as a linked view with the graphical view, clicking on a row in the table will highlight that sequence in the graphical view. See section 2.1 for information on linked views and section 9 for information about working with tables.
- **Deleting annotations on sequences** can be done in the Annotation Table (📋) view. Right click anywhere in this view to reveal a menu containing relevant options. To delete all annotations on the sequence list, ensure all annotation types are enabled in the side panel settings to the right.
- **To delete many sequences** from a list, you can mark the few you wish to retain, and then invert the marking by right-clicking on any selection checkbox and choosing the option **Invert All Marks** (figure 15.4).

Then right-click on any sequence or sequence name and choose the option to **Delete Marked Sequences** (figure 15.3). If the sequence list contains more than 1000 sequences, a warning will appear noting that, if you proceed, the deletion cannot be undone.

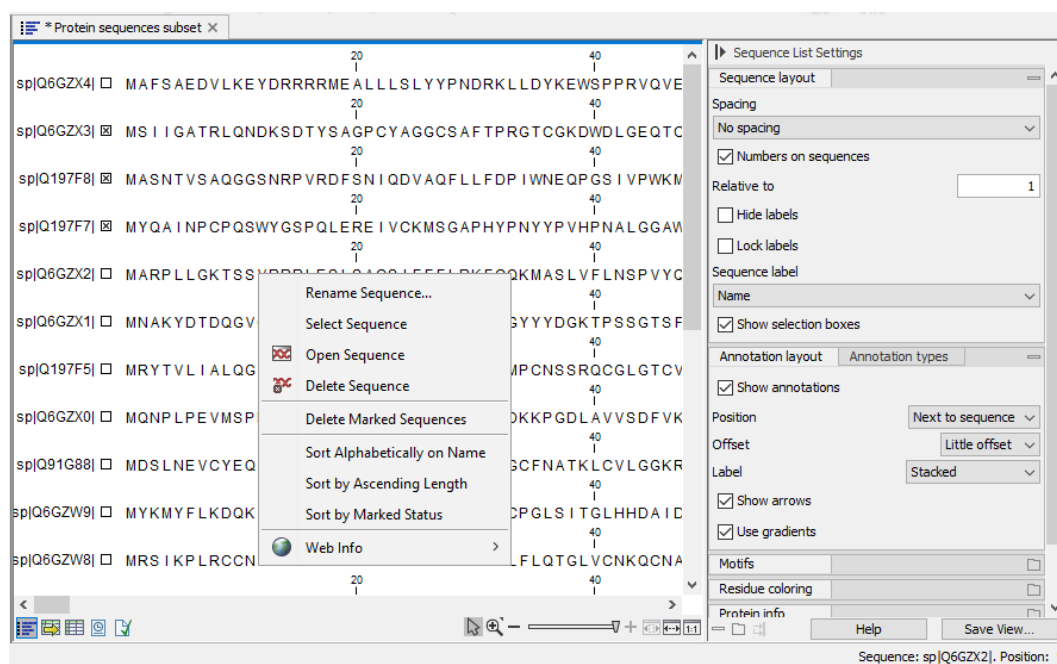


Figure 15.3: Options to rename, select, open, or delete a sequence are available when you right-click on the name or residues for a given sequence. Also in this menu are options for sorting the list and deleting marked sequences.



Figure 15.4: Which sequences are marked can be quickly adjusted using the options in the right-click menu for any selection checkbox. The Show Selection boxes option in the side panel must be enabled to see these boxes.

- **Renaming multiple sequences** in a list following the same renaming pattern can be done using the dedicated tool, **Rename Sequences in Lists**, described in section 37.15.

15.1.3 Table view of sequence lists

Sequence attributes are listed the Table (📄) view (figure 15.5). This includes information like the name, description, accession, length, etc. of each sequence, as well as any other attributes for that element, such as those added using Update Sequence Attributes in Lists (section 37.11), or custom attributes for that location (section 3.3).

The number of rows reported at the top is the number of sequences in the list (figure 15.5).

Standard functionality for working with tables applies (see section 9).

Actions on the sequence list can be taken directly, or via a right-click menu (figure 15.5):

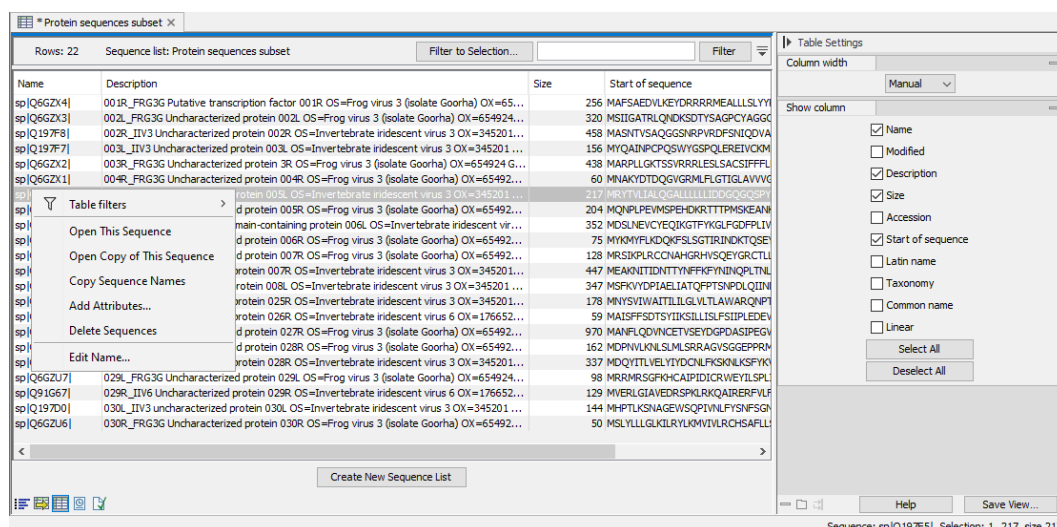


Figure 15.5: In Table view there is a row for each sequence in the sequence list. The number of rows equates to the number of sequences and is reported at the top left side. Right-click to display a menu with actions. This menu differs slightly depending on which column you click upon.

- **Add sequences** Add sequences to this list by dragging and dropping sequence elements or sequence lists from the Navigation Area into the table. Sequences can also be added from the graphical view using a right-click option, as described earlier in this section.
- **Remove sequences** by selecting the relevant rows and either:
 - Clicking on the **Delete** (X) icon in the top toolbar.
 - Right-clicking and choose the option **Delete Sequences** from the menu (figure 15.5).

Sequences can also be deleted from other views.

- **Copy sequence names** Select the relevant rows, right-click and choose **Copy Sequence Names** from the menu. This list can be used within the Workbench, for example, in table filters with the action "is in list" or "is not in list" to find these names in other elements, or they can be pasted to other programs that accept text lists, such as Excel or text editors.
- **Edit attributes** Right-click in the the cell you wish to edit, and then update the contents of that cell. For example, if you right-click on a cell in the Name column, an option called "Edit Name..." will be in the menu presented (figure 15.5).

If you select multiple rows, you will be able to edit the attribute, with the value you provide being applied to all the selected rows.

Values calculated from the sequence itself cannot be edited directly. E.g. The Size column contains the length of each sequence, and the Start of sequence column contains the first 50 residues.


- **Extract selected sequences** Specified sequences in the list can be extracted:
 - **To a new sequence list** by selecting relevant rows and clicking on the **Create New Sequence List** button. This new list must be saved if you wish to keep it.
 - **To a individual sequence elements** by selecting relevant rows and dragging them into the Navigation Area.


Adding attributes

Attributes (columns in Table view) can be added using the right-click menu option **Add Attributes**. This is good for small lists and simple changes. You are prompted for an attribute name and a single value. A new column is added to the table with the name you provide, and the value you provided is added for all of the selected rows. This option can also be used to edit contents of an existing column, if desired.

The **Update Sequence Attributes in Lists** tool supports more detailed work, including importing from external sources, such as Excel and CSV format files. See (section 37.11) for more details.

15.1.4 Annotation Table view of sequence lists

Options in the Annotation Table () view are the same as those in this view for sequences. They are described in the following sections:

- Working with the Annotation Table () view: section 15.3.1
- Adding annotations: section 15.3.2
- Editing annotations: section 15.3.3
- Exporting annotations to a gff3 format file: section 15.3.4
- Deleting annotations: section 15.3.5

15.1.5 Working with paired sequences in lists

When paired sequences are imported, the sequence list created is marked as containing paired data. This can be seen in the Element info view and can be edited within that view (section 15.4).

A sequence lists can contain single end reads or it can contain paired end reads. A given list cannot contain a mixture of these.

The most common way that paired sequence lists are created is when importing NGS sequence data, as described in section 7.3. See section 7.3.10 for more information about handling paired data after import.

To create a paired sequence list from existing sequence lists, for example by merging lists, the input lists must be marked as paired and must have the same distance settings. If the input lists do not meet these criteria, a message is shown warning that the resulting sequence list will be unpaired (figure 15.6).

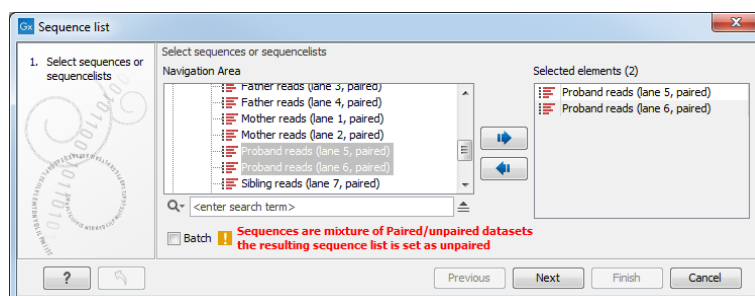


Figure 15.6: A warning appears when trying to create a new sequence list from a mixture of paired and unpaired sequence lists.

15.2 View sequences

This section describes options available when viewing and editing sequences. Many of these options also apply to sequence lists (section 15.1) and alignments (section 24.2).

15.2.1 Sequence settings in Side Panel

Settings for a view are available in the **Side Panel** at the right hand side of the view (figure 15.7). The view is instantly updated when these settings are changed. The options available for sequence views are described in the sections that follow.

Note: Side Panel settings are not automatically saved when you save the sequence. See section 4.6 for information on saving and applying view settings.

For general information about Side Panels, see section 2.1.6.

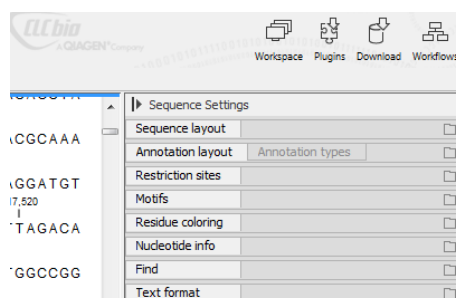


Figure 15.7: Overview of the Side Panel for a sequence. Each tab can be expanded to reveal settings that can be configured.

Sequence Layout

These preferences determine the overall layout of the sequence:

- **Spacing.** Inserts a space at a specified interval:
 - **No spacing.** The sequence is shown with no spaces.
 - **Every 10 residues.** There is a space every 10 residues, starting from the beginning of the sequence.
 - **Every 3 residues, frame 1.** There is a space every 3 residues, corresponding to the reading frame starting at the first residue.

- **Every 3 residues, frame 2.** There is a space every 3 residues, corresponding to the reading frame starting at the second residue.
- **Every 3 residues, frame 3.** There is a space every 3 residues, corresponding to the reading frame starting at the third residue.
- **Wrap sequences.** Shows the sequence on more than one line.
 - **No wrap.** The sequence is displayed on one line.
 - **Auto wrap.** Wraps the sequence to fit the width of the view, not matter if it is zoomed in our out (displays minimum 10 nucleotides on each line).
 - **Fixed wrap.** Makes it possible to specify when the sequence should be wrapped. In the text field below, you can choose the number of residues to display on each line.
- **Double stranded.** Shows both strands of a sequence (only applies to DNA sequences).
- **Numbers on sequences.** Shows residue positions along the sequence. The starting point can be changed by setting the number in the field below. If you set it to e.g. 101, the first residue will have the position of -100. This can also be done by right-clicking an annotation and choosing **Set Numbers Relative to This Annotation**.
- **Numbers on plus strand.** Whether to set the numbers relative to the positive or the negative strand in a nucleotide sequence (only applies to DNA sequences).
- **Lock numbers.** When you scroll vertically, the position numbers remain visible. (Only possible when the sequence is not wrapped.)
- **Lock labels.** When you scroll horizontally, the label of the sequence remains visible.
- **Sequence label.** Defines the label to the left of the sequence.
 - Name (this is the default information to be shown).
 - Accession (sequences downloaded from databases like GenBank have an accession number).
 - Latin name.
 - Latin name (accession).
 - Common name.
 - Common name (accession).
- **Matching residues as dots** Residues in aligned sequences identical to residues in the first (reference) sequence will be presented as dots. An option that is only available for "Alignments" and "Read mappings".

Annotation Layout and Annotation Types See section [15.3.1](#).

Restriction sites

Please see section [23.1.1](#).

Motifs

See section [18.9.1](#).

Residue coloring

These preferences make it possible to color both the residue letter and set a background color for the residue.

- **Non-standard residues.** For nucleotide sequences this will color the residues that are not C, G, A, T or U. For amino acids only B, Z, and X are colored as non-standard residues.
 - **Foreground color.** Sets the color of the letter. Click the color box to change the color.
 - **Background color.** Sets the background color of the residues. Click the color box to change the color.
- **Rasmol colors.** Colors the residues according to the Rasmol color scheme.
See <http://www.openrasmol.org/doc/rasmol.html>
 - **Foreground color.** Sets the color of the letter. Click the color box to change the color.
 - **Background color.** Sets the background color of the residues. Click the color box to change the color.
- **Polarity colors (only protein).** Colors the residues according to the following categories:
 - **Green** neutral, polar
 - **Black** neutral, nonpolar
 - **Red** acidic, polar
 - **Blue** basic ,polar
 - As with other options, you can choose to set or change the coloring for either the residue letter or its background:
 - * **Foreground color.** Sets the color of the letter. Click the color box to change the color.
 - * **Background color.** Sets the background color of the residues. Click the color box to change the color.
- **Trace colors (only DNA).** Colors the residues according to the color conventions of chromatogram traces: A=green, C=blue, G=black, and T=red.
 - **Foreground color.** Sets the color of the letter.
 - **Background color.** Sets the background color of the residues.

Nucleotide info

These preferences apply only to nucleotide sequences.

The data points for graph representations can be exported (see section [8.3](#)).

- **Translation.** Displays a translation into protein just below the nucleotide sequence. Depending on the zoom level, the amino acids are displayed with three letters or one letter. In cases where variants are present in the reads, synonymous variants are shown in orange in the translated sequence whereas non-synonymous are shown in red.
 - **Frame.** Determines where to start the translation.
 - * **ORF/CDS.** If the sequence is annotated, the translation will follow the CDS or ORF annotations. If annotations overlap, only one translation will be shown. If only one annotation is visible, the Workbench will attempt to use this annotation to mark the start and stop for the translation. In cases where this is not possible, the first annotation will be used (i.e. the one closest to the 5' end of the sequence).
 - * **Selection.** This option will only take effect when you make a selection on the sequence. The translation will start from the first nucleotide selected. Making a new selection will automatically display the corresponding translation. Read more about selecting in section 15.2.2.
 - * **+1 to -1.** Select one of the six reading frames.
 - * **All forward/All reverse.** Shows either all forward or all reverse reading frames.
 - * **All.** Select all reading frames at once. The translations will be displayed on top of each other.
 - **Table.** The translation table to use in the translation. For more about translation tables, see section 19.4.
 - **Only AUG start codons.** For most genetic codes, a number of codons can be start codons (TTG, CTG, or ATG). These will be colored green, unless selecting the "Only AUG start codons" option, which will result in only the AUG codons colored in green.
 - **Single letter codes.** Choose to represent the amino acids with a single letter instead of three letters.
- **Quality scores.** For sequencing data containing quality scores, the quality score information can be displayed along the sequence.
 - **Show as probabilities.** Converts quality scores to error probabilities on a 0-1 scale, i.e. not log-transformed.
 - **Foreground color.** Colors the letter using a gradient, where the left side color is used for low quality and the right side color is used for high quality. The sliders just above the gradient color box can be dragged to highlight relevant levels. The colors can be changed by clicking the box. This will show a list of gradients to choose from.
 - **Background color.** Sets a background color of the residues using a gradient in the same way as described above.
 - **Graph.** The quality scores are displayed as a graph.
 - * **Height.** Specifies the height of the graph.
 - * **Type.** The graph can be displayed as Line plot, Bar plot or as a Color bar.
 - * **Color box.** For Line and Bar plots, the color of the plot can be set by clicking the color box. For Colors, the color box is replaced by a gradient color box as described under Foreground color.
- **Trace data.** See section 22.1.

- **G/C content.** Calculates the G/C content of a part of the sequence and shows it as a gradient of colors or as a graph below the sequence.
 - **Window length.** Determines the length of the part of the sequence to calculate. A window length of 9 will calculate the G/C content for the nucleotide in question plus the 4 nucleotides to the left and the 4 nucleotides to the right. A narrow window will focus on small fluctuations in the G/C content level, whereas a wider window will show fluctuations between larger parts of the sequence.
 - **Foreground color.** Colors the letter using a gradient, where the left side color is used for low levels of G/C content and the right side color is used for high levels of G/C content. The sliders just above the gradient color box can be dragged to highlight relevant levels of G/C content. The colors can be changed by clicking the box. This will show a list of gradients to choose from.
 - **Background color.** Sets a background color of the residues using a gradient in the same way as described above.
 - **Graph.** The G/C content levels are displayed as a graph.
 - * **Height.** Specifies the height of the graph.
 - * **Type.** The graph can be displayed as Line plot, Bar plot or as a Color bar.
 - * **Color box.** For Line and Bar plots, the color of the plot can be set by clicking the color box. For Colors, the color box is replaced by a gradient color box as described under Foreground color.

When zoomed out, the graph displays G/C content only for a subset of evenly spaced positions. Because insertions shifts reference positions, zoomed-out graphs with and without insertions may not be directly comparable, as G/C content may be displayed for different positions.
- **Secondary structure.** Allows you to choose how to display a symbolic representation of the secondary structure along the sequence.

See section 26.2.3 for a detailed description of the settings.

Protein info

These preferences only apply to proteins. The first nine items are different hydrophobicity scales. These are described in section 20.3.1.

- **Kyte-Doolittle.** The Kyte-Doolittle scale is widely used for detecting hydrophobic regions in proteins. Regions with a positive value are hydrophobic. This scale can be used for identifying both surface-exposed regions as well as transmembrane regions, depending on the window size used. Short window sizes of 5-7 generally work well for predicting putative surface-exposed regions. Large window sizes of 19-21 are well suited for finding transmembrane domains if the values calculated are above 1.6 [Kyte and Doolittle, 1982]. These values should be used as a rule of thumb and deviations from the rule may occur.
- **Cornette.** Cornette *et al.* computed an optimal hydrophobicity scale based on 28 published scales [Cornette *et al.*, 1987]. This optimized scale is also suitable for prediction of alpha-helices in proteins.

- **Engelman.** The Engelman hydrophobicity scale, also known as the GES-scale, is another scale which can be used for prediction of protein hydrophobicity [Engelman et al., 1986]. As the Kyte-Doolittle scale, this scale is useful for predicting transmembrane regions in proteins.
- **Eisenberg.** The Eisenberg scale is a normalized consensus hydrophobicity scale which shares many features with the other hydrophobicity scales [Eisenberg et al., 1984].
- **Rose.** The hydrophobicity scale by Rose et al. is correlated to the average area of buried amino acids in globular proteins [Rose et al., 1985]. This results in a scale which is not showing the helices of a protein, but rather the surface accessibility.
- **Janin.** This scale also provides information about the accessible and buried amino acid residues of globular proteins [Janin, 1979].
- **Hopp-Woods.** Hopp and Woods developed their hydrophobicity scale for identification of potentially antigenic sites in proteins. This scale is basically a hydrophilic index where apolar residues have been assigned negative values. Antigenic sites are likely to be predicted when using a window size of 7 [Hopp and Woods, 1983].
- **Welling.** [Welling et al., 1985] Welling et al. used information on the relative occurrence of amino acids in antigenic regions to make a scale which is useful for prediction of antigenic regions. This method is better than the Hopp-Woods scale of hydrophobicity which is also used to identify antigenic regions.
- **Kolaskar-Tongaonkar.** A semi-empirical method for prediction of antigenic regions has been developed [Kolaskar and Tongaonkar, 1990]. This method also includes information of surface accessibility and flexibility and at the time of publication the method was able to predict antigenic determinants with an accuracy of 75%.
- **Surface Probability.** Display of surface probability based on the algorithm by [Emini et al., 1985]. This algorithm has been used to identify antigenic determinants on the surface of proteins.
- **Chain Flexibility.** Display of backbone chain flexibility based on the algorithm by [Karplus and Schulz, 1985]. It is known that chain flexibility is an indication of a putative antigenic determinant.

Find

The Find function can be used for searching the sequence and is invoked by pressing Ctrl + Shift + F (⌘ + Shift + F on Mac). Initially, specify the 'search term' to be found, select the type of search (see various options in the following) and finally click on the Find button. The first occurrence of the search term will then be highlighted. Clicking the find button again will find the next occurrence and so on. If the search string is found, the corresponding part of the sequence will be selected.

- **Search term.** Enter the text or number to search for. The search function does not discriminate between lower and upper case characters.
- **Sequence search.** Search the nucleotides or amino acids. For amino acids, the single letter abbreviations should be used for searching. The sequence search also has a set of advanced search parameters:

- Include negative strand. This will search on the negative strand as well.
- Treat ambiguous characters as wildcards in search term. If you search for e.g. ATN, you will find both ATG and ATC. If you wish to find literally exact matches for ATN (i.e. only find ATN - not ATG), this option should not be selected.
- Treat ambiguous characters as wildcards in sequence. If you search for e.g. ATG, you will find both ATG and ATN. If you have large regions of Ns, this option should not be selected.

Note that if you enter a position instead of a sequence, it will automatically switch to position search.

- **Annotation search.** Search the annotations on the sequence. The search is performed both on the labels of the annotations, but also on the text appearing in the tooltip that you see when you keep the mouse cursor fixed. If the search term is found, the part of the sequence corresponding to the matching annotation is selected. The option "Include translations" means that you can choose to search for translations *which are part of an annotation* (in some cases, CDS annotations contain the amino acid sequence in a "/translation" field). But it will not dynamically translate nucleotide sequences, nor will it search the translations that can be enabled using the "Nucleotide info" side panel.
- **Position search.** Find a specific position on the sequence. In order to find an interval, e.g. from position 500 to 570, enter "500..570" in the search field. This will make a selection from position 500 to 570 (both included). Notice the two periods (..) between the start and end number. If you enter positions including thousands separators like 123,345, the comma will just be ignored and it would be equivalent to entering 123345.
- **Include negative strand.** When searching the sequence for nucleotides or amino acids, you can search on both strands.
- **Name search.** Search for sequence names. This is useful for searching sequence lists and mapping results for example.

This concludes the description of the **View Preferences**. Next, the options for selecting and editing sequences are described.

Text format

These preferences allow you to adjust the format of all the text in the view (both residue letters, sequence name and translations if they are shown).


- **Text size.** Specify a font size for the text in the view.
- **Font.** Specify a font for the text in the view.
- **Bold.** Make the text for the residues bold.

Restriction sites in the Side Panel

Please see section [23.1.1](#).

15.2.2 Selecting parts of the sequence

You can select parts of a sequence:

Click Selection () in Toolbar | Press and hold down the mouse button on the sequence where you want the selection to start | move the mouse to the end of the selection while holding the button | release the mouse button

Alternatively, you can search for a specific interval using the find function described above.

If you have made a selection and wish to adjust it:

drag the edge of the selection (you can see the mouse cursor change to a horizontal arrow

or **press and hold the Shift key while using the right and left arrow keys to adjust the right side of the selection.**

If you wish to select the entire sequence:

double-click the sequence name to the left

Selecting non-contiguous parts of a sequence (multiselect) You can select non-contiguous parts of sequence by holding down the **Ctrl** button while making selections. Holding down the **Shift** button lets you extend or reduce an existing selection to the position you clicked.

To select a part of a sequence covered by an annotation:

right-click the annotation | Select annotation

or **double-click the annotation**

To select a fragment between two restriction sites that are shown on the sequence:

double-click the sequence between the two restriction sites

(Read more about restriction sites in section [15.2.1.](#))

Open a selection in a new view A selection can be opened in a new view and saved as a new sequence:

right-click the selection | Open selection in New View ()

This opens the annotated part of the sequence in a new view. The new sequence can be saved by dragging the tab of the sequence view into the **Navigation Area**.

The process described above is also the way to manually translate coding parts of sequences (CDS) into protein. You simply translate the new sequence into protein. This is done by:

**right-click the tab of the new sequence | Tools | Classical Sequence Analysis ()
| Nucleotide Analysis () | Translate to Protein ()**

A selection can also be copied to the clipboard and pasted into another program:

make a selection | Ctrl + C (⌘ + C on Mac)

Note! The annotations covering the selection will not be copied.

A selection of a sequence can be edited as described in the following section.

15.2.3 Editing the sequence

When you make a selection, it can be edited by:

right-click the selection | Edit Selection (✎)

A dialog appears displaying the sequence. You can add, remove or change the text and click **OK**. The original selected part of the sequence is now replaced by the sequence entered in the dialog. This dialog also allows you to paste text into the sequence using Ctrl + V (⌘ + V on Mac).

If you delete the text in the dialog and press **OK**, the selected text on the sequence will also be deleted. Another way to delete a part of the sequence is to:

right-click the selection | Delete Selection (✂)

If you wish to correct only one residue, this is possible by simply making the selection cover only one residue and then type the new residue.

Another way to edit the sequence is by inserting a restriction site. See section 23.1.3.

Note When editing annotated nucleotide sequences, the annotation content is not updated automatically (but its position is). Please refer to section 15.3.3 for details on annotation editing.

Before exporting annotated nucleotide sequences in GenBank format, ensure that the annotations in the Annotations Table reflect the edits that have been made to the sequence.

15.2.4 Sequence region types

The various annotations on sequences cover parts of the sequence. Some cover an interval, some cover intervals with unknown endpoints, some cover more than one interval etc. In the following, all of these will be referred to as *regions*. Regions are generally illustrated by markings (often arrows) on the sequences. An arrow pointing to the right indicates that the corresponding region is located on the positive strand of the sequence. Figure 15.8 is an example of three regions with separate colors.

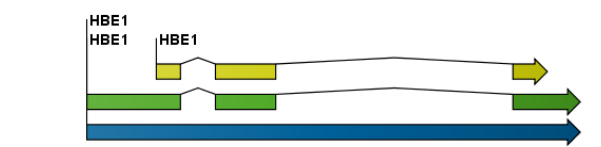


Figure 15.8: Three regions on a human beta globin DNA sequence (HUMHBB).

Figure 15.9 shows an artificial sequence with all the different kinds of regions.

15.2.5 Circular DNA

A sequence can be shown as a circular molecule:

Select a sequence in the Navigation Area and right-click on the file name | Hold the mouse over "Show" to enable a list of options | Select "Circular View" (🔄)

or **If the sequence is already open | Click "Show Circular View"** (🔄) at the lower left part of the view

This will open a view of the molecule similar to the one in figure 15.10.

- In the **Sequence Layout** preferences, only the following options are available in the circular view: **Numbers on plus strand**, **Numbers on sequence** and **Sequence label**.
- You cannot zoom in to see the residues in the circular molecule. If you wish to see these details, split the view with a linear view of the sequence
- In the **Annotation Layout**, you also have the option of showing the labels as **Stacked**. This means that there are no overlapping labels and that all labels of both annotations and restriction sites are adjusted along the left and right edges of the view.

Using split views to see details of the circular molecule

To see the nucleotides of a circular molecule you can open a new view displaying a circular view of the molecule:

Press and hold the Ctrl button (⌘ on Mac) | click Show Sequence (📄) at the bottom of the view

This will open a linear view of the sequence below the circular view. When you zoom in on the linear view you can see the residues as shown in figure 15.11.

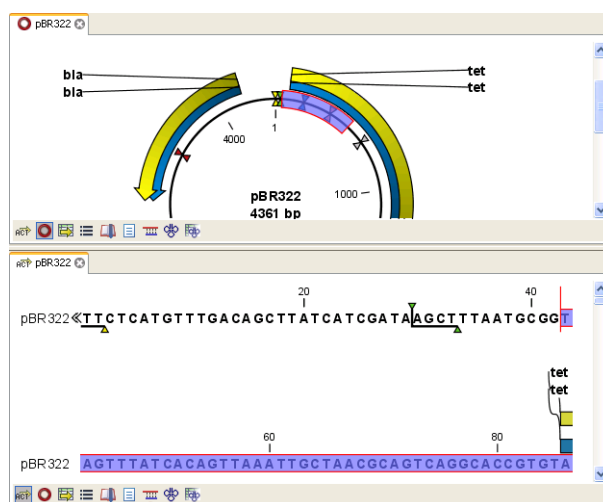


Figure 15.11: Two views showing the same sequence. The bottom view is zoomed in.

Note! If you make a selection in one of the views, the other view will also make the corresponding selection, providing an easy way for you to focus on the same region in both views.

Mark molecule as circular and specify starting point

You can mark a DNA molecule as circular or linear by right-clicking on its name in either the Sequence view or the Circular view. If the sequence is linear, you will see the option to mark it as circular and vice versa (see figure 15.12).

In the Sequence view, a sequence marked as circular is indicated by the use of double angle brackets at the start and end of the sequence. The linear or circular status of a sequence can also be seen in the Locus line of the Text view for a Sequence, or in the Linear column of the Table view of a Sequence List.

The starting point of a circular sequence can be changed by selecting the position of the new starting point and right-clicking on that selection to choose the option **Move Starting Point to**

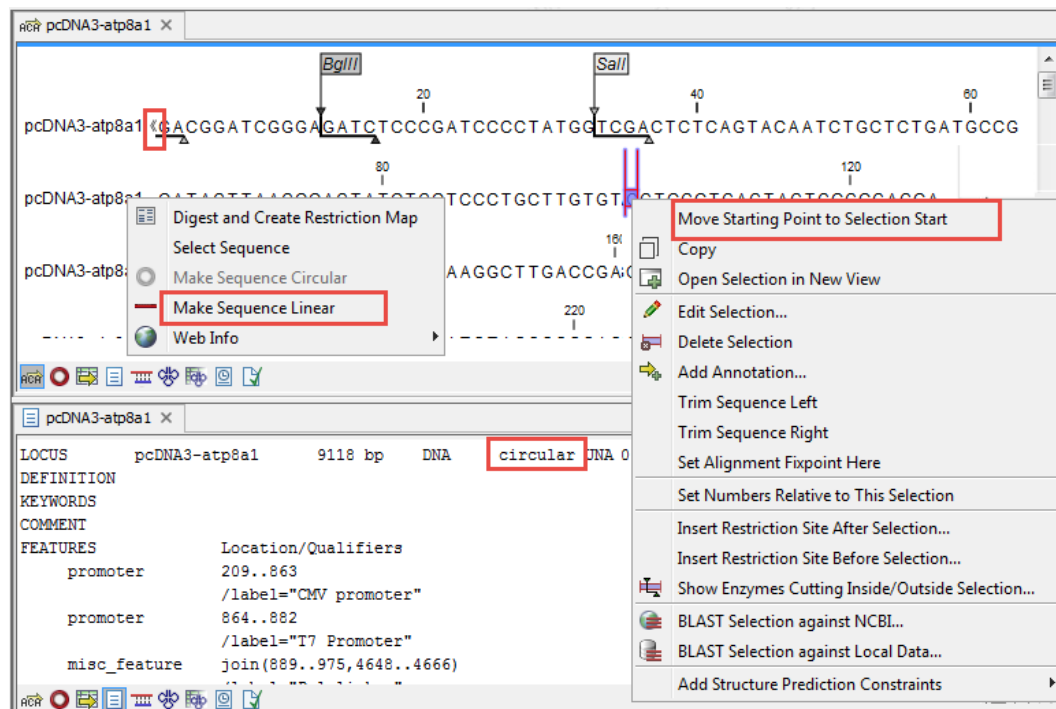


Figure 15.12: Double angle brackets mark the start and end of a circular sequence in linear view (top). The first line in the text view (bottom) contains information that the sequence is circular.

Selection Start (figure 15.13).

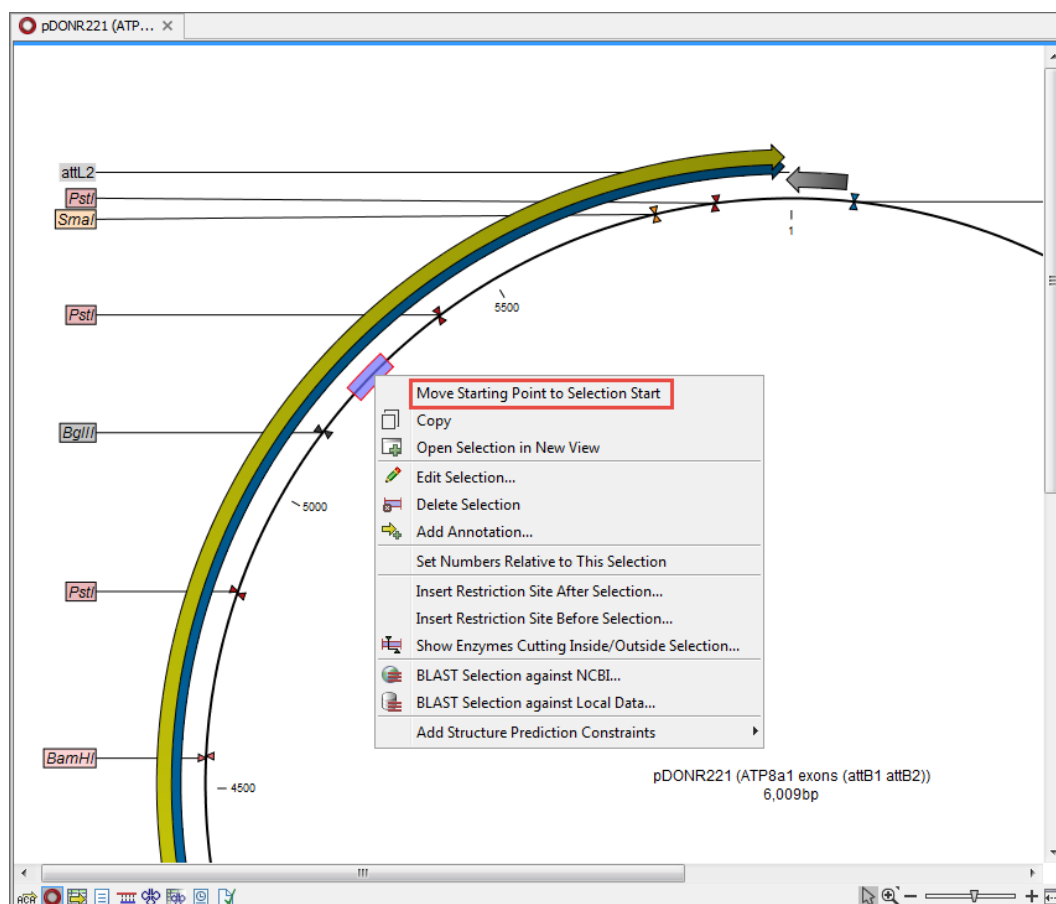


Figure 15.13: Right-click on a circular sequence to move the starting point to the selected position.

15.3 Working with annotations

Annotations provide information about specific regions of a sequence.

A typical example is the annotation of a gene on a genomic DNA sequence.

Annotations derive from different sources:

- Sequences downloaded from databases like GenBank are annotated.
- In some of the data formats that can be imported into *CLC Genomics Workbench*, sequences can have annotations (GenBank, EMBL and Swiss-Prot format).
- The result of a number of analyses in *CLC Genomics Workbench* are annotations on the sequence (e.g. finding open reading frames and restriction map analysis).
- A protein structure can be linked with a sequence (section 17.4.2), and atom groups defined on the structure transferred to sequence annotations or vice versa (section 17.4.3).
- You can manually add annotations to a sequence (described in the section 15.3.2).

If you would like to extract parts of a sequence (or several sequences) based on its annotations, you can find a description of how to do this in section 37.1.

Note! Annotations are included if you export the sequence in GenBank, Swiss-Prot, EMBL or CLC format. When exporting in other formats, annotations are not preserved in the exported file.

15.3.1 Viewing annotations

Annotations can be viewed in a number of different ways:

- As graphical arrows or boxes in all views displaying sequences (sequence lists, alignments etc)
- In a table in the Annotation (📄) view
- In the text view of sequences (📋)

These views are described in more detail in the following sections.

View Annotations in sequence views

Figure 15.14 shows an annotation displayed on a sequence.

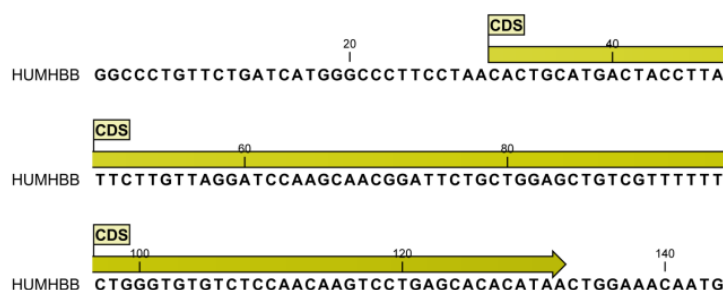


Figure 15.14: An annotation showing a coding region on a genomic dna sequence.

The various sequence views listed in section 15.3.1 have different default settings for showing annotations. However, they all have two groups in the **Side Panel** in common:

- **Annotation Layout**
- **Annotation Types**

The two groups are shown in figure 15.15.

In the **Annotation layout** group, you can specify how the annotations should be displayed (notice that there are some minor differences between the different sequence views):

- **Show annotations.** Determines whether the annotations are shown.
- **Position.**
 - **On sequence.** The annotations are placed on the sequence. The residues are visible through the annotations (if you have zoomed in to 100%).
 - **Next to sequence.** The annotations are placed above the sequence.
 - **Separate layer.** The annotations are placed above the sequence and above restriction sites (only applicable for nucleotide sequences).

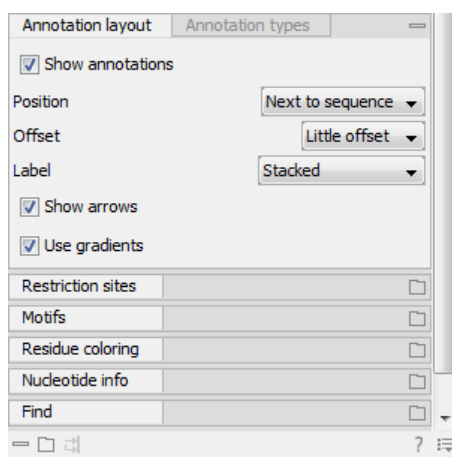


Figure 15.15: The annotation layout in the Side Panel. The annotation types can be shown by clicking on the "Annotation types" tab.

- **Offset.** If several annotations cover the same part of a sequence, they can be spread out.
 - **Piled.** The annotations are piled on top of each other. Only the one at front is visible.
 - **Little offset.** The annotations are piled on top of each other, but they have been offset a little.
 - **More offset.** Same as above, but with more spreading.
 - **Most offset.** The annotations are placed above each other with a little space between. This can take up a lot of space on the screen.
- **Label.** The name of the annotation can shown as a label. Additional information about the sequence is shown if you place the mouse cursor on the annotation and keep it still.
 - **No labels.** No labels are displayed.
 - **On annotation.** The labels are displayed in the annotation's box.
 - **Over annotation.** The labels are displayed above the annotations.
 - **Before annotation.** The labels are placed just to the left of the annotation.
 - **Flag.** The labels are displayed as flags at the beginning of the annotation.
 - **Stacked.** The labels are offset so that the text of all labels is visible. This means that there is varying distance between each sequence line to make room for the labels.
- **Show arrows.** Displays the end of the annotation as an arrow. This can be useful to see the orientation of the annotation (for DNA sequences). Annotations on the negative strand will have an arrow pointing to the left.
- **Use gradients.** Fills the boxes with gradient color.

In the **Annotation types** group, you can choose which kinds of annotations that should be displayed. This group lists all the types of annotations that are attached to the sequence(s) in the view. For sequences with many annotations, it can be easier to get an overview if you deselect the annotation types that are not relevant.

Unchecking the checkboxes in the **Annotation layout** will not remove this type of annotations from the sequence - it will just hide them from the view.

Besides selecting which types of annotations that should be displayed, the **Annotation types** group is also used to change the color of the annotations on the sequence. Click the colored square next to the relevant annotation type to change the color.

This will display a dialog with five tabs: Swatches, HSB, HSI, RGB, and CMYK. They represent five different ways of specifying colors. Apply your settings and click **OK**. When you click **OK**, the color settings cannot be reset. The **Reset** function only works for changes made before pressing **OK**.

Furthermore, the **Annotation types** can be used to easily browse the annotations by clicking the small button (🔍) next to the type. This will display a list of the annotations of that type (see figure 15.16).

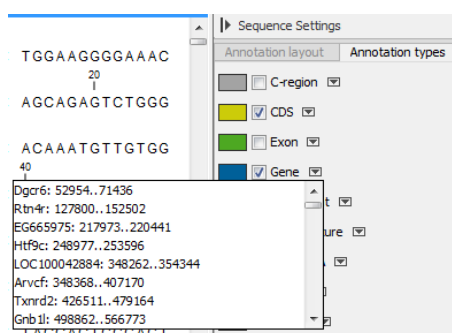


Figure 15.16: *Browsing the gene annotations on a sequence.*

Clicking an annotation in the list will select this region on the sequence. In this way, you can quickly find a specific annotation on a long sequence.

Note: A waved end on an annotation (figure 15.17) means that the annotation is torn, i.e., it extends beyond the sequence displayed. An annotation can be torn when a new, smaller sequence has been created from a larger sequence. A common example of this situation is when you select a section of a stand-alone sequence and open it in a new view. If there are annotations present within this selected region that extend beyond the selection, then the selected sequence shown in the new view will exhibit these torn annotations.

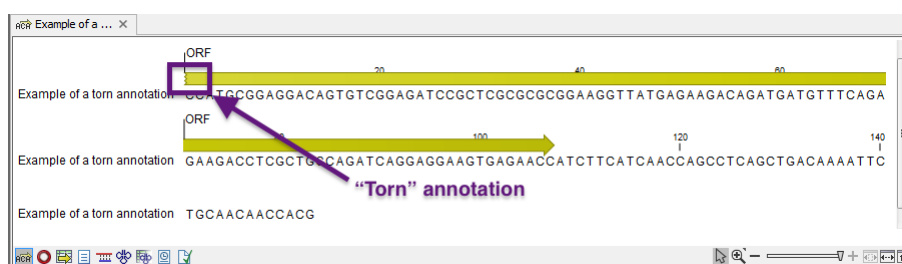


Figure 15.17: *Example of a torn annotation on a sequence.*

View Annotations in a table

The Annotation Table (📄) view lists the annotations in a table. Standard functionality for working with tables applies, as described in section 9. Here we highlight functionality of particular interest when working with annotations.

This view is useful for getting a quick overview of annotations, and for filtering so that only the annotations of interest are listed. From this view, you can edit and add annotations, export selected annotations to a gff3 format file, and delete annotations. This functionality is described in more detail below.

To open the Annotation Table (📄) view:

Select a sequence in the Navigation Area and right-click on the file name | Hold the mouse over "Show" to enable a list of options | Annotation Table (📄)

or **If the sequence is already open | Click Show Annotation Table (📄) at the lower left part of the view**

This will open a view similar to the one in figure 15.18).

Name	Type	Region	Qualifiers
Atp8a1	Gene	1..228194	/gene=Atp8a1 /note=Derived by automated computational analysis using gene prediction method: BestRefseq Supporting evidence includes similarity to: 2 mRNAs /db_xref="GeneID:11980" /db_xref=MGI:1330848
Atp8a1	CDS	join(222..270,32851..32...	/gene=Atp8a1 /GO_component=integral to membrane; membrane /GO_function=ATP binding; ATPase activity; ATPase activity, coupled to transmembrane movement of ions, phosphorylative mechanism; catalytic activity; hydrolase activity; hydrolase activity, acting on acid anhydrides, catalyzing transmembrane movement of substances; magnesium ion binding; metal ion binding; nucleotide binding; phospholipid-translocating ATPase activity /GO_process=cation transport; metabolism /note=isoform b is encoded by transcript variant 2; ATPase 8A1, p

Figure 15.18: A table showing annotations on the sequence.

In the **Side Panel** you can show or hide individual annotation types in the table. E.g. if you only wish to see "gene" annotations, de-select the other annotation types so that only "gene" is selected.

Each row in the table is an annotation which is represented with the following information:

- **Name.**
- **Type.**
- **Region.**
- **Qualifiers.**

This information corresponds to the information in the dialog when you edit and add annotations (see section 15.3.2).

The Name, Type and Region for each annotation can be edited simply by double-clicking, typing the change directly, and pressing **Enter**. See section 15.3.3 for further information about editing annotations.

15.3.2 Adding annotations

Annotations can be added to a sequence from the graphical view or the Annotation Table view.

Graphical view Select the region you want the annotation to cover, right-click on the selected region, and choose the option **Add Annotation** (➡) from the menu.

See section 15.2.2 on how to make selections that are not contiguous.

Annotation Table view Right-click anywhere in this view and select the option **Add Annotation** (➡) from the menu. (Note that this option is not enabled in sequence lists.)

In both cases, a dialog opens where you can provide information about the annotation (figure 15.19).

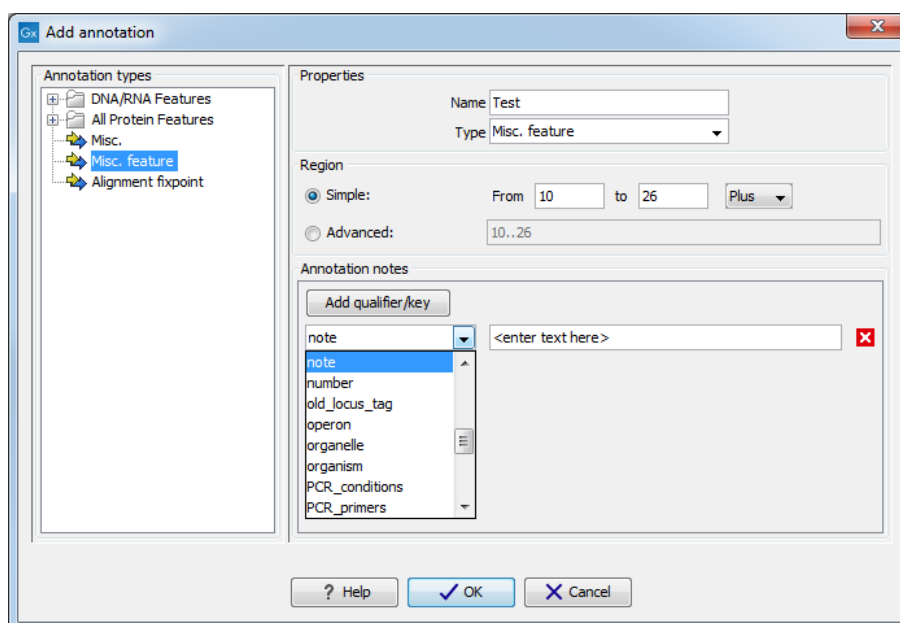


Figure 15.19: The Add annotation dialog.

The left-hand part of the dialog lists a number of **Annotation types**. When you have selected an annotation type, it appears in **Type** to the right. You can also select an annotation directly in this list. Choosing an annotation type is mandatory. If you wish to use an annotation type which is not present in the list, simply enter this type into the **Type** field ¹.

The right-hand part of the dialog contains the following text fields:

- **Name.** The name of the annotation which can be shown on the label in the sequence views. (Whether the name is actually shown depends on the **Annotation Layout** preferences, see section 15.3.1).
- **Type.** Reflects the left-hand part of the dialog as described above. You can also choose directly in this list or type your own annotation type.
- **Region.** If you have already made a selection, this field will show the positions of the selection. You can modify the region further using the conventions of DDBJ, EMBL and

¹Note that your own annotation types will be converted to "unsure" when exporting in GenBank format. As long as you use the sequence in CLC format, your own annotation type will be preserved


GenBank. The following are examples of how to use the syntax (based on <https://www.ncbi.nlm.nih.gov/collab/FT/>):

- **467**. Points to a single residue in the presented sequence.
 - **340..565**. Points to a continuous range of residues bounded by and including the starting and ending residues.
 - **<345..500**. Indicates that the exact lower boundary point of a region is unknown. The location begins at some residue previous to the first residue specified (which is not necessarily contained in the presented sequence) and continues up to and including the ending residue.
 - **<1..888**. The region starts before the first sequenced residue and continues up to and including residue 888.
 - **1..>888**. The region starts at the first sequenced residue and continues beyond residue 888.
 - **(102.110)**. Indicates that the exact location is unknown, but that it is one of the residues between residues 102 and 110, inclusive.
 - **123^124**. Points to a site between residues 123 and 124.
 - **join(12..78,134..202)**. Regions 12 to 78 and 134 to 202 should be joined to form one contiguous sequence.
 - **complement(34..126)** Start at the residue complementary to 126 and finish at the residue complementary to residue 34 (the region is on the strand complementary to the presented strand).
 - **complement(join(2691..4571,4918..5163))**. Joins regions 2691 to 4571 and 4918 to 5163, then complements the joined segments (the region is on the strand complementary to the presented strand).
 - **join(complement(4918..5163),complement(2691..4571))**. Complements regions 4918 to 5163 and 2691 to 4571, then joins the complemented segments (the region is on the strand complementary to the presented strand).
- **Annotations**. In this field, you can add more information about the annotation like comments and links. Click the **Add qualifier/key** button to enter information. Select a qualifier which describes the kind of information you wish to add. If an appropriate qualifier is not present in the list, you can type your own qualifier. The pre-defined qualifiers are derived from the GenBank format. You can add as many qualifier/key lines as you wish by clicking the button. Redundant lines can be removed by clicking the delete icon (✖). The information entered on these lines is shown in the annotation table (see section 15.3.1) and in the yellow box which appears when you place the mouse cursor on the annotation. If you write a hyperlink in the **Key** text field, like e.g. "digitalinsights.qiagen.com", it will be recognized as a hyperlink. Clicking the link in the annotation table will open a web browser.

Click **OK** to add the annotation.

Note! The annotation will be included if you export the sequence in GenBank, Swiss-Prot or CLC format. When exporting in other formats, annotations are not preserved in the exported file.

15.3.3 Editing annotations

Annotations can be edited from the graphical sequence view or from the Annotation Table  view.

For general information about the Annotation Table  view, see section 15.3.1.


Basic annotation editing in the sequence view

To edit an annotation from the sequence view:

Right-click on the annotation | Edit Annotation 

A dialog like that in figure 15.19 will appear. Edit the fields as needed and click on **OK** to save your changes.

Editing annotations in the Annotation Table view

In the Annotation Table  view, each part of an annotation can be updated by double-clicking in a cell, editing the contents, and pressing **Enter**. In addition, editing options are among the options available in the right-click menu (figure 15.20).

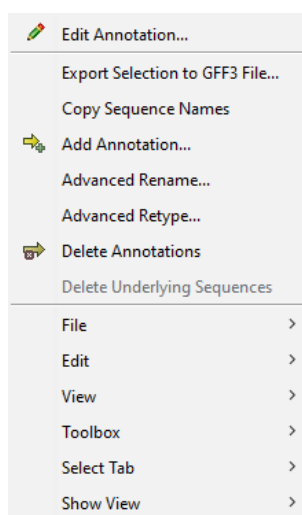


Figure 15.20: The right-click menu in the Annotation Table view contains options for adding, editing, exporting and deleting annotations.

Options related to editing annotations in this menu are:

- **Edit Annotation...** This option is only enabled if a single annotation is selected in the table. It will open the same dialog used to edit annotations from the sequence view (figure 15.19).
- **Advanced Rename...** Choose this to rename the selected annotations using qualifiers or annotation types. The options in the Rename dialog (figure 15.21) are:
 - **Use this qualifier** Choose the qualifier to use as that annotation name from a drop-down list of qualifiers available in the selected annotations. Selected annotations that do not include the selected qualifier will not be renamed. If an annotation has multiple qualifiers of the same type, the first is used for renaming.

- **Use annotation type as name** The annotation's type will be used for the annotation name E.g. if you have an annotation of type "Promoter", it will get "Promoter" as its name by using this option.
- **Advanced Retype...** Choose this to edit the type of one or more annotations. The options in the Retype dialog (figure 15.22) are:
 - **Use this qualifier** Choose the qualifier to use as the annotation type from a drop-down list of qualifiers available in the selected annotations. Selected annotations that do not include the selected qualifier will not be retyped. If an annotation has multiple qualifiers of the same type, the first is used for the new type.
 - **New type** Enter an annotation type to apply or click on the arrows at the right of the field to see a drop-down list of pre-defined annotation types.
 - **Use annotation name as type** Use the annotation name as its type. E.g. if you have an annotation named "Promoter", it will get "Promoter" as its type by using this option.

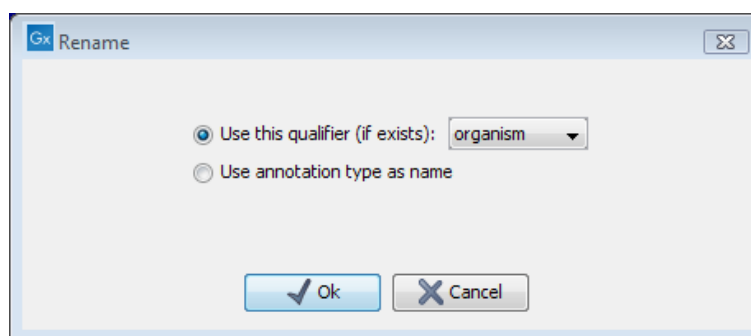


Figure 15.21: The Advanced Rename dialog.

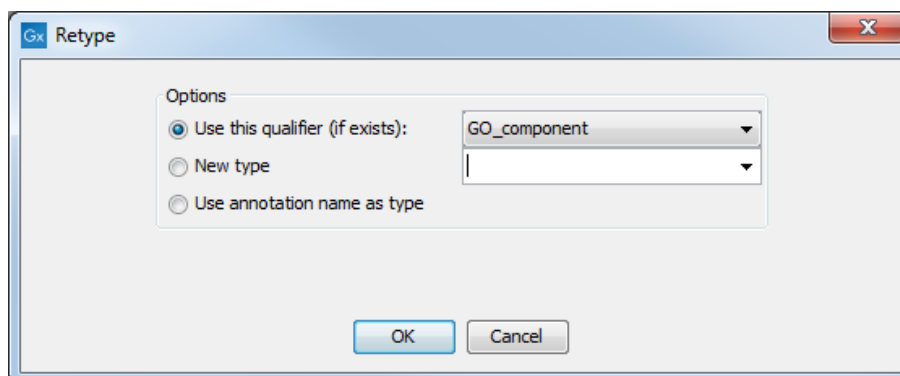


Figure 15.22: The Advanced Retype dialog.

15.3.4 Export annotations to a gff3 format file

Annotations selected in the Annotation Table () view can be exported to a gff3 format file by selecting the option **Export Selection to GFF3 File** in the right-click menu (figure 15.20).

15.3.5 Removing annotations

Annotations can be hidden using the **Annotation Types** preferences in the **Side Panel** to the right of the view (see section 15.3.1). In order to completely remove the annotation:


right-click the annotation | Delete Annotation ()

If you want to remove all annotations of one type:

right-click an annotation of the type you want to remove | Delete | Delete Annotations of Type "type"

If you want to remove all annotations from a sequence:

right-click an annotation | Delete | Delete All Annotations

The removal of annotations can be undone using Ctrl + Z or Undo () in the Toolbar.

If you have more sequences (e.g. in a sequence list, alignment or contig), you have two additional options:

right-click an annotation | Delete | Delete All Annotations from All Sequences


right-click an annotation | Delete | Delete Annotations of Type "type" from All Sequences

15.4 Element information

The normal view of a sequence (by double-clicking) shows the annotations as boxes along the sequence, but often there is more information available about sequences. This information is available through the **Element info** view.

To view the sequence information:

Select a sequence in the Navigation Area and right-click on the file name | Hold the mouse over "Show" to enable a list of options | Element Info ()

Another way to show the text view is to open the sequence in the **View Area** and click on the "Show Element Info" icon () found at the bottom of the window.

This will display a view similar to fig 15.23.

All the lines in the view are headings, and the corresponding text can be shown by clicking the text. The information available depends on the origin of the sequence.

- **Name.** The name of the sequence which is also shown in sequence views and in the **Navigation Area**.
- **Description.** A description of the sequence.
- **Metadata.** The Metadata table and the detailed metadata values associated with the sequence.
- **Comments.** The author's comments about the sequence.
- **Keywords.** Keywords describing the sequence.
- **Db source.** Accession numbers in other databases concerning the same sequence.



Figure 15.23: The initial display of sequence info for the HUMHBB DNA sequence from the Example data.


- **Gb Division.** Abbreviation of GenBank divisions. See section 3.3 in the GenBank release notes for a full list of GenBank divisions.
- **Length.** The length of the sequence.
- **Modification date.** Modification date from the database. This means that this date does not reflect your own changes to the sequence. See the History view, described in section 2.5 for information about the latest changes to the sequence after it was downloaded from the database.
- **Latin name.** Latin name of the organism.
- **Common name.** Scientific name of the organism.
- **Taxonomy name.** Taxonomic classification levels.
- **Read group** Read group identifier "ID", technology used to produced the reads "Platform", and sample name "Sample".
- **Paired Status.** Unpaired or Paired sequences, with in this case the Minimum and Maximum distances as well as the Read orientation set during import.

Some of the information can be edited by clicking the blue **Edit** text. This means that you can add your own information to sequences that do not derive from databases.

15.5 View as text

A sequence can be viewed as text without any layout and text formatting. This displays all the information about the sequence in the GenBank file format. To view a sequence as text:

Select a sequence in the Navigation Area and right-click on the file name | Hold the mouse over "Show" to enable a list of options | Select "Text View" ()

Another way to show the text view is to open the sequence in the **View Area** and click on the "Show Text View" icon () found at the bottom of the window.

This makes it possible to see background information about e.g. the authors and the origin of DNA and protein sequences. Selections or the entire text of the **Sequence Text View** can be copied and pasted into other programs:

Much of the information is also displayed in the **Sequence info**, where it is easier to get an overview (see section [15.4](#).)

In the **Side Panel**, you find a search field for searching the text in the view.

Chapter 16

BLAST search

Contents

16.1 Running BLAST searches	412
16.1.1 BLAST at NCBI	412
16.1.2 BLAST against local data	416
16.2 Output from BLAST searches	419
16.2.1 Graphical overview for each query sequence	419
16.2.2 Overview BLAST table	420
16.2.3 BLAST graphics	421
16.2.4 BLAST HSP table	422
16.2.5 BLAST hit table	424
16.2.6 Extracting a consensus sequence from a BLAST result	425
16.3 Local BLAST databases	425
16.3.1 Download NCBI pre-formatted BLAST databases	426
16.3.2 Make pre-formatted BLAST databases available	427
16.3.3 Create local BLAST databases	427
16.4 Manage BLAST databases	428
16.5 Bioinformatics explained: BLAST	429
16.5.1 How does BLAST work?	430
16.5.2 Which BLAST program should I use?	432
16.5.3 Which BLAST options should I change?	433
16.5.4 Where can I get the BLAST+ programs	434
16.5.5 What you cannot get out of BLAST	434
16.5.6 Other useful resources	435

CLC Genomics Workbench offers to conduct BLAST searches on protein and DNA sequences. In short, a BLAST search identifies homologous sequences between your input (query) query sequence and a database of sequences [McGinnis and Madden, 2004]. BLAST (Basic Local Alignment Search Tool), identifies homologous sequences using a heuristic method which finds short matches between two sequences. After initial match BLAST attempts to start local alignments from these initial matches.

If you are interested in the bioinformatics behind BLAST, there is an easy-to-read explanation of this in section 16.5.

Figure 16.1 shows an example of a BLAST result in the CLC Genomics Workbench.

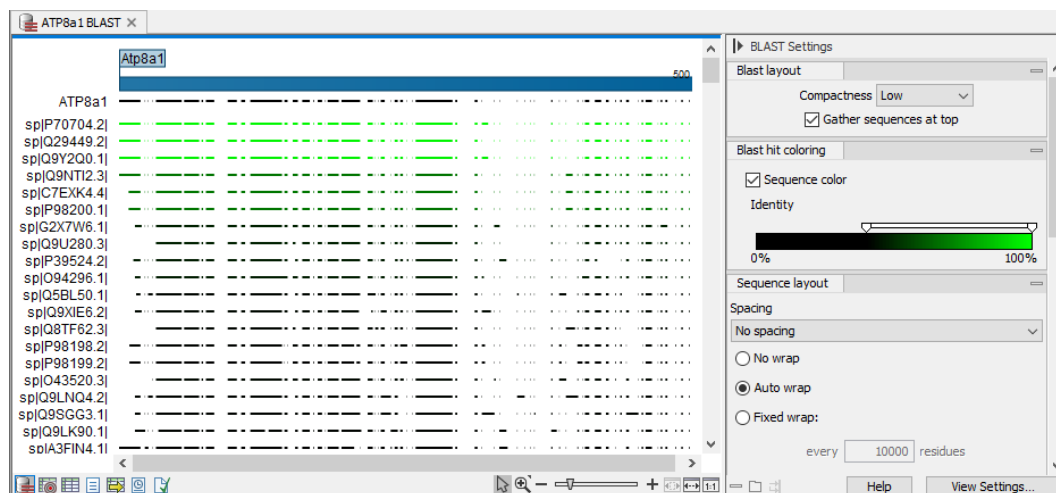


Figure 16.1: Display of the output of a BLAST search. At the top is there a graphical representation of BLAST hits with tool-tips showing additional information on individual hits. Below is a tabular form of the BLAST results.

16.1 Running BLAST searches

With the CLC Genomics Workbench there are two ways of performing BLAST searches: You can either have the BLAST process run on NCBI's BLAST servers (<https://blast.ncbi.nlm.nih.gov/Blast.cgi>) or you can perform the BLAST search on your own computer.

The advantage of running the BLAST search on NCBI servers is that you have readily access to the popular, and often very large, BLAST databases without having to download them to your own computer. The advantages of running BLAST on your own computer include that you can use your own sequence collections as blast databases, and that running big batch BLAST jobs can be faster and more reliable when done locally.

16.1.1 BLAST at NCBI

When running a BLAST search at the NCBI, the Workbench sends the sequences you select to the NCBI's BLAST servers. When the results are ready, they will be automatically downloaded and displayed in the Workbench. When you enter a large number of sequences for searching with BLAST, the Workbench automatically splits the sequences up into smaller subsets and sends one subset at the time to NCBI. This is to avoid exceeding any internal limits the NCBI places on the number of sequences that can be submitted to them for BLAST searching. The size of the subset created in the CLC software depends both on the number and size of the sequences.

To start a BLAST job to search your sequences against databases held at the NCBI, go to:

Tools | BLAST () | BLAST at NCBI ()

Alternatively, use the keyboard shortcut: Ctrl+Shift+B for Windows and ⌘ +Shift+B on Mac OS.

This opens the dialog seen in figure 16.2

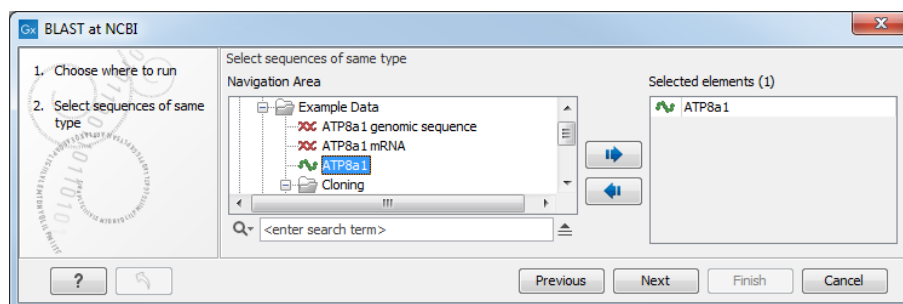


Figure 16.2: Choose one or more sequences to conduct a BLAST search with.

Select one or more sequences of the same type (either DNA or protein) and click **Next**.

In this dialog, you choose which type of BLAST search to conduct, and which database to search against (figure 16.3). The databases at the NCBI listed in the dropdown box will correspond to the query sequence type you have, DNA or protein, and the type of blast search you can choose among to run. A complete list of these databases can be found in Appendix C. Here you can also read how to add additional databases available the NCBI to the list provided in the dropdown menu.

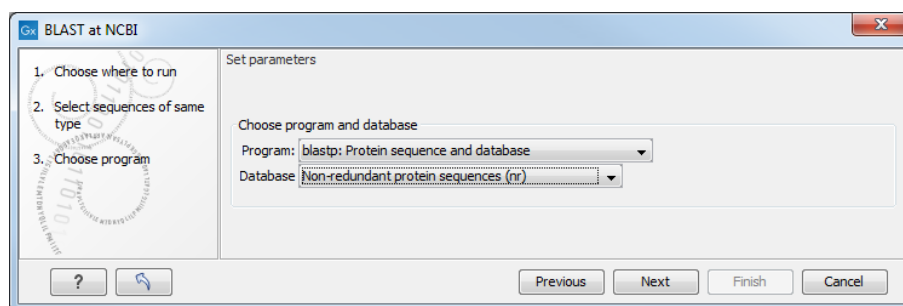


Figure 16.3: Choose a BLAST Program and a database for the search.

BLAST programs for DNA query sequences:

- **blastn: DNA sequence against a DNA database.** Searches for DNA sequences with homologous regions to your nucleotide query sequence.
- **blastx: Translated DNA sequence against a Protein database.** Automatic translation of your DNA query sequence in six frames; these translated sequences are then used to search a protein database.
- **tblastx: Translated DNA sequence against a Translated DNA database.** Automatic translation of your DNA query sequence and the DNA database, in six frames. The resulting peptide query sequences are used to search the resulting peptide database. Note that this type of search is computationally intensive.

BLAST programs for protein query sequences:

- **blastp: Protein sequence against Protein database.** Used to look for peptide sequences with homologous regions to your peptide query sequence.

- **tblastn: Protein sequence against Translated DNA database.** Peptide query sequences are searched against an automatically translated, in six frames, DNA database.

If you search against the **Protein Data Bank protein** database homologous sequences are found to the query sequence, these can be downloaded and opened with the 3D view.

Click **Next**.

This window, see figure 16.4, allows you to choose parameters to tune your BLAST search, to meet your requirements.

Figure 16.4: Parameters that can be set before submitting a BLAST search.

When choosing blastx or tblastx to conduct a search, you get the option of selecting a translation table for the genetic code. The standard genetic code is set as default. This setting is particularly useful when working with organisms or organelles that have a genetic code different from the standard genetic code.

BLAST search parameters are described below. See <https://blast.ncbi.nlm.nih.gov/doc/blast-topics/> for further details.

- **Limit by Entrez query.** BLAST searches can be limited to the results of an Entrez query against the database chosen. This can be used to limit searches to subsets of entries in the BLAST databases. Any terms can be entered that would normally be allowed in an Entrez search session. More information about Entrez queries can be found at https://www.ncbi.nlm.nih.gov/books/NBK3837/#EntrezHelp.Entrez_Searching_Options. The syntax described there is the same as would be accepted in the CLC interface. Some commonly used Entrez queries are pre-entered and can be chosen in the drop down menu.
- **Mask low complexity regions.** Mask off segments of the query sequence that have low compositional complexity.
- **Mask low complexity regions.** Mask off segments of the query sequence that have low compositional complexity. Filtering can eliminate statistically significant, but biologically uninteresting reports from the BLAST output (e.g. hits against common acidic-, basic- or proline-rich regions), leaving the more biologically interesting regions of the query sequence available for specific matching against database sequences.

- **Expect.** The threshold for reporting matches against database sequences. The Expect value (E-value) describes the number of hits one can expect to see matching a query by chance when searching against a database of a given size. If the E-value ascribed to a match is greater than the value entered in the Expect field, the match will not be reported. Details of how E-values are calculated can be found at the NCBI: <https://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-1.html>. Lower thresholds are more stringent, leading to fewer chance matches being reported. Increasing the threshold results in more matches being reported, but many may just matching by chance, not due to any biological similarity. Values lower than 1 can be entered as decimals, or in scientific notation. For example, 0.001, 1e-3 and 10e-4 would be equivalent and acceptable values.
- **Word Size.** BLAST is a heuristic that works by finding word-matches between the query and database sequences. You may think of this process as finding "hot-spots" that BLAST can then use to initiate extensions that might lead to full-blown alignments. For nucleotide-nucleotide searches (i.e. "BLASTn") an exact match of the entire word is required before an extension is initiated, so that you normally regulate the sensitivity and speed of the search by increasing or decreasing the wordsize. For other BLAST searches non-exact word matches are taken into account based upon the similarity between words. The amount of similarity can be varied so that you normally uses just the wordsizes 2 and 3 for these searches.
- **Match/mismatch.** A key element in evaluating the quality of a pairwise sequence alignment is the "substitution matrix", which assigns a score for aligning any possible pair of residues. The matrix used in a BLAST search can be changed depending on the type of sequences you are searching with (see the BLAST Frequently Asked Questions). Only applicable for protein sequences or translated DNA sequences.
- **Gap Cost.** The pull down menu shows the Gap Costs (Penalty to open Gap and penalty to extend Gap). Increasing the Gap Costs and Lambda ratio will result in alignments which decrease the number of Gaps introduced.
- **Max number of hit sequences.** The maximum number of database sequences, where BLAST found matches to your query sequence, to be included in the BLAST report.

The parameters you choose will affect how long BLAST takes to run. A search of a small database, requesting only hits that meet stringent criteria will generally be quite quick. Searching large databases, or allowing for very remote matches, will of course take longer.

Click **Finish** to start the tool.

BLAST a partial sequence against NCBI You can search a database using only a part of a sequence directly from the sequence view:

select the sequence region to send to BLAST | right-click the selection | BLAST Selection Against NCBI 

This will go directly to the dialog shown in figure 16.3 and the rest of the options are the same as when performing a BLAST search with a full sequence.

16.1.2 BLAST against local data

Running BLAST searches on your local machine can have several advantages over running the searches remotely at the NCBI:

- It can be faster.
- It does not rely on having a stable internet connection.
- It does not depend on the availability of the NCBI BLAST servers.
- You can use longer query sequences.
- You use your own data sets to search against.

On a technical level, *CLC Genomics Workbench* uses the NCBI's blast+ software (see <ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/>). Thus, the results of using a particular data set to search the same database with the same search parameters would give the same results, whether run locally or at the NCBI.

There are a number of options for what you can search against:

- You can create a database based on data already imported into your Workbench (see section 16.3.3)
- You can add pre-formatted databases (see section 16.3.2)
- You can use sequence data from the **Navigation Area** directly, without creating a database first.

To conduct a local BLAST search, go to:

Tools | BLAST (📄) | BLAST (📄)

This opens the dialog seen in figure 16.5:

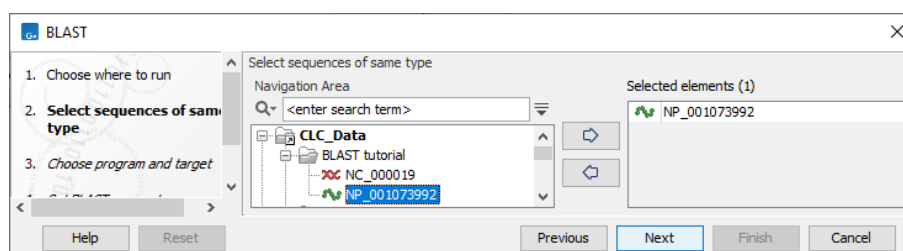


Figure 16.5: Choose one or more sequences to conduct a BLAST search.

Select one or more sequences of the same type (DNA or protein) and click **Next**.

This opens the dialog seen in figure 16.6:

At the top, you can choose between different BLAST programs.

BLAST programs for DNA query sequences:

- **blastn: DNA sequence against a DNA database.** Searches for DNA sequences with homologous regions to your nucleotide query sequence.

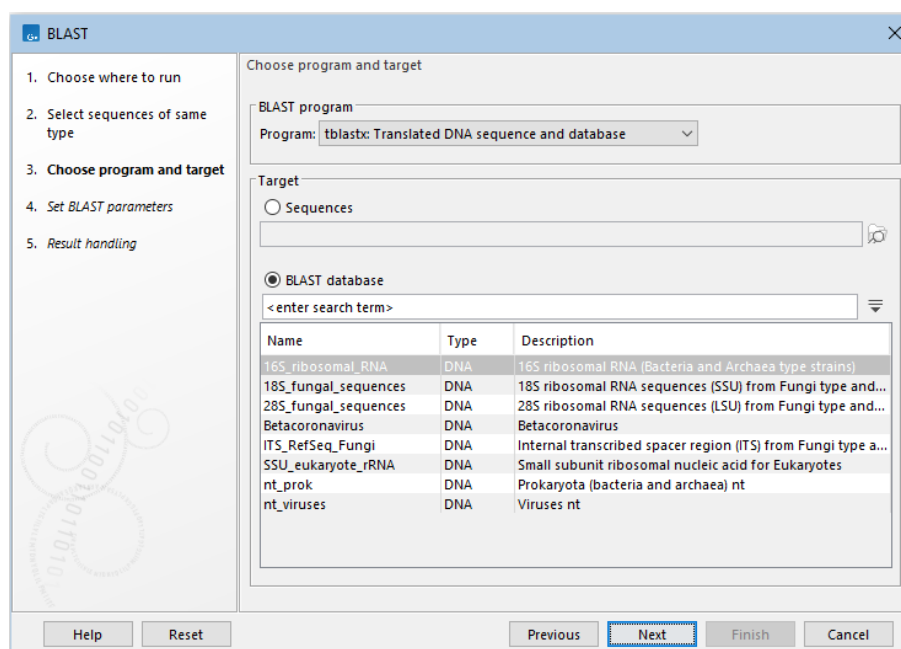


Figure 16.6: Choose a BLAST program and a target database.

- **blastx: Translated DNA sequence against a Protein database.** Automatic translation of your DNA query sequence in six frames; these translated sequences are then used to search a protein database.
- **tblastx: Translated DNA sequence against a Translated DNA database.** Automatic translation of your DNA query sequence and the DNA database, in six frames. The resulting peptide query sequences are used to search the resulting peptide database. Note that this type of search is computationally intensive.

BLAST programs for protein query sequences:

- **blastp: Protein sequence against Protein database.** Used to look for peptide sequences with homologous regions to your peptide query sequence.
- **tblastn: Protein sequence against Translated DNA database.** Peptide query sequences are searched against an automatically translated, in six frames, DNA database.

In cases where you have selected blastx or tblastx to conduct a search, you will get the option of selecting a translation table for the genetic code. The standard genetic code is set as default. This setting is particularly useful when working with organisms or organelles that have a genetic code that differs from the standard genetic code.

If you search against the **Protein Data Bank** database and homologous sequences are found to the query sequence, these can be downloaded and opened with the **3D Molecule Viewer** (see section 17.1.3).

You then specify the target database to use:

- **Sequences.** When you choose this option, you can use sequence data from the **Navigation Area** as database by clicking the **Browse and select** icon (🔍). A temporary BLAST

database will be created from these sequences and used for the BLAST search. It is deleted afterwards. If you want to be able to click in the BLAST result to retrieve the hit sequences from the BLAST database at a later point, you should *not* use this option; create a BLAST database first, see section 16.3.3.

- **BLAST Database.** Select a database already available in one of your designated BLAST database folders. Read more in section 16.4.

When a database or a set of sequences has been selected, click **Next**.

The next dialog allows you to adjust the parameters to meet the requirements of your BLAST search (figure 16.7).

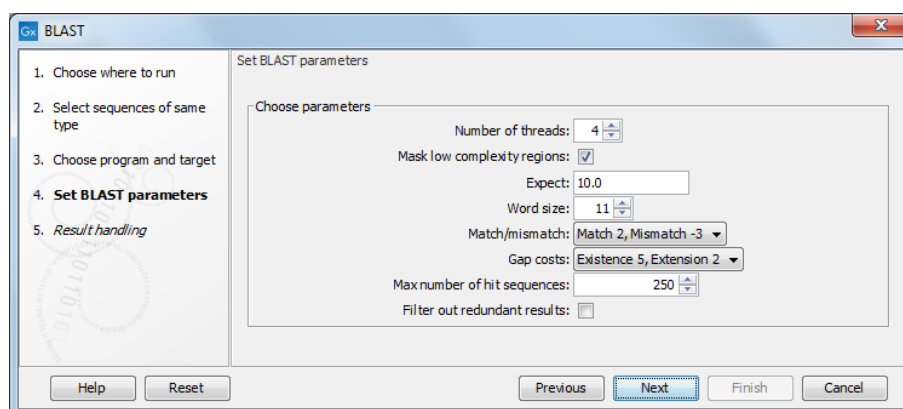


Figure 16.7: Parameters that can be set before submitting a local BLAST search.

- **Number of threads.** You can specify the number of threads, which should be used if your Workbench is installed on a multi-threaded system.
- **Mask low complexity regions.** Mask off segments of the query sequence that have low compositional complexity. Filtering can eliminate statistically significant, but biologically uninteresting reports from the BLAST output (e.g. hits against common acidic-, basic- or proline-rich regions), leaving the more biologically interesting regions of the query sequence available for specific matching against database sequences.
- **Expect.** The threshold for reporting matches against database sequences. The Expect value (E-value) describes the number of hits one can expect to see matching a query by chance when searching against a database of a given size. If the E-value ascribed to a match is greater than the value entered in the Expect field, the match will not be reported. Details of how E-values are calculated can be found at the NCBI: <https://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-1.html>. Lower thresholds are more stringent, leading to fewer chance matches being reported. Increasing the threshold results in more matches being reported, but many may just matching by chance, not due to any biological similarity. Values lower than 1 can be entered as decimals, or in scientific notation. For example, 0.001, 1e-3 and 10e-4 would be equivalent and acceptable values.
- **Word Size.** BLAST is a heuristic that works by finding word-matches between the query and database sequences. You may think of this process as finding "hot-spots" that BLAST can then use to initiate extensions that might lead to full-blown alignments. For nucleotide-nucleotide searches (i.e. "BLASTn") an exact match of the entire word is required before

an extension is initiated, so that you normally regulate the sensitivity and speed of the search by increasing or decreasing the wordsize. For other BLAST searches non-exact word matches are taken into account based upon the similarity between words. The amount of similarity can be varied so that you normally uses just the wordsizes 2 and 3 for these searches.

- **Match/mismatch.** A key element in evaluating the quality of a pairwise sequence alignment is the "substitution matrix", which assigns a score for aligning any possible pair of residues. The matrix used in a BLAST search can be changed depending on the type of sequences you are searching with (see the BLAST Frequently Asked Questions). Only applicable for protein sequences or translated DNA sequences.
- **Gap Cost.** The pull down menu shows the Gap Costs (Penalty to open Gap and penalty to extend Gap). Increasing the Gap Costs and Lambda ratio will result in alignments which decrease the number of Gaps introduced.
- **Max number of hit sequences.** The maximum number of database sequences, where BLAST found matches to your query sequence, to be included in the BLAST report.
- **Filter out redundant results.** This option culls HSPs on a per subject sequence basis by removing HSPs that are completely enveloped by another HSP.

BLAST a partial sequence against a local database You can search a database using only a part of a sequence directly from the sequence view:

select the region that you wish to BLAST | right-click the selection | BLAST Selection Against Local Database ()

This will go directly to the dialog shown in figure 16.6 and the rest of the options are the same as when performing a BLAST search with a full sequence.

16.2 Output from BLAST searches

The output of a BLAST search is similar whether you have chosen to run your search locally or at the NCBI.

If a **single query** sequence was used, then the results will show the hits and High-Scoring Segment Pairs (HSPs) found in that database with that single sequence. If **more than one query** sequence was used, the default view of the results is a summary table, where the description of the top match found for each query sequence and the number of matches found is reported. The summary table is described in detail in section 16.2.2.

16.2.1 Graphical overview for each query sequence

Double clicking on a given row of a tabular blast table opens a graphical overview of the blast results for a particular query sequence, as shown in figure figure 16.8. In cases where only one sequence was entered into a BLAST search, such a graphical overview is the default output.

Figure 16.8 shows an example of a BLAST result for an individual query sequence in the *CLC Genomics Workbench*.

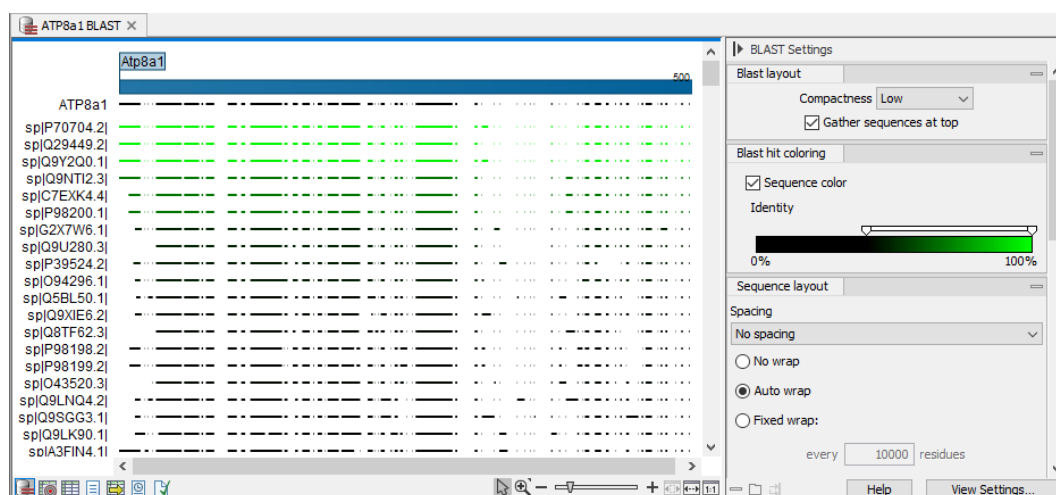


Figure 16.8: Default display of the output of a BLAST search for one query sequence. At the top is there a graphical representation of BLAST hits with tooltips showing additional information on individual hits.

Detailed descriptions of the overview BLAST table and the graphical BLAST results view are described below.

16.2.2 Overview BLAST table

In the overview BLAST table for a multi-sequence blast search, as shown in figure 16.9, there is one row for each query sequence. Each row represents the BLAST result for this query sequence.

Rows: 2				Filter
Query	Number of hits	Lowest E-value	Accession (E-value)	
ATP8a1 genomic sequence	624	0.00	ATP8a1_genomic_sequence	
ATP8a1 mRNA	50	0.00	ATP8a1_genomic_sequence	

Open BLAST Output Extract Consensus Open Query Sequence

Figure 16.9: An overview BLAST table summarizing the results for a number of query sequences.

Double-clicking a row will open the BLAST result for this query sequence, allowing more detailed investigation of the result. You can also select one or more rows and click the **Open BLAST Output** button at the bottom of the view. Consensus sequence can be extracted by clicking the **Extract Consensus** button at the bottom. Clicking the **Open Query Sequence** will open a sequence list with the selected query sequences. This can be useful in work flows where BLAST is used as a filtering mechanism where you can filter the table to include e.g. sequences that have a certain top hit and then extract those.

In the overview table, the following information is shown:

- **Query:** Since this table displays information about several query sequences, the first column is the name of the query sequence.

- Number of HSPs: The number of High-scoring Segment Pairs (HSPs) for this query sequence.
- For the following list, the value of the best HSP is displayed together with accession number and description of this HSP, with respect to E-value, identity or positive value, hit length or bit score.
 - Lowest E-value
 - Accession (E-value)
 - Description (E-value)
 - Greatest identity %
 - Accession (identity %)
 - Description (identity %)
 - Greatest positive %
 - Accession (positive %)
 - Description (positive %)
 - Greatest HSPs length
 - Accession (HSP length)
 - Description (HSP length)
 - Greatest bit score
 - Accession (bit score)
 - Description (bit score)

If you wish to save some of the BLAST results as individual elements in the **Navigation Area**, open them and click **Save As** in the **File** menu.

16.2.3 BLAST graphics

The **BLAST editor** shows the sequences hits which were found in the BLAST search. The hit sequences are represented by colored horizontal lines, and when hovering the mouse pointer over a BLAST hit sequence, a tooltip appears, listing the characteristics of the sequence. As default, the query sequence is fitted to the window width, but it is possible to zoom in the windows and see the actual sequence alignments returned from the BLAST server.

There are several settings available in the **BLAST Settings** side panel.

- **Blast layout.** You can control the level of **Compactness** for displaying sequences:
 - **Not compact.** Full detail and spaces between the sequences.
 - **Low.** The normal settings where the residues are visible (when zoomed in) but with no extra spaces between.
 - **Medium.** The sequences are represented as lines and the residues are not visible. There is some space between the sequences.
 - **Compact.** Even less space between the sequences.

You can also choose to **Gather sequences at top**. Enabling this option affects the view that is shown when scrolling horizontally along a BLAST result. If selected, the sequence hits which did not contribute to the visible part of the BLAST graphics will be omitted whereas the found BLAST hits will automatically be placed right below the query sequence.

- **BLAST hit coloring.** You can choose whether to color hit sequences and adjust the coloring scale for visualisation of identity level.

The remaining View preferences for BLAST Graphics are the same as those of alignments. See section 15.2.

Some of the information available in the tooltips when hovering over a particular hit sequence is:

- **Name of sequence.** Here is shown some additional information of the sequence which was found. This line corresponds to the description line in GenBank (if the search was conducted on the nr database).
- **Score.** This shows the bit score of the local alignment generated through the BLAST search.
- **Expect.** Also known as the E-value. A low value indicates a homologous sequence. Higher E-values indicate that BLAST found a less homologous sequence.
- **Identities.** This number shows the number of identical residues or nucleotides in the obtained alignment.
- **Gaps.** This number shows whether the alignment has gaps or not.
- **Strand.** This is only valid for nucleotide sequences and show the direction of the aligned strands. Minus indicate a complementary strand.

The numbers of the query and subject sequences refer to the sequence positions in the submitted and found sequences. If the subject sequence has number 59 in front of the sequence, this means that 58 residues are found upstream of this position, but these are not included in the alignment.

By right clicking the sequence name in the Graphical BLAST output it is possible to download the full hits sequence from NCBI with accompanying annotations and information. It is also possible to just open the actual hit sequence in a new view.

16.2.4 BLAST HSP table

In addition to the graphical display of a BLAST result, it is possible to view the BLAST results in a tabular view. In the tabular view, one can get a quick and fast overview of the results. Here you can also select multiple sequences and download or open all of these in one single step. Moreover, there is a link from each sequence to the sequence at NCBI. These possibilities are either available through a right-click with the mouse or by using the buttons below the table.

The **BLAST table** view can be shown in the following way:

Click the Show BLAST HSP Table button  at the bottom of the view

Figure 16.10 is an example of a BLAST HSP Table.

The BLAST HSP Table includes the following information:

Hit	Description	E-value	Score	%Gaps
XM_008123010	PREDICTED: Anolis carolinensis bone morphogenetic protein 4 (bmp4), mRNA	2.76E-167	662.00	11.99
XM_007436872	PREDICTED: Python bivittatus bone morphogenetic protein 4-like (LOC103054912), mRNA	3.16E-46	216.00	0.00
XM_007436872	PREDICTED: Python bivittatus bone morphogenetic protein 4-like (LOC103054912), mRNA	6.53E-17	108.00	2.03
XM_005295477	PREDICTED: Chrysemys picta bellii bone morphogenetic protein 4 (BMP4), mRNA	3.16E-46	216.00	5.38
XM_010712170	PREDICTED: Meleagris gallopavo bone morphogenetic protein 4 (BMP4), mRNA	1.10E-45	214.00	6.38
NM_205237	Gallus gallus bone morphogenetic protein 4 (BMP4), mRNA	1.10E-45	214.00	6.38
AC172371	Gallus gallus BAC clone CH261-20124 from chromosome ul, complete sequence	1.10E-45	214.00	6.38
BX932038	Gallus gallus finished cDNA, clone CHEST895j20	1.10E-45	214.00	6.38
XM_005029381	PREDICTED: Anas platyrhynchos bone morphogenetic protein 4 (BMP4), mRNA	3.86E-45	212.00	7.58
EF540749	Anas platyrhynchos bone morphogenetic protein 4 (BMP4) mRNA, complete cds	1.35E-44	210.00	6.11
XM_005495201	PREDICTED: Zonotrichia albicollis bone morphogenetic protein 4 (BMP4), mRNA	3.61E-39	190.00	7.85

Figure 16.10: *BLAST HSP Table*. The HSPs can be sorted by the different columns, simply by clicking the column heading.

- **Query sequence.** The sequence which was used for the search.
- **HSP.** The Name of the sequences found in the BLAST search.
- **Id.** GenBank ID.
- **Description.** Text from NCBI describing the sequence.
- **E-value.** Measure of quality of the match. Higher E-values indicate that BLAST found a less homologous sequence.
- **Score.** This shows the score of the local alignment generated through the BLAST search.
- **Bit score.** This shows the bit score of the local alignment generated through the BLAST search. Bit scores are normalized, which means that the bit scores from different alignments can be compared, even if different scoring matrices have been used.
- **HSP start.** Shows the start position in the HSP sequence.
- **HSP end.** Shows the end position in the HSP sequence.
- **HSP length.** The length of the HSP.
- **Query start.** Shows the start position in the query sequence.
- **Query end.** Shows the end position in the query sequence.
- **Overlap.** Display a percentage value for the overlap of the query sequence and HSP sequence. Only the length of the local alignment is taken into account and not the full length query sequence.
- **Identity.** Shows the number of identical residues in the query and HSP sequence.
- **%Identity.** Shows the percentage of identical residues in the query and HSP sequence.
- **Positive.** Shows the number of similar but not necessarily identical residues in the query and HSP sequence.
- **%Positive.** Shows the percentage of similar but not necessarily identical residues in the query and HSP sequence.
- **Gaps.** Shows the number of gaps in the query and HSP sequence.

- **%Gaps.** Shows the percentage of gaps in the query and HSP sequence.
- **Query Frame/Strand.** Shows the frame or strand of the query sequence.
- **HSP Frame/Strand.** Shows the frame or strand of the HSP sequence.

In the **BLAST table** view you can handle the HSP sequences. Select one or more sequences from the table, and apply one of the following functions.

- **Download and Open.** Download the full sequence from NCBI and opens it. If multiple sequences are selected, they will all open (if the same sequence is listed several times, only one copy of the sequence is downloaded and opened).
- **Download and Save.** Download the full sequence from NCBI and save it. When you click the button, there will be a save dialog letting you specify a folder to save the sequences. If multiple sequences are selected, they will all open (if the same sequence is listed several times, only one copy of the sequence is downloaded and opened).
- **Open at NCBI.** Opens the corresponding sequence(s) at GenBank at NCBI. Here is stored additional information regarding the selected sequence(s). The default Internet browser is used for this purpose.
- **Open structure.** If the HSP sequence contain structure information, the sequence is opened in a text view or a 3D view. Note that the 3D view has special system requirements, see section 1.3.

The HSPs can be sorted by the different columns, simply by clicking the column heading. In cases where individual rows have been selected in the table, the selected rows will still be selected after sorting the data.

You can do a text-based search in the information in the BLAST table by using the filter at the upper right part of the view. In this way you can search for e.g. species or other information which is typically included in the "Description" field.

The table is integrated with the graphical view described in section 16.2.3 so that selecting a HSP in the table will make a selection on the corresponding sequence in the graphical view.

16.2.5 BLAST hit table

The **BLAST Hit table** view can be shown in the following way:

Click the Show BLAST Hit Table button  at the bottom of the view

Figure 16.11 is an example of a BLAST Hit Table.

The BLAST Hit Table includes the following information:

- **Query sequence.** The sequence which was used for the search.
- **Hit.** The Name of the sequences found in the BLAST search.
- **Id.** GenBank ID.

Hit	Description	Total score	Max score	Min E-value	Max %Ident...
XM_008123010	PREDICTED: Anolis carolinensis bone morphogenetic protein 4 (bmp4), mRNA	662.00	662.00	2.76E-167	88.01
XM_007436872	PREDICTED: Python bivittatus bone morphogenetic protein 4-like (LOC103054912), mRNA	324.00	216.00	3.16E-46	79.82
XM_005295477	PREDICTED: Chrysemys picta bellii bone morphogenetic protein 4 (BMP4), mRNA	216.00	216.00	3.16E-46	71.54
XM_010712170	PREDICTED: Meleagris gallopavo bone morphogenetic protein 4 (BMP4), mRNA	214.00	214.00	1.10E-45	71.68
NM_205237	Gallus gallus bone morphogenetic protein 4 (BMP4), mRNA	214.00	214.00	1.10E-45	71.68
AC172371	Gallus gallus BAC clone CH261-20124 from chromosome ul, complete sequence	214.00	214.00	1.10E-45	71.68
BX932038	Gallus gallus finished cDNA, clone CHEST895J20	214.00	214.00	1.10E-45	71.68
XM_005029381	PREDICTED: Anas platyrhynchos bone morphogenetic protein 4 (BMP4), mRNA	212.00	212.00	3.86E-45	71.97
EF540749	Anas platyrhynchos bone morphogenetic protein 4 (BMP4) mRNA, complete cds	210.00	210.00	1.35E-44	71.76
EF540750	Anas platyrhynchos bone morphogenetic protein 4 (BMP4) gene, complete cds	206.00	206.00	1.64E-43	71.72
X75915	G.gallus mRNA for bone morphogenetic protein 4	198.00	198.00	2.43E-41	70.89
XM_002200411	PREDICTED: Taeniopygia guttata bone morphogenetic protein 4 (BMP4), mRNA	196.00	196.00	8.49E-41	70.89
XM_009986929	PREDICTED: Tauraco erythrolophus bone morphogenetic protein 4 (BMP4), partial mRNA	196.00	196.00	8.49E-41	70.89
XM_009709165	PREDICTED: Cariamia cristata bone morphogenetic protein 4 (BMP4), partial mRNA	196.00	196.00	8.49E-41	71.14
XM_008919900	PREDICTED: Manacus vitellinus bone morphogenetic protein 4 (BMP4), mRNA	196.00	196.00	8.49E-41	70.89
XM_007072683	PREDICTED: Chelonia mydas bone morphogenetic protein 4 (BMP4), mRNA	196.00	196.00	8.49E-41	70.51
XM_006115854	PREDICTED: Pelodiscus sinensis bone morphogenetic protein 4 (BMP4), partial mRNA	196.00	196.00	8.49E-41	71.09
XM_005437901	PREDICTED: Falco cherrug bone morphogenetic protein 4 (BMP4), mRNA	196.00	196.00	8.49E-41	70.89

Figure 16.11: *BLAST Hit Table*. The hits can be sorted by the different columns, simply by clicking the column heading.

- **Description.** Text from NCBI describing the sequence.
- **Total Score.** Total score for all HSPs.
- **Max Score.** Maximum score of all HSPs.
- **Min E-value.** Minimum e-value of all HSPs.
- **Max Bit score.** Maximum Bit score of all HSPs.
- **Max Identity.** Shows the maximum number of identical residues in the query and Hit sequence.
- **Max %Identity.** Shows the percentage of maximum identical residues in the query and Hit sequence.
- **Max Positive.** Shows the maximum number of similar but not necessarily identical residues in the query and Hit sequence.
- **Max %Positive.** Shows the percentage of maximum similar but not necessarily identical residues in the query and Hit sequence.

16.2.6 Extracting a consensus sequence from a BLAST result

A consensus sequence can be extracted from nucleotide BLAST results, as described in section 22.10. That section focuses on working with read mappings, but the same underlying tool is used to extract consensus sequences from nucleotide BLAST results.

16.3 Local BLAST databases

Databases can be made available for **BLAST** searches in several ways:

- Download pre-formatted BLAST databases from the NCBI using **Download BLAST Databases** (see section 16.3.1).

- Specify locations where BLAST databases are stored at your site using **Manage Blast Databases** (see section 16.4).
- Use **Create BLAST Database** to create databases using sequences or sequence lists selected from the Workbench Navigation Area (see section 16.3.3).

For BLAST searches against a small amount of sequence data, a sequence list can be specified instead of a database when launching the **BLAST** tool (see section 16.1.2). A database for those sequences will be created as part of that job. This adds to the overall execution time, so if those sequences will be used for multiple searches, creating a BLAST database and referring to that when launching searches is likely to be preferable.

16.3.1 Download NCBI pre-formatted BLAST databases

Using **Download BLAST Databases**, many pre-formatted BLAST databases can be downloaded from the NCBI. The source of these databases is <https://ftp.ncbi.nlm.nih.gov/blast/db/>.

Your Workbench must be connected to the internet to use this tool.

To download a BLAST database from the NCBI, go to:

Tools | BLAST (📁) | Download BLAST Databases (🌐)

A dialog listing the available databases will open (figure 16.12). The date each database was made available for download on the NCBI site, the size of the files associated with that database, and a brief description of each database is provided.

Note: Many of the databases listed are very large. If you are working on a shared system, we recommend discussing your download plans with others that make use of, or who administer, that system.

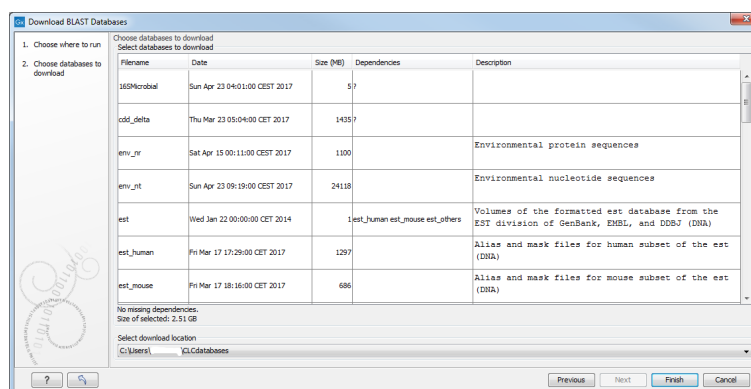


Figure 16.12: Choose from pre-formatted BLAST databases at the NCBI available for download.

If there is more than one BLAST database location configured, you will be able to specify which one to store the BLAST database files in. See section 16.4 for details about adding BLAST database locations.

16.3.2 Make pre-formatted BLAST databases available

To use databases that have been downloaded or created outside the Workbench, you can either:

- Add the location where BLAST database files are stored as a BLAST database location (see section 16.4).
- OR
- Move the files that make up the BLAST database to a location already configured as a BLAST database location. All the files that comprise a given BLAST database must be moved. This may be as few as three files, but can be more (figure 16.13).

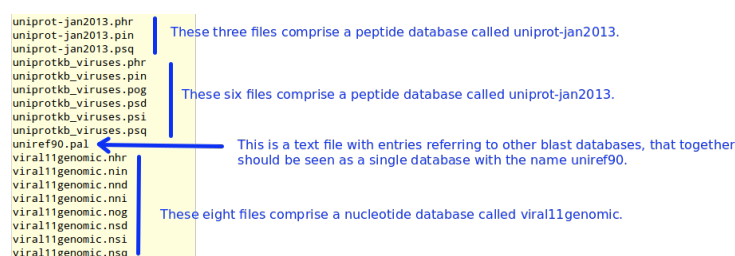


Figure 16.13: BLAST databases are made up of several files. The exact number varies. Large databases will be split into the number of volumes and there will be several files per volume.

16.3.3 Create local BLAST databases

You can create a BLAST database from DNA, RNA, or protein sequences to use in local BLAST searches.

The BLAST database files will be saved to an area that is specified as a BLAST database location (see section 16.4). The BLAST databases found in these locations are listed when launching the **BLAST** tool (see section 16.1.2).

Making pre-existing BLAST databases available for searching using your Workbench is described in section 16.3.2.

To create a BLAST database, go to:

Tools | BLAST (📁) | **Create BLAST Database** (🛠️)

This opens the dialog seen in figure 16.14.

After selecting the sequences or sequence lists to include in your database and clicking on **Next**, you provide information about the BLAST database being made figure 16.15:

- **Name.** The name of the BLAST database. This name will be used when running BLAST searches and also as the base file name for the BLAST database files.
- **Description.** A short description. This is displayed along with the database name in the list of available databases when launching a local BLAST search. If no description is entered, the creation date is used as the description.
- **Location.** The BLAST database location to save the BLAST database files to.

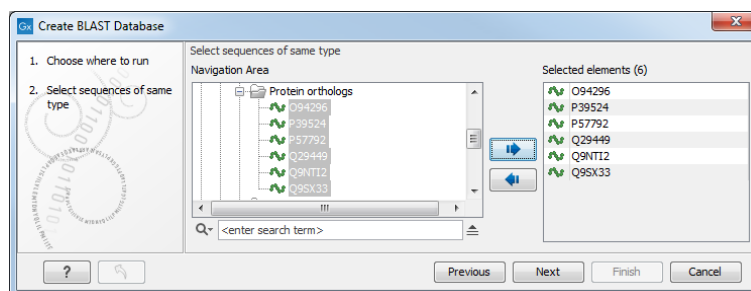


Figure 16.14: Select sequences to be used in the creation of a BLAST database.

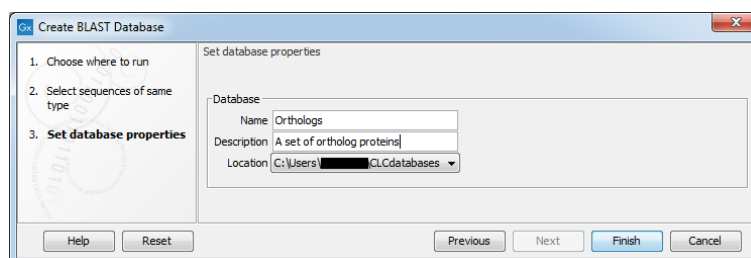


Figure 16.15: Provide information about the BLAST database being created and specify where the files should be saved to.

Click **Finish** to create the BLAST database. Once the process is complete, the new database will be available in the **Manage BLAST Databases** dialog, see section 16.4, and when launching IBLAST jobs (see section 16.1.2).

Create BLAST Database creates BLAST+ version 4 (dbV4) databases.

Sequence identifiers and BLAST databases

Restrictions on sequence identifier lengths, format, and duplicates present in the underlying BLAST+ program for making databases, *makeblastdb*, do not apply when making databases using **Create BLAST Database**.

Internal handling of sequence names, introduced in version 21.0, allows this level of naming flexibility with newer versions of BLAST+. There should be no obvious effects of this internal handling of sequence names on local **BLAST** search results, including the names written to BLAST reports. For further details, see the FAQ **Can I search BLAST databases made using Create BLAST Databases with BLAST+ software?** at <https://qiagen.my.salesforce-sites.com/KnowledgeBase/KnowledgeNavigatorPage?id=kA41i000000CjJ9CAK>

16.4 Manage BLAST databases

The BLAST databases available to search using the **BLAST** tool can be managed using **Manage BLAST Databases** (figure 16.16), launched by going to:

Tools | BLAST () | Manage BLAST Databases ()

At the top of the dialog, there is a list of the **BLAST database locations**. These locations are folders where the Workbench will look for valid BLAST databases. These can either be created

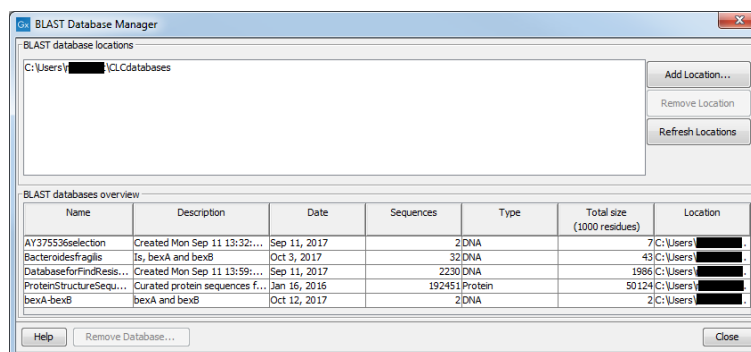


Figure 16.16: BLAST databases are listed and can be managed using Manage BLAST Databases.

from within the Workbench using the **Create BLAST Database tool**, see section 16.3.3, or they can be pre-formatted BLAST databases.

The list of locations can be modified using the **Add Location** and **Remove Location** buttons. Once the Workbench has scanned the locations, it will keep a cache of the databases (in order to improve performance). If you have added new databases that are not listed, you can press **Refresh Locations** to clear the cache and search the database locations again.

By default a BLAST database location will be added under your home area in a folder called CLCdatabases. This folder is scanned recursively, through all subfolders, to look for valid databases. All other folder locations are scanned only at the top level.

Below the list of locations, all the BLAST databases are listed with the following information:

- **Name.** The name of the BLAST database.
- **Description.** Detailed description of the contents of the database.
- **Date.** The date the database was created.
- **Sequences.** The number of sequences in the database.
- **Type.** The type can be either nucleotide (DNA) or protein.
- **Total size (1000 residues).** The number of residues in the database, either bases or amino acid.
- **Location.** The location of the database.

Below the list of BLAST databases, there is a button to **Remove Database**. This option will delete the database files belonging to the database selected.

16.5 Bioinformatics explained: BLAST

BLAST (Basic Local Alignment Search Tool) has become the *defacto* standard in search and alignment tools [Altschul et al., 1990]. The BLAST algorithm is still actively being developed and is one of the most cited papers ever written in this field of biology. Many researchers use BLAST as an initial screening of their sequence data from the laboratory and to get an idea of what they are working on. BLAST is far from being basic as the name indicates; it is a highly advanced

algorithm which has become very popular due to availability, speed, and accuracy. In short, BLAST search programs look for potentially homologous sequences to your query sequences in databases, either locally held databases or those hosted elsewhere, such as at the NCBI (<http://www.ncbi.nlm.nih.gov/>) [McGinnis and Madden, 2004].

BLAST can be used for a lot of different purposes. Some of the most popular purposes are listed on the BLAST webpage at the NCBI: <https://blast.ncbi.nlm.nih.gov/Blast.cgi>.

Searching for homology Most research projects involving sequencing of either DNA or protein have a requirement for obtaining biological information of the newly sequenced and maybe unknown sequence. If the researchers have no prior information of the sequence and biological content, valuable information can often be obtained using BLAST. The BLAST algorithm will search for homologous sequences in predefined and annotated databases of the users choice.

In an easy and fast way the researcher can gain knowledge of gene or protein function and find evolutionary relations between the newly sequenced DNA and well established data.

A BLAST search generates a report specifying the potentially homologous sequences found and their local alignments with the query sequence.

16.5.1 How does BLAST work?

BLAST identifies homologous sequences using a heuristic method which initially finds short matches between two sequences. After finding initial matches, BLAST attempts to build local alignments with the query sequence using these. Thus, BLAST does not guarantee the optimal alignment and some sequence hits may be missed. To find optimal alignments, the Smith-Waterman algorithm should be used (see below). Below, the BLAST algorithm is described in more detail.

Seeding When finding a match between a query sequence and a hit sequence, the starting point is the *words* that the two sequences have in common. A word is simply defined as a number of letters. For blastp the default word size is 3 $W=3$. If a query sequence has a QWRTG, the searched words are QWR, WRT, RTG. See figure 16.17 for an illustration of words in a protein sequence.



Figure 16.17: Generation of exact BLAST words with a word size of $W=3$.

During the initial BLAST seeding, the algorithm finds all common words between the query sequence and the hit sequence(s). Only regions with a word hit will be used to build on an alignment.

BLAST will start out by making words for the entire query sequence (see figure 16.17). For each word in the query sequence, a compilation of neighborhood words, which exceed the threshold

of T , is also generated.

A neighborhood word is a word obtaining a score of at least T when comparing, using a selected scoring matrix (see figure 16.18). The default scoring matrix for blastp is BLOSUM62. The compilation of exact words and neighborhood words is then used to match against the database sequences.



Figure 16.18: Neighborhood BLAST words based on the BLOSUM62 matrix. Only words where the threshold T exceeds 13 are included in the initial seeding.

After the initial finding of words (seeding), the BLAST algorithm will extend the (only 3 residues long) alignment in both directions (see figure 16.19). Each time the alignment is extended, an alignment score is increases/decreased. When the alignment score drops below a predefined threshold, the extension of the alignment stops. This ensures that the alignment is not extended to regions where only very poor alignment between the query and hit sequence is possible. If the obtained alignment receives a score above a certain threshold, it will be included in the final BLAST result.

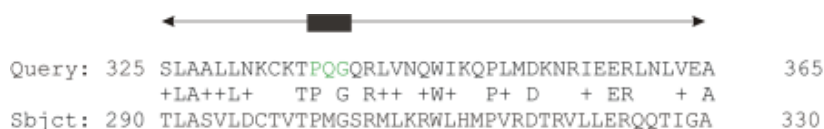


Figure 16.19: Blast aligning in both directions. The initial word match is marked green.

By tweaking the word size W and the neighborhood word threshold T , it is possible to limit the search space. E.g. by increasing T , the number of neighboring words will drop and thus limit the search space as shown in figure 16.20.

This will increase the speed of BLAST significantly but may result in loss of sensitivity. Increasing the word size W will also increase the speed but again with a loss of sensitivity.

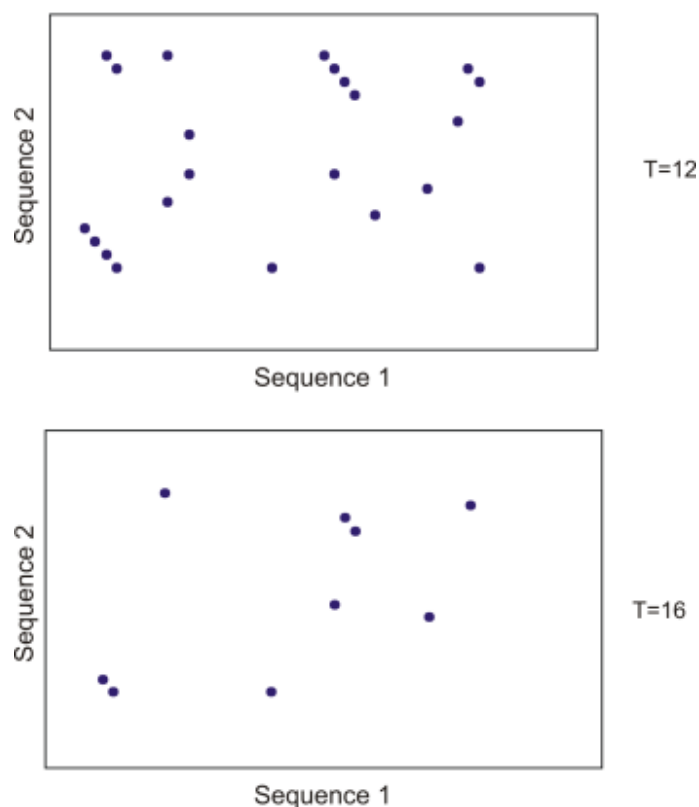


Figure 16.20: Each dot represents a word match. Increasing the threshold of T limits the search space significantly.

16.5.2 Which BLAST program should I use?

Depending on the nature of the sequence it is possible to use different BLAST programs for the database search. There are five versions of the BLAST program, blastn, blastp, blastx, tblastn, tblastx included in CLC software:

Option	Query Type	DB Type	Comparison	Note
blastn	Nucleotide	Nucleotide	Nucleotide-Nucleotide	
blastp	Protein	Protein	Protein-Protein	
tblastn	Protein	Nucleotide	Protein-Protein	The database is translated into protein
blastx	Nucleotide	Protein	Protein-Protein	The queries are translated into protein
tblastx	Nucleotide	Nucleotide	Protein-Protein	The queries and database are translated into protein

The most commonly used method is to BLAST a nucleotide sequence against a nucleotide database (blastn) or a protein sequence against a protein database (blastp). But often another BLAST program will produce more interesting hits. E.g. if a nucleotide sequence is translated before the search, it is more likely to find better and more accurate hits than just a blastn search. One of the reasons for this is that protein sequences are evolutionarily more conserved than nucleotide sequences. Another good reason for translating the query sequence before the search is that you get protein hits which are likely to be annotated. Thus you can directly see the protein function of the sequenced gene.

16.5.3 Which BLAST options should I change?

There are a number of options that can be configured when using BLAST search programs. Setting these options to relevant values can have a great impact on the search result. A few of the key settings are described briefly below.

The E-value The *expect value* (E-value) describes the number of hits one can expect to see matching the query by chance when searching against a database of a given size. An E-value of 1 can be interpreted as meaning that in a search like the one just run, you could expect to see 1 match of the same score by chance once. That is, a match that is not homologous to the query sequence. When looking for very similar sequences in a database, it is often beneficial to use very low E-values.

E-values depend on the query sequence length and the database size. Short identical sequence may have a high E-value and may be regarded as "false positive" hits. This is often seen if one searches for short primer regions, small domain regions etc. Below are some comments on what one could infer from results with E-values in particular ranges.

- **E-value < 10e-100** Identical sequences. You will get long alignments across the entire query and hit sequence.
- **10e-100 < E-value < 10e-50** Almost identical sequences. A long stretch of the query matches the hit sequence.
- **10e-50 < E-value < 10e-10** Closely related sequences, could be a domain match or similar.
- **10e-10 < E-value < 1** Could be a true homolog, but it is a gray area.
- **E-value > 1** Proteins are most likely not related
- **E-value > 10** Hits are most likely not related unless the query sequence is very short.

Gap costs For blastp it is possible to specify gap cost for the chosen substitution matrix. There is only a limited number of options for these parameters. The *open gap cost* is the price of introducing gaps in the alignment, and *extension gap cost* is the price of every extension past the initial opening gap. Increasing the gap costs will result in alignments with fewer gaps.

Filters It is possible to set different filter options before running a BLAST search. Low-complexity regions have a very simple composition compared to the rest of the sequence and may result in problems during the BLAST search [Wootton and Federhen, 1993]. A low complexity region of a protein can for example look like this 'fftflllsss', which in this case is a region as part of a signal peptide. In the output of the BLAST search, low-complexity regions will be marked in lowercase gray characters (default setting). The low complexity region cannot be thought of as a significant match; thus, disabling the low complexity filter is likely to generate more hits to sequences which are not truly related.

Word size Changing the word size has a great impact on the seeded sequence space as described above. But one can change the word size to find sequence matches which would otherwise not be found using the default parameters. For instance the word size can be

decreased when searching for primers or short nucleotides. For `blastn` a suitable setting would be to decrease the default word size of 11 to 7, increase the E-value significantly (1000) and turn off the complexity filtering.

For `blastp` a similar approach can be used. Decrease the word size to 2, increase the E-value and use a more stringent substitution matrix, e.g. a PAM30 matrix.

The BLAST search programs at the NCBI adjust settings automatically when short sequences are being used for searches, and there is a dedicated page, Primer-BLAST, for searching for primer sequences. <https://blast.ncbi.nlm.nih.gov/Blast.cgi>.

Substitution matrix For protein BLAST searches, a default substitution matrix is provided. If you are looking at distantly related proteins, you should either choose a high-numbered PAM matrix or a low-numbered BLOSUM matrix. The default scoring matrix for `blastp` is BLOSUM62.

16.5.4 Where can I get the BLAST+ programs

The BLAST+ package can be downloaded for use on your own computer, institution computer cluster or similar from <ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/>. It is available for a wide range of different operating systems.

Pre-formatted databases are available from a dedicated BLAST ftp site <ftp://ftp.ncbi.nlm.nih.gov/blast/db/>. Most BLAST databases on the NCBI site are updated on a daily basis.

A few commercial software packages are available for searching your own data. The advantage of using a commercial program is obvious when BLAST is integrated with the existing tools of these programs. Furthermore, they let you perform BLAST searches and retain annotations on the query sequence (see figure 16.21). It is also much easier to batch download a selection of hit sequences for further inspection.

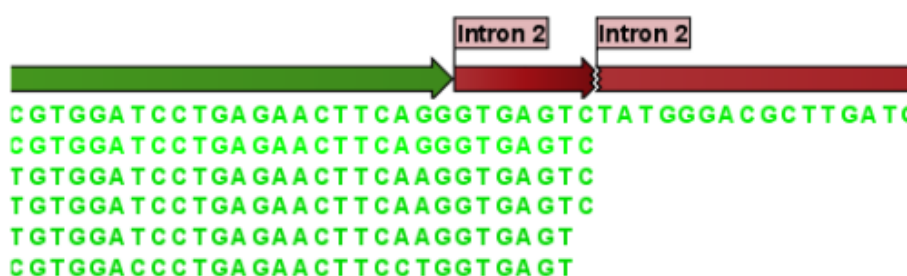


Figure 16.21: Snippet of alignment view of BLAST results. Individual alignments are represented directly in a graphical view. The top sequence is the query sequence and is shown with a selection of annotations.

16.5.5 What you cannot get out of BLAST

Don't expect BLAST to produce the best available alignment. BLAST is a heuristic method which does not guarantee the best results, and therefore you cannot rely on BLAST if you wish to find all the hits in the database.

Instead, use the Smith-Waterman algorithm for obtaining the best possible local alignments [Smith and Waterman, 1981].

BLAST only makes local alignments. This means that a great but short hit in another sequence may not at all be related to the query sequence even though the sequences align well in a small region. It may be a domain or similar.

It is always a good idea to be cautious of the material in the database. For instance, the sequences may be wrongly annotated; hypothetical proteins are often simple translations of a found ORF on a sequenced nucleotide sequence and may not represent a true protein.

Don't expect to see the best result using the default settings. As described above, the settings should be adjusted according to the what kind of query sequence is used, and what kind of results you want. It is a good idea to perform the same BLAST search with different settings to get an idea of how they work. There is not a final answer on how to adjust the settings for your particular sequence.

16.5.6 Other useful resources

The NCBI BLAST web page

<https://blast.ncbi.nlm.nih.gov/Blast.cgi>

The latest BLAST+ release

<ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST>

Download pages for pre-formatted BLAST databases

<ftp://ftp.ncbi.nlm.nih.gov/blast/db/>

O'Reilly book on BLAST

<http://www.oreilly.com/catalog/blast/>

Chapter 17

3D Molecule Viewer

Contents

17.1 Importing molecule structure files	437
17.1.1 From the Protein Data Bank	438
17.1.2 From your own file system	438
17.1.3 BLAST search against the PDB database	439
17.1.4 Import issues	441
17.2 Viewing molecular structures in 3D	441
17.3 Customizing the visualization	442
17.3.1 Visualization styles and colors	443
17.3.2 Project settings	449
17.4 Tools for linking sequence and structure	451
17.4.1 Show sequence associated with molecule	452
17.4.2 Link sequence or sequence alignment to structure	452
17.4.3 Transfer annotations between sequence and structure	453
17.5 Align Protein Structure	455
17.5.1 Example: alignment of calmodulin	456
17.5.2 The Align Protein Structure algorithm	460
17.6 Generate Biomolecule	460

Proteins are amino acid polymers that are involved in all aspects of cellular function. The structure of a protein is defined by its particular amino acid sequence, with the amino acid sequence being referred to as the primary protein structure. The amino acids fold up in local structural elements; helices and sheets, also called the secondary structure of the protein. These structural elements are then packed into globular folds, known as the tertiary structure or the three dimensional structure.

In order to understand protein function it is often valuable to see the three dimensional structure of the protein. This is possible when the structure of the protein has been resolved and published. Structure files are usually deposited in the Protein Data Bank (PDB) <https://www.rcsb.org/>, where the publicly available protein structure files can be searched and downloaded. The vast majority of the protein structures have been determined by X-ray crystallography (88%) while the rest of the structures predominantly have been obtained by Nuclear Magnetic Resonance techniques.

In addition to protein structures, the PDB entries also contain structural information about molecules that interact with the protein, such as nucleic acids, ligands, cofactors, and water. There are also entries, which contain nucleic acids and no protein structure. The **3D Molecule Viewer** in the *CLC Genomics Workbench* is an integrated viewer of such structure files.

If you have problems viewing 3D structures, please check your system matches the requirements for 3D viewers. See section [1.3](#).

The **3D Molecule Viewer** offers a range of tools for inspection and visualization of molecular structures:

- Automatic sorting of molecules into categories: Proteins, Nucleic acids, Ligands, Cofactors, Water molecules
- Hide/unhide individual molecules from the view
- Four different atom-based molecule visualizations
- Backbone visualization for proteins and nucleic acids
- Molecular surface visualization
- Selection of different color schemes for each molecule visualization
- Customized visualization for user selected atoms
- Alignment of protein structures
- Browse amino acids and nucleic acids from sequence editors started from within the 3D Molecule Viewer
- Link a sequence or alignment to a protein structure
- Transfer annotations between the linked sequence and the structure

17.1 Importing molecule structure files

The supported file format for three dimensional protein structures in the **3D Molecule Viewer** is the Protein Data Bank (PDB) format, which upon import is converted to a CLC Molecule Project. PDB files can be imported to a Molecule Project in three different ways:

- from the Protein Data Bank
- from your own file system
- using BLAST search against the PDB database

17.1.1 From the Protein Data Bank

Molecule structures can be imported in the workbench from the Protein Data Bank using the "Download" function:

Toolbar | Download (📄) | Search for PDB structures at NCBI (🔍)

Type the molecule name or accession number into the search field and click on the "Start search" button (as shown in figure 17.1). The search hits will appear in the table below the search field.

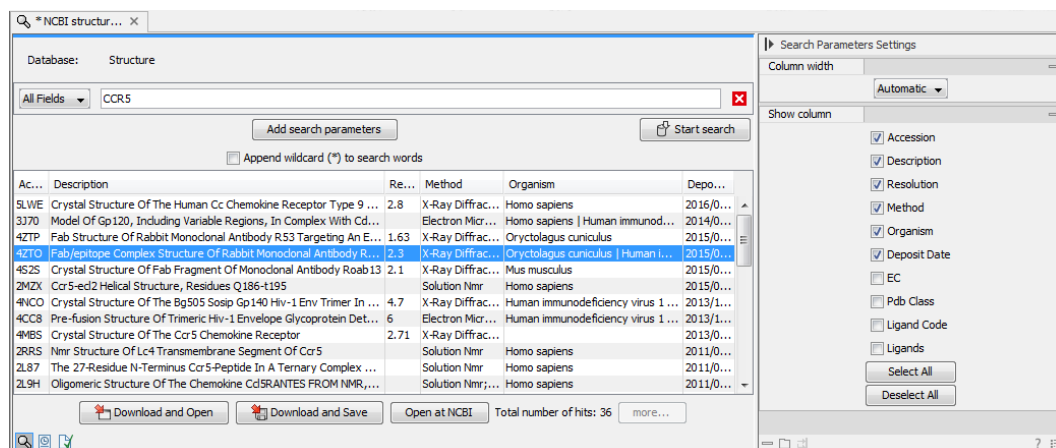


Figure 17.1: Download protein structure from the Protein Data Bank. It is possible to open a structure file directly from the output of the search by clicking the "Download and Open" button or by double clicking directly on the relevant row.

Select the molecule structure of interest and click on the button labeled "Download and Open" - or double click on the relevant row - in the table to open the protein structure.

Pressing the "Download and Save" button will save the molecule structure at a user defined destination in the Navigation Area.

The button "Open at NCBI" links directly to the structure summary page at NCBI: clicking this button will open individual NCBI pages describing each of the selected molecule structures.

17.1.2 From your own file system

A PDB file can also be imported from your own file system using the standard import function:

Toolbar | Import (📄) | Standard Import (📄)

In the Import dialog, select the structure(s) of interest from a data location and tick "Automatic import" (figure 17.2). Specify where to save the imported PDB file and click **Finish**.

Double clicking on the imported file in the **Navigation Area** will open the structure as a **Molecule Project** in the **View Area** of the *CLC Genomics Workbench*. Another option is to drag the PDB file from the **Navigation Area** to the **View Area**. This will automatically open the protein structure as a **Molecule Project**.

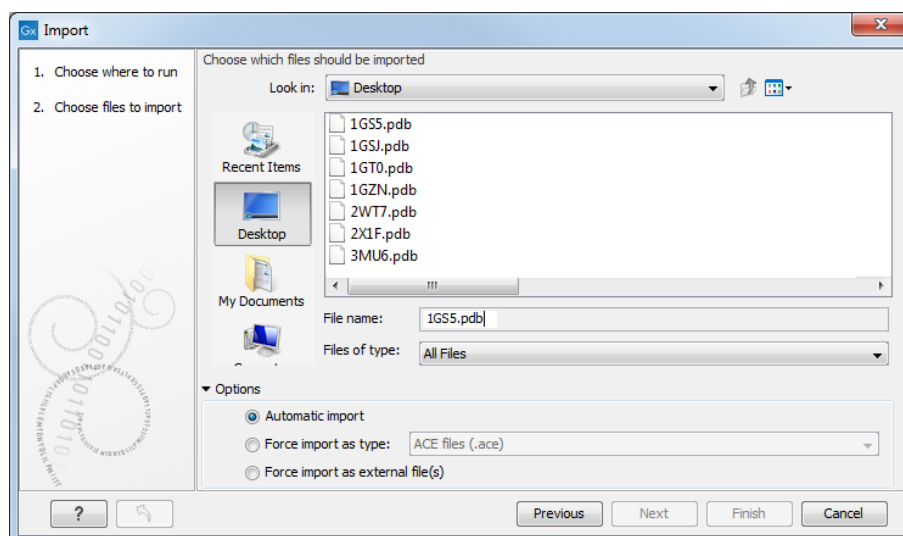


Figure 17.2: A PDB file can be imported using the Standard Import tool.

17.1.3 BLAST search against the PDB database

It is also possible to make a BLAST search against the PDB database, by going to:

Tools | BLAST (📄) | **BLAST at NCBI** (🌐)

After selecting where to run the analysis, specify which input sequences to use for the BLAST search in the "BLAST at NCBI" dialog, within the box named "Select sequences of same type". More than one sequence can be selected at the same time, as long as the sequences are of the same type (figure 17.3).

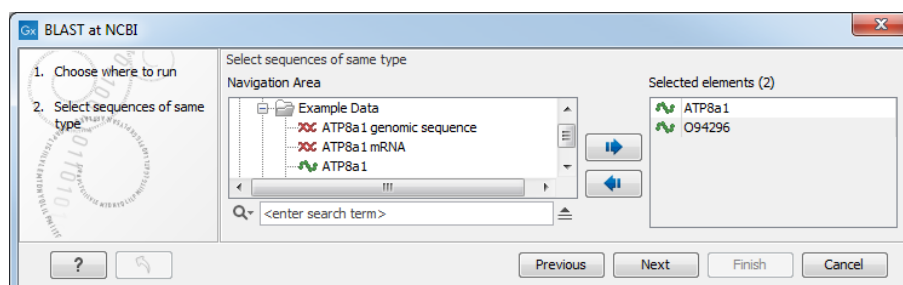


Figure 17.3: Select the input sequence of interest. In this example a protein sequence for ATPase class I type 8A member 1 and an ATPase ortholog from *S. pombe* have been selected.

Click **Next** and choose program and database (figure 17.4). When a protein sequence has been used as input, select "Program: blastp: Protein sequence and database" and "Database: Protein Data Bank proteins (pdb)".

It is also possible to use mRNA and genomic sequences as input. In such cases the program "blastx: Translated DNA sequence and protein database" should be used.

Please refer to section 16.1.1 for further description of the individual parameters in the wizard steps.

When you click on the button labeled **Finish**, a BLAST output is generated that shows local sequence alignments between your input sequence and a list of matching proteins with known structures available.

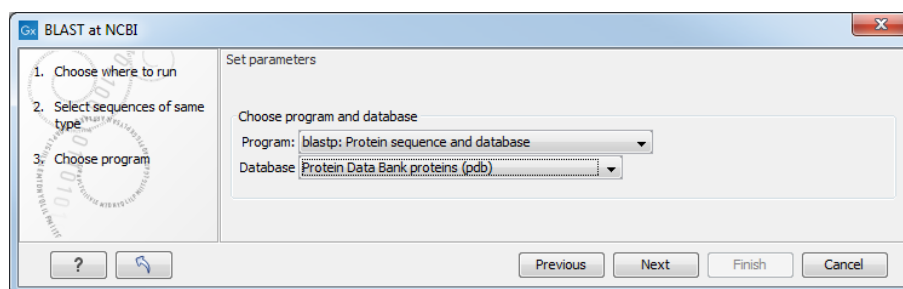


Figure 17.4: Select database and program.

Note! The BLAST at NCBI search can take up to several minutes, especially when mRNA and genomic sequences are used as input.

Switch to the "BLAST Table" editor view to select the desired entry (figure 17.5). If you have performed a multi BLAST, to get access to the "BLAST Table" view, you must first double click on each row to open the entries individually.

In this view four different options are available:

- **Download and Open** The sequence that has been selected in the table is downloaded and opened in the **View Area**.
- **Download and Save** The sequence that has been selected in the table is downloaded and saved in the **Navigation Area**.
- **Open at NCBI** The protein sequence that has been selected in the table is opened at NCBI.
- **Open Structure** Opens the selected structure in a **Molecule Project** in the **View Area**.

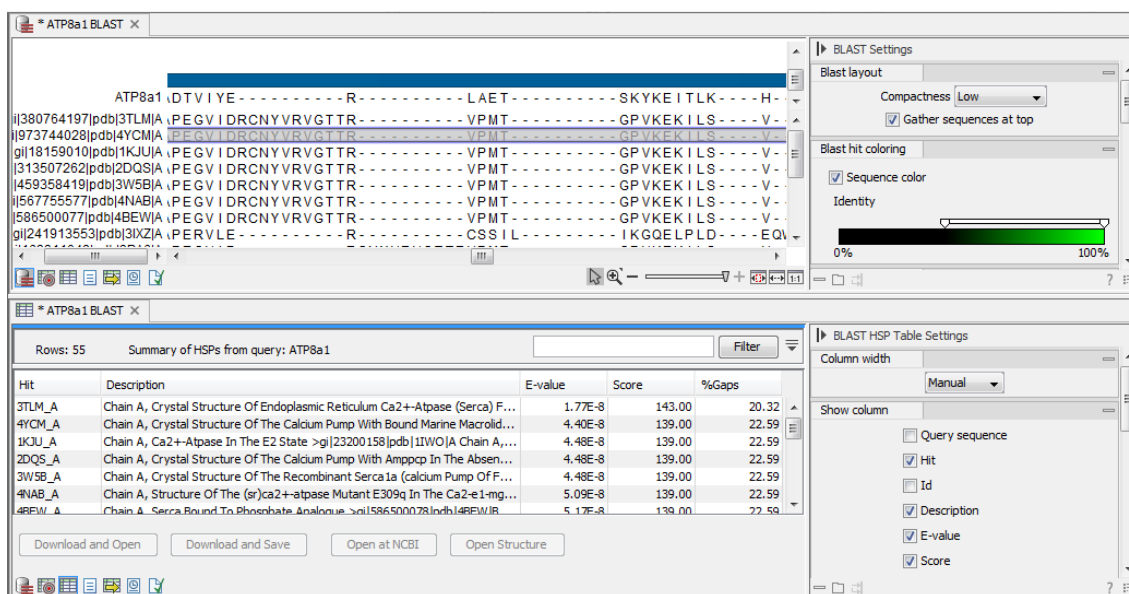


Figure 17.5: Top: The output from "BLAST at NCBI". Bottom: The "BLAST table". One of the protein sequences has been selected. This activates the four buttons under the table. Note that the table and the BLAST Graphics are linked, this means that when a sequence is selected in the table, the same sequence will be highlighted in the BLAST Graphics view.

17.1.4 Import issues

When opening an imported molecule file for the first time, a notification is briefly shown in the lower left corner of the **Molecule Project** editor, with information of the number of issues encountered during import of the file. The issues are categorized and listed in a table view in the Issues view. The Issues list can be opened by selecting **Show | Issues** from the menu appearing when right-clicking in an empty space in the 3D view (figure 17.6).

Alternatively, the issues can be accessed from the lower left corner of the view, where buttons are shown for each available view. If you hold down the Ctrl key (Cmd on Mac) while clicking on the Issues icon (🔍), the list will be shown in a split view together with the 3D view. The issues list is linked with the molecules in the 3D view, such that selecting an entry in the list will select the implicated atoms in the view, and zoom to put them into the center of the 3D view.

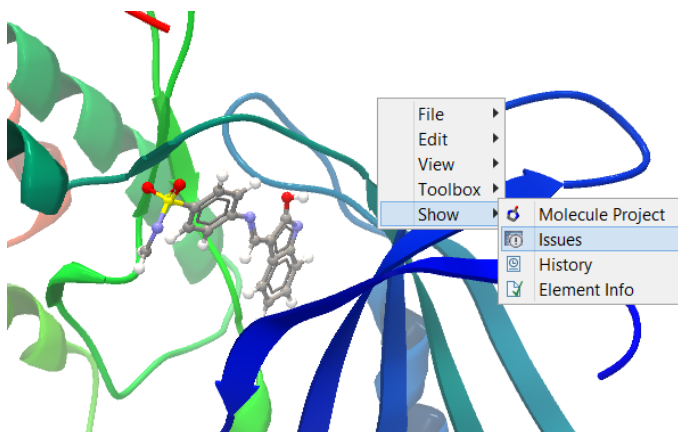


Figure 17.6: At the bottom of the Molecule Project it is possible to switch to the "Show Issues" view by clicking on the "table-with-exclamation-mark" icon.

17.2 Viewing molecular structures in 3D

An example of a 3D structure that has been opened as a **Molecule Project** is shown in figure 17.7.

If you have problems viewing 3D structures, please check your system matches the requirements for 3D viewers. See section 1.3.

Moving and rotating The molecules can be rotated by holding down the left mouse button while moving the mouse. The right mouse button can be used to move the view.

Zooming can be done with the scroll-wheel or by holding down both left and right buttons while moving the mouse up and down.

All molecules in the **Molecule Project** are listed in categories in the **Project Tree**. The individual molecules or whole categories can be hidden from the view by un-checking the boxes next to them.

It is possible to bring a particular molecule or a category of molecules into focus by selecting the molecule or category of interest in the **Project Tree** view and double-click on the molecule or category of interest. Another option is to use the zoom-to-fit button (↔) at the bottom of the **Project Tree** view.

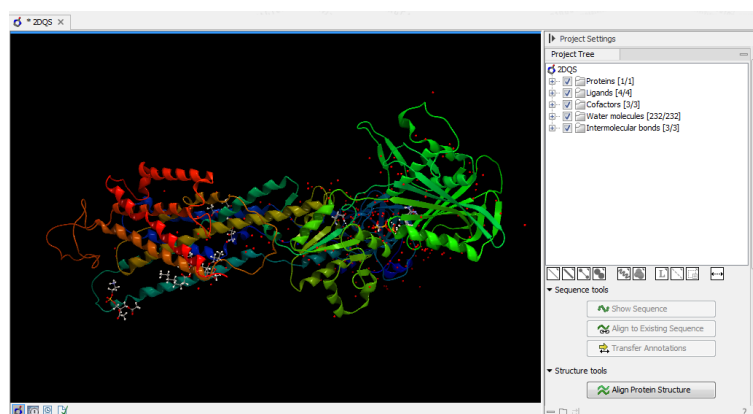


Figure 17.7: 3D view of a calcium ATPase. All molecules in the PDB file are shown in the Molecule Project. The Project Tree in the right side of the window lists the involved molecules.

Troubleshooting 3D graphics errors The 3D viewer uses OpenGL graphics hardware acceleration in order to provide the best possible experience. If you experience any graphics problems with the 3D view, please make sure that the drivers for your graphics card are up-to-date.

If the problems persist after upgrading the graphics card drivers, it is possible to change to a rendering mode, which is compatible with a wider range of graphic cards. To change the graphics mode go to Edit in the menu bar, select "Preferences", Click on "View", scroll down to the bottom and find "Molecule Project 3D Editor" and uncheck the box "Use modern OpenGL rendering".

Finally, it should be noted that certain types of visualization are more demanding than others. In particular, using multiple molecular surfaces may result in slower drawing, and even result in the graphics card running out of available memory. Consider creating a single combined surface (by using a selection) instead of creating surfaces for each single object. For molecules with a large number of atoms, changing to wireframe rendering and hiding hydrogen atoms can also greatly improve drawing speed.

17.3 Customizing the visualization

The molecular visualization of all molecules in the Molecule Project can be customized using different visualization styles. The styles can be applied to one molecule at a time, or to a whole category (or a mixture), by selecting the name of either the molecule or the category. Holding down the Ctrl (Cmd on Mac) or shift key while clicking the entry names in the **Project Tree** will select multiple molecules/categories.

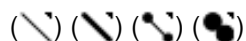
The six leftmost quick-style buttons below the **Project Tree** view give access to the molecule visualization styles, while context menus on the buttons (accessible via right-click or left-click-hold) give access to the color schemes available for the visualization styles. Visualization styles and color schemes are also available from context menus directly on the selected entries in the **Project Tree**. Other quick-style buttons are available for displaying hydrogen bonds between Project Tree entries, for displaying labels in the 3D view and for creating custom atom groups. They are all described in detail below.

Note! Whenever you wish to change the visualization styles by right-clicking the entries in the **Project Tree**, please be aware that you must first click on the entry of interest, and ensure it is

highlighted in blue, before right-clicking.

17.3.1 Visualization styles and colors

Wireframe, Stick, Ball and stick, Space-filling/CPK



Four different ways of visualizing molecules by showing all atoms are provided: Wireframe, Stick, Ball and stick, and Space-filling/CPK.

The visualizations are mutually exclusive meaning that only one style can be applied at a time for each selected molecule or atom group.

Six color schemes are available and can be accessed via right-clicking on the quick-style buttons:

- Color by Element. Classic CPK coloring based on atom type (e.g. oxygen red, carbon gray, hydrogen white, nitrogen blue, sulfur yellow).
- Color by Temperature. For PDB files, this is based on the b-factors. For structure models created with tools in a CLC workbench, this is based on an estimate of the local model quality. The color scale goes from blue (0) over white (50) to red (100). The b-factors as well as the local model quality estimate are measures of uncertainty or disorder in the atom position; the higher the number, the higher the uncertainty.
- Color Carbons by Entry. Each entry (molecule or atom group) is assigned its own specific color. Only carbon atoms are colored by the specific color, other atoms are colored by element.
- Color by Entry. Each entry (molecule or atom group) is assigned its own specific color.
- Custom Color. The user selects a molecule color from a palette.
- Custom Carbon Color. The user selects a molecule color from a palette. Only carbon atoms are colored by the specific color, other atoms are colored by element.

Backbone



For the molecules in the Proteins and Nucleic Acids categories, the backbone structure can be visualized in a schematic rendering, highlighting the secondary structure elements for proteins and matching base pairs for nucleic acids. The backbone visualization can be combined with any of the atom-level visualizations.

Five color schemes are available for backbone structures:

- Color by Residue Position. Rainbow color scale going from blue over green to yellow and red, following the residue number.
- Color by Type. For proteins, beta sheets are blue, helices red and loops/coil gray. For nucleic acids backbone ribbons are white while the individual nucleotides are indicated in green (T/U), red (A), yellow (G), and blue (C).

- Color by Backbone Temperature. For PDB files, this is based on the b-factors for the C α atoms (the central carbon atom in each amino acid). For structure models created with tools in the workbench, this is based on an estimate of the local model quality. The color scale goes from blue (0) over white (50) to red (100). The b-factors as well as the local model quality estimate are measures of uncertainty or disorder in the atom position; the higher the number, the higher the uncertainty.
- Color by Entry. Each chain/molecule is assigned its own specific color.
- Custom Color. The user selects a molecule color from a palette.

Surfaces



Molecular surfaces can be visualized.

Five color schemes are available for surfaces:

- Color by Charge. Charged amino acids close to the surface will show as red (negative) or blue (positive) areas on the surface, with a color gradient that depends on the distance of the charged atom to the surface.
- Color by Element. Smoothed out coloring based on the classic CPK coloring of the heteroatoms close to the surface.
- Color by Temperature. Smoothed out coloring based on the temperature values assigned to atoms close to the surface (See the "Wireframe, Stick, Ball and stick, Space-filling/CPK" section above).
- Color by Entry. Each surface is assigned its own specific color.
- Custom Color. The user selects a surface color from a palette.

A surface spanning multiple molecules can be visualized by creating a custom atom group that includes all atoms from the molecules (see section 17.3.1).

It is possible to adjust the opacity of a surface by adjusting the transparency slider at the bottom of the menu.

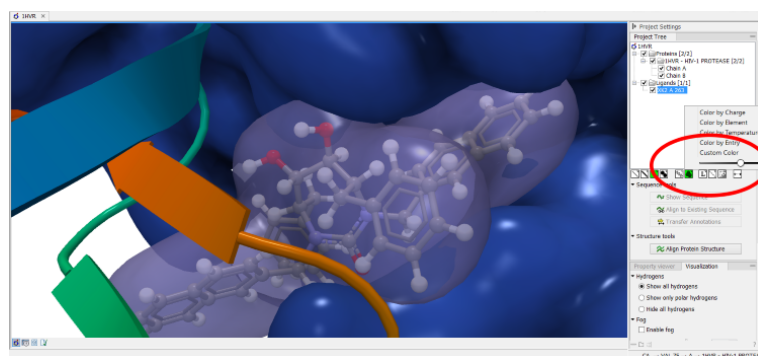


Figure 17.8: Transparent surfaces

Notice that visual artifacts may appear when rotating a transparent surface. These artifacts disappear as soon as the mouse is released.

Labels

(L)

Labels can be added to the molecules in the view by selecting an entry in the Project Tree and clicking the label button at the bottom of the Project Tree view. The color of the labels can be adjusted from the context menu by right clicking on the selected entry (which must be highlighted in blue first) or on the label button in the bottom of the Project Tree view (see figure 17.9).

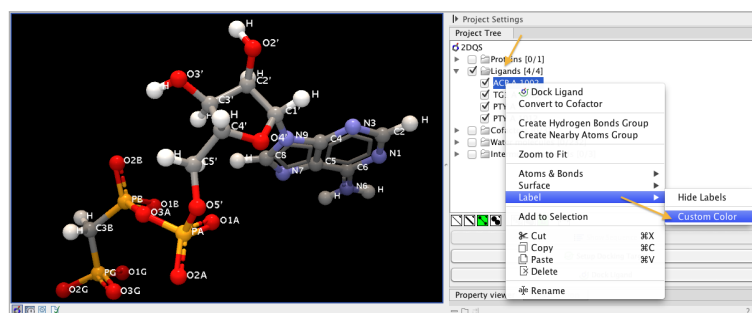


Figure 17.9: The color of the labels can be adjusted in two different ways. Either directly using the label button by right clicking the button, or by right clicking on the molecule or category of interest in the Project Tree.

- For proteins and nucleic acids, each residue is labeled with the PDB name and number.
- For ligands, each atom is labeled with the atom name as given in the input.
- For cofactors and water, one label is added with the name of the molecule.
- For atom groups including protein atoms, each protein residue is labeled with the PDB name and number.
- For atom groups not including protein atoms, each atom is labeled with the atom name as given in the input.

Labels can be removed again by clicking on the label button.

Hydrogen bonds

(H)

The Show Hydrogen Bond visualization style may be applied to molecules and atom group entries in the project tree. If this style is enabled for a project tree entry, hydrogen bonds will be shown to all other currently visible objects. The hydrogen bonds are updated dynamically: if a molecule is toggled off, the hydrogen bonds to it will not be shown.

It is possible to customize the color of the hydrogen bonds using the context menu.

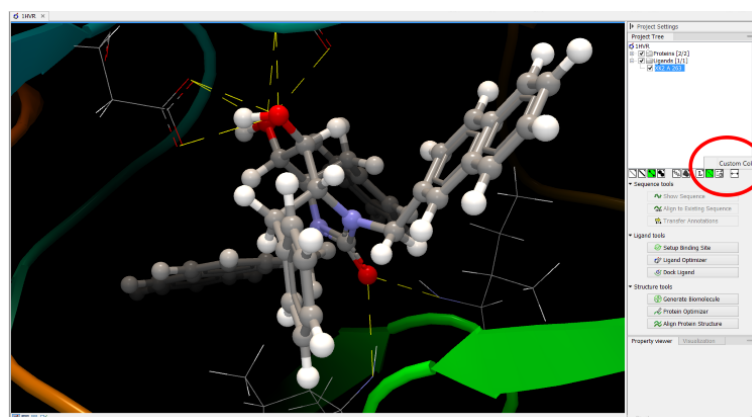


Figure 17.10: The hydrogen bond visualization setting, with custom bond color.

Create atom group



Often it is convenient to use a unique visualization style or color to highlight a particular set of atoms, or to visualize only a subset of atoms from a molecule. This can be achieved by creating an atom group. Atom groups can be created based on atoms selected in the 3D view or entries selected in the Project Tree. When an atom group has been created, it appears as an entry in the Project Tree in the category "Atom groups". The atoms can then be hidden or shown, and the visualization changed, just as for the molecule entries in the Project Tree.

Note that an atom group entry can be renamed. Select the atom group in the Project Tree and invoke the right-click context menu. Here, the Rename option is found.

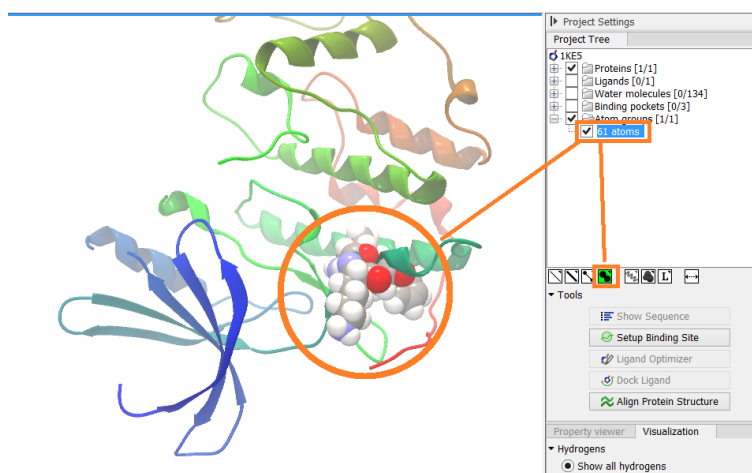


Figure 17.11: An atom group that has been highlighted by adding a unique visualization style.

Create atom group based on atoms selected in 3D view

When atoms are selected in the 3D view, brown spheres indicate which atoms are included in the selection. The selection will appear as the entry "Current" in the Selections category in the Project Tree.

Once a selection has been made, press the "Create Atom Group" button and a context menu will

show different options for creating a new atom group based on the selection:

- **Selected Atoms.** Creates an atom group containing exactly the selected atoms (those indicated by brown spheres). If an entire molecule or residue is selected, this option is not displayed.
- **Selected Residue(s)/Molecules.** Creates an atom group that includes all atoms in the selected residues (for entries in the protein and nucleic acid categories) and molecules (for the other categories).
- **Nearby Atoms.** Creates an atom group that contains residues (for the protein and nucleic acid categories) and molecules (for the other categories) within 5 Å of the selected atoms. Only atoms from currently visible Project Tree entries are considered.
- **Hydrogen Bonded Atoms.** Creates an atom group that contains residues (for the protein and nucleic acid categories) and molecules (for the other categories) that have hydrogen bonds to the selected atoms. Only atoms from currently visible Project Tree entries are considered.

There are several ways to select atoms in the 3D view:

- **Double click to select.** Click on an atom to select it. When you double click on an atom that belongs to a residue in a protein or in a nucleic acid chain, the entire residue will be selected. For small molecules, the entire molecule will be selected.
- **Adding atoms to a selection.** Holding down Ctrl while picking atoms, will pile up the atoms in the selection. All atoms in a molecule or category from the Project Tree, can be added to the "Current" selection by choosing "Add to Current Selection" in the context menu. Similarly, entire molecules can be removed from the current selection via the context menu.
- **Spherical selection.** Hold down the shift-key, click on an atom and drag the mouse away from the atom. Then a sphere centered on the atom will appear, and all atoms inside the sphere, visualized with one of the all-atom representations will be selected. The status bar (lower right corner) will show the radius of the sphere.
- **Show Sequence.** Another option is to select protein or nucleic acid entries in the Project Tree, and click the "Show Sequence" button found below the Project Tree, see section 17.4.1. A split-view will appear with a sequence editor for each of the sequence data types (Protein, DNA, RNA) (figure 17.12). If you then select residues in the sequence view, the backbone atoms of the selected residues will show up as the "Current" selection in the 3D view and the Project Tree view. Notice that the link between the 3D view and the sequence editor is lost if either window is closed, or if the sequence is modified.
- **Align to Existing Sequence.** If a single protein chain is selected in the Project Tree, the "Align to Existing Sequence" button can be clicked, see section 17.4.2. This links the protein sequence with a sequence or sequence alignment found in the Navigation Area. A split-view appears with a sequence alignment where the sequence of the selected protein chain is linked to the 3D structure, and atoms can be selected in the 3D view, just as for the "Show Sequence" option.

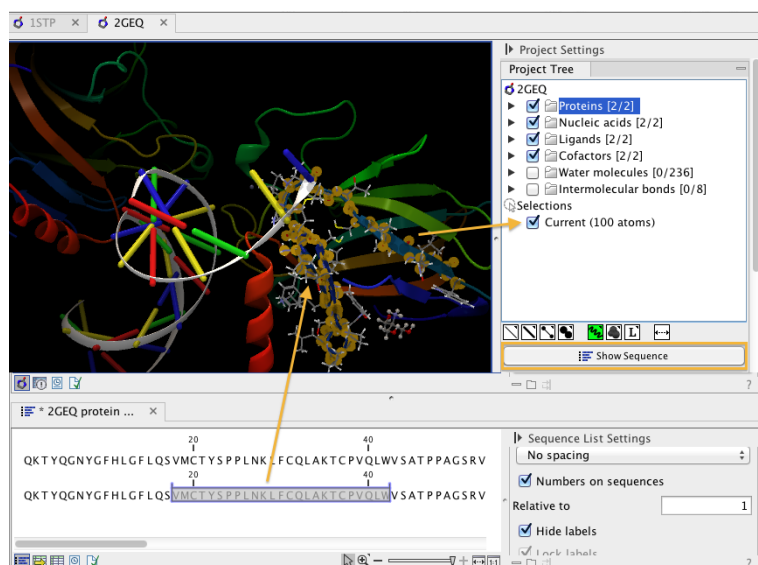


Figure 17.12: The protein sequence in the split view is linked with the protein structure. This means that when a part of the protein sequence is selected, the same region in the protein structure will be selected.

Create atom group based on entries selected in the Project Tree

Select one or more entries in the Project Tree, and press the "Create Atom Group" button, then a context menu will show different options for creating a new atom group based on the selected entries:

- **Nearby Atoms.** Creates an atom group that contains residues (for the protein and nucleic acid categories) and molecules (for the other categories) within 5 Å of the selected entries. Only atoms from currently visible Project Tree entries are considered.
- **Hydrogen Bonded Atoms.** Creates an atom group that contains residues (for the protein and nucleic acid categories) and molecules (for the other categories) that have hydrogen bonds to the selected entries. Only atoms from currently visible Project Tree entries are considered.

If a Binding Site Setup is present in the Project Tree (A Binding Site Setup could only be created using the now discontinued CLC Drug Discovery Workbench), and entries from the Ligands or Docking results categories are selected, two extra options are available under the header **Create Atom Group (Binding Site)**. For these options, atom groups are created considering all molecules included in the Binding Site Setup, and thus not taking into account which Project Tree entries are currently visible.

Zoom to fit

(\longleftrightarrow)

The "Zoom to fit" button can be used to automatically move a region of interest into the center of the screen. This can be done by selecting a molecule or category of interest in the Project Tree view followed by a click on the "Zoom to fit" button (\longleftrightarrow) at the bottom of the Project Tree view

(figure 17.13). Double-clicking an entry in the Project Tree will have the same effect.

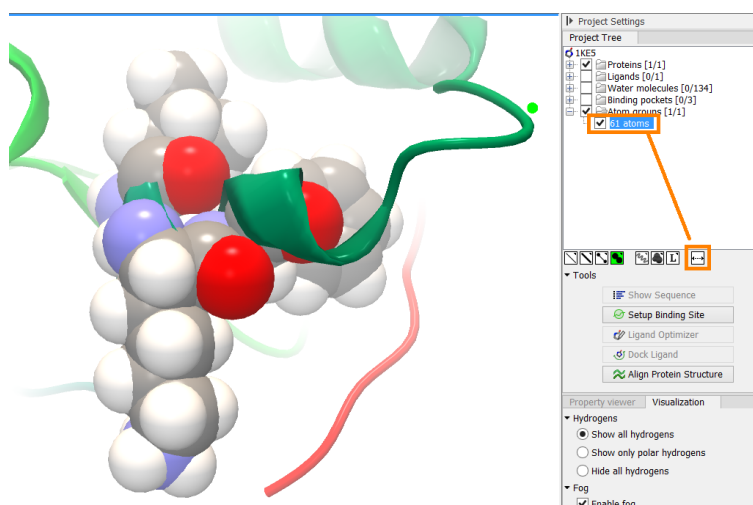


Figure 17.13: The "Fit to screen" button can be used to bring a particular molecule or category of molecules in focus.

17.3.2 Project settings

A number of general settings can be adjusted from the **Side Panel**. Personal settings as well as molecule visualizations can be saved by clicking in the lower right corner of the **Side Panel** (☰). This is described in detail in section 4.6.

Project Tree Tools

Just below the Project Tree, the following tools are available

- **Show Sequence** Select molecules which have sequences associated (Protein, DNA, RNA) in the Project Tree, and click this button. Then, a split-view will appear with a sequence editor for each of the sequence data types (Protein, DNA, RNA). This is described in section 17.4.1.
- **Align to Existing Sequence** Select a protein chain in the Project Tree, and click this button. Then protein sequences and sequence alignments found in the Navigation Area, can be linked with the protein structure. This is described in section 17.4.2.
- **Transfer Annotations** Select a protein chain in the Project Tree, that has been linked with a sequence using either the "Show Sequence" or "Align to Existing Sequence" options. Then it is possible to transfer annotations between the structure and the linked sequence. This is described in section 17.4.3.
- **Align Protein Structure** This will invoke the dialog for aligning protein structures based on global alignment of whole chains or local alignment of e.g. binding sites defined by atom groups. This is described in section 17.5.

Property viewer

The Property viewer, found in the Side Panel, lists detailed information about the atoms that the mouse hovers over. For all atoms the following information is listed:

- **Molecule** The name of the molecule the atom is part of.
- **Residue** For proteins and nucleic acids, the name and number of the residue the atom belongs to is listed, and the chain name is displayed in parentheses.
- **Name** The particular atom name, if given in input, with the element type (Carbon, Nitrogen, Oxygen...) displayed in parentheses.
- **Hybridization** The atom hybridization assigned to the atom.
- **Charge** The atomic charge as given in the input file. If charges are not given in the input file, some charged chemical groups are automatically recognized and a charge assigned.

For atoms in molecules imported from a PDB file, extra information is given:

- **Temperature** Here is listed the b-factor assigned to the atom in the PDB file. The b-factor is a measure of uncertainty or disorder in the atom position; the higher the number, the higher the disorder.
- **Occupancy** For each atom in a PDB file, the occupancy is given. It is typically 1, but if atoms are modeled in the PDB file, with no foundation in the raw data, the occupancy is 0. If a residue or molecule has been resolved in multiple positions, the occupancy is between 0 and 1.

For atoms in protein models created by tools in the workbench, the following extra information is given:

- **Temperature** For structure models, the temperature value is an estimate of local structure uncertainty. The three aspects contributing to the assigned atom temperature is also listed, and described in section [20.6.2](#). The temperature value is a measure of uncertainty or disorder in the atom position; the higher the number, the higher the disorder.
- **Occupancy** For modeled structures and atoms, the occupancy is set to zero.

If an atom is selected, the Property view will be frozen with the details of the selected atom shown. If then a second atom is selected (by holding down Ctrl while clicking), the distance between the two selected atoms is shown. If a third atom is selected, the angle for the second atom selected is shown. If a fourth atom is selected, the dihedral angle measured as the angle between the planes formed by the three first and three last selected atoms is given.

If a molecule is selected in the Project Tree, the Property view shows information about this molecule. Two measures are always shown:

- **Atoms** Number of atoms in the molecule.
- **Weight** The weight of the molecule in Daltons.

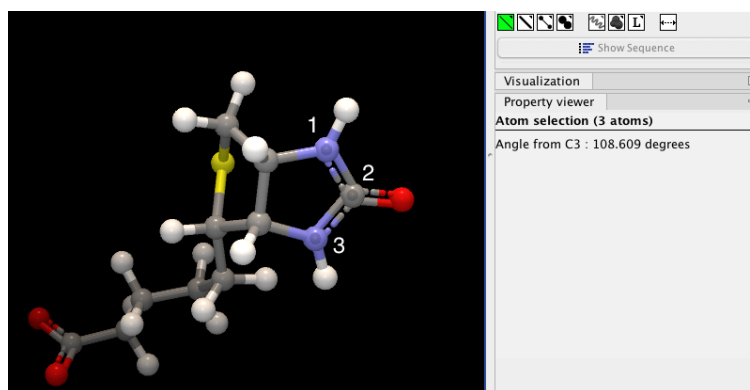


Figure 17.14: Selecting two, three, or four atoms will display the distance, angle, or dihedral angle, respectively.

Visualization settings

Under "Visualization" five options exist:

- **Hydrogens** Hydrogen atoms can be shown (Show all hydrogens), hidden (Hide all hydrogens) or partially shown (Show only polar hydrogens).
- **Fog** "Fog" is added to give a sense of depth in the view. The strength of the fog can be adjusted or it can be disabled.
- **Clipping plane** This option makes it possible to add an imaginary plane at a specified distance along the camera's line of sight. Only objects behind this plane will be drawn. It is possible to clip only surfaces, or to clip surfaces together with proteins and nucleic acids. Small molecules, like ligands and water molecules, are never clipped.
- **3D projection** The view is opened up towards the viewer, with a "Perspective" 3D projection. The field of view of the perspective can be adjusted, or the perspective can be disabled by selecting an orthographic 3D projection.
- **Coloring** The background color can be selected from a color palette by clicking on the colored box.

Snapshots of the molecule visualization To save the current view as a picture, right-click in the **View Area** and select "File" and "Export Graphics". Another way to save an image is by pressing the "Graphics" button in the Workbench toolbar (🖨️). Next, select the location where you wish to save the image, select file format (PNG, JPEG, or TIFF), and provide a name, if you wish to use another name than the default name.

You can also save the current view directly on data with a custom name, so that it can later be applied (see section 4.6).

17.4 Tools for linking sequence and structure

The *CLC Genomics Workbench* has functionality that allows you to link a protein sequence to a protein structure. Selections made on the sequence will show up on the structure. This allows

you to explore a protein sequence in a 3D structure context. Furthermore, sequence annotations can be transferred to annotations on the structure and annotations on the structure can be transferred to annotations on the sequence (see section 17.4.3).

17.4.1 Show sequence associated with molecule

From the Side Panel, sequences associated with the molecules in the Molecule Project can be opened as separate objects by selecting protein or nucleic acid entries in the Project Tree and clicking the button labeled "Show Sequence" (figure 17.15). This will generate a Sequence or Sequence List for each selected sequence type (protein, DNA, RNA). The sequences can be used to select atoms in the Molecular Project as described in section 17.3.1. The sequences can also be used as input for sequence analysis tools or be saved as independent objects. You can later re-link to the sequence using "Align to Existing Sequence" (see section 17.4.2).

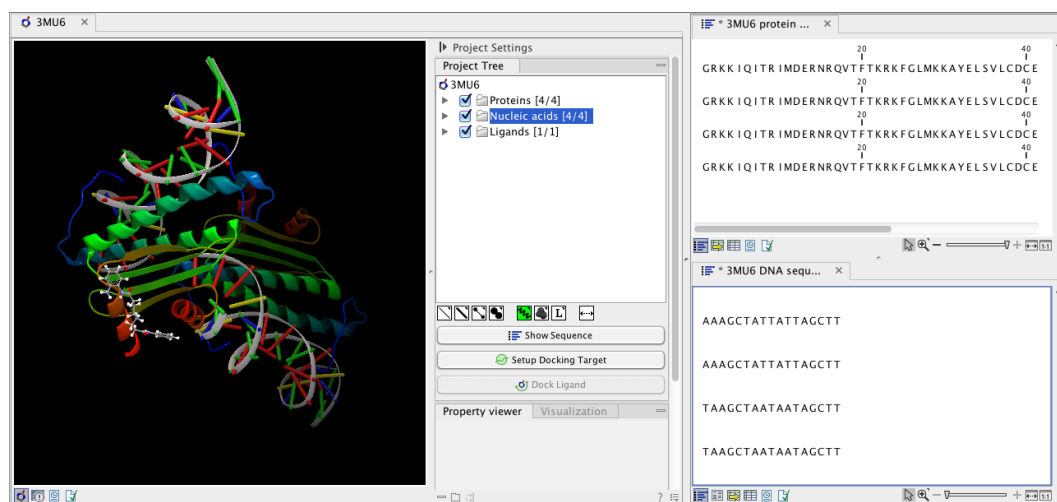


Figure 17.15: Protein chain sequences and DNA sequences are shown in separate views.

17.4.2 Link sequence or sequence alignment to structure

The "Align to Existing Sequence" button can be used to map and link existing sequences or sequence alignments to a protein structure chain in a Molecule Project (3D view). It can also be used to reconnect a protein structure chain to a sequence or sequence alignment previously created by Show Sequence (section 17.4.1) or Align to Existing Sequence.

Select a single protein chain in the project tree (see figure 17.16). Pressing "Align to Existing Sequence" then opens a Navigation Area browser, where it is possible to select one or more Sequence, Sequence Lists, or Alignments, to link with the selected protein chain.

If the sequences or alignments already contain a sequence identical to the protein chain selected in the Molecule Project (i.e. same name and amino acid sequence), this sequence is linked to the protein structure. If no identical sequence is present, a sequence is extracted from the protein structure (as for Show Sequence, see section 17.4.1), and a sequence alignment is created between this sequence and the sequences or alignments selected from the Navigation Area. The new sequence alignment is created (see section 24.1) with the following settings:

- Gap open cost: 10.0

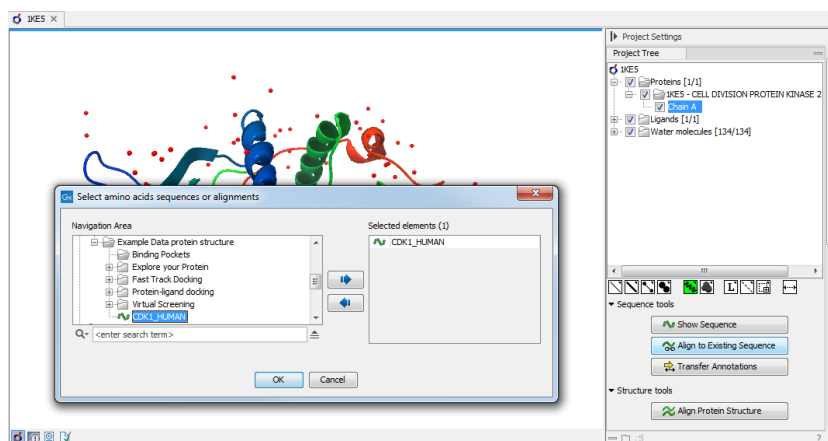


Figure 17.16: Select a single protein chain in the Project Tree and invoke "Align to Existing Sequence".

- Gap Extension cost: 1.0
- End gap cost: free
- Existing alignments are not redone

When the link is established, selections on the linked sequence in the sequence editor will create atom selections in the 3D view, and it is possible to transfer annotations between the linked sequence and the 3D protein chain (see section 17.4.3). Note that the link will be broken if either the sequence or the 3D protein chain is modified.

Two tips if the link is to a sequence in an alignment:

1. Read about how to change the layout of sequence alignments in section 24.2
2. It is only annotations present on the sequence linked to the 3D view that can be transferred to atom groups on the structure. To transfer sequence annotations from other sequences in the alignment, first copy the annotations to the sequence in the alignment that is linked to the structure (see figure 17.19 and section 24.3).

17.4.3 Transfer annotations between sequence and structure

The Transfer Annotations dialog makes it possible to create new atom groups (annotations on structure) based on protein sequence annotations and vice versa.

You can read more about sequence annotations in section 15.3 17.3.1 and more about atom groups in section 17.3.1.

Before it is possible to transfer annotations, a link between a protein sequence editor and a Molecule Project (a 3D view) must be established. This is done either by opening a sequence associated with a protein chain in the 3D view using the 'Show Sequence' button (see section 17.4.1 17.4.2) or by mapping to an existing sequence or sequence alignment using the 'Align to Existing Sequence' button (see section 17.4.2).

Invoke the Transfer Annotations dialog by selecting a linked protein chain in the Project Tree and press 'Transfer Annotations' (see figure 17.17).

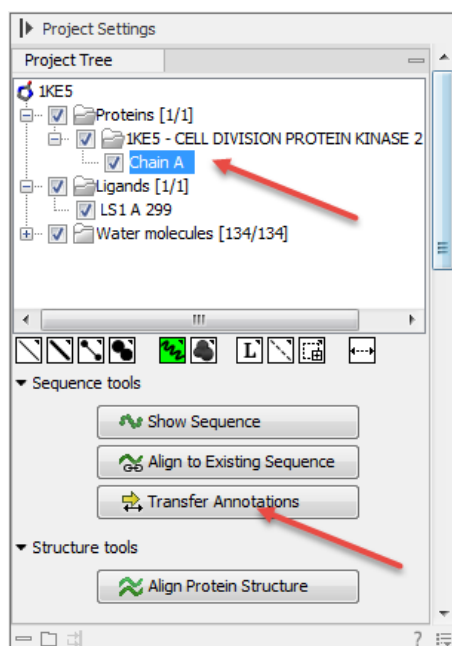


Figure 17.17: Select a single protein chain in the Project Tree and invoke "Transfer Annotations".

The dialog contains two tables (see figure 17.18). The left table shows all atom groups in the Molecule Project, with at least one atom on the selected protein chain. The right table shows all annotations present on the linked sequence. While the Transfer Annotations dialog is open, it is not possible to make changes to neither the sequence nor the Molecule Project, however, changes to the visualization styles are allowed.

How to undo annotation transfers

In order to undo operations made using the Transfer Annotations dialog, the dialog must first be closed. To undo atom groups added to the structure, activate the 3D view by clicking in it and press Undo in the Toolbar. To undo annotations added to the sequence, activate the sequence view by clicking in it and press Undo in the Toolbar.

Transfer sequence annotations from aligned sequences

It is only annotations present on the sequence linked to the 3D view that can be transferred to atom groups on the structure. If you wish to transfer annotations that are found on other sequences in a linked sequence alignment, you need first to copy the sequence annotations to the actual sequence linked to the 3D view (the sequence with the same name as the protein structure). This is done by invoking the context menu on the sequence annotation you wish to copy (see figure 17.19 and section 24.3).

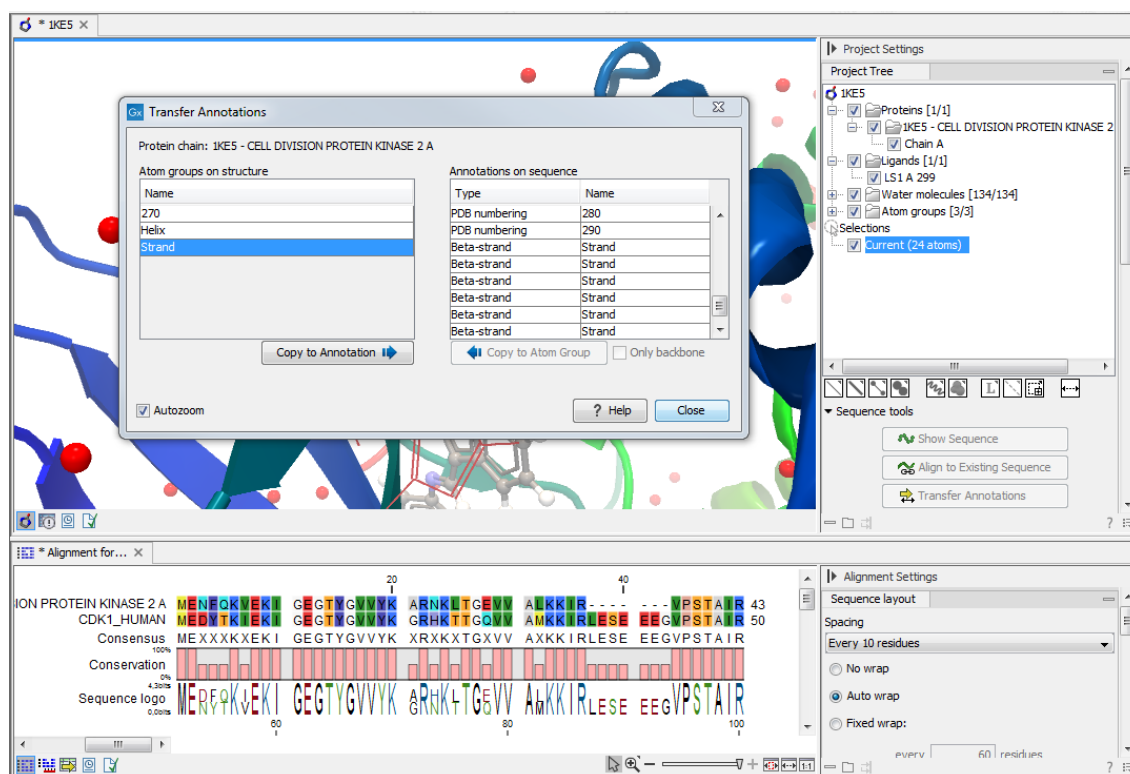



Figure 17.18: The Transfer Annotations dialog allow you to select annotations listed in the two tables, and copy them from structure to sequence or vice versa.



Figure 17.19: Copy annotations from sequences in the alignment to the sequence linked to the 3D view.

17.5 Align Protein Structure

The Align Protein Structure tool allows you to compare a protein or binding pocket in a **Molecule Project** with proteins from other **Molecule Projects**. The tool is invoked using the  Align Protein Structure action from the **Molecule Project Side Panel**. This action will open an interactive dialog box (figure 17.20). By default, when the dialog box is closed with an "OK", a new **Molecule Project** will be opened containing all the input protein structures laid on top of one another. All molecules coming from the same input Molecule Project will have the same color in the initial visualization.

The dialog box contains three fields:

- **Select reference (protein chain or atom group)** This drop-down menu shows all the protein

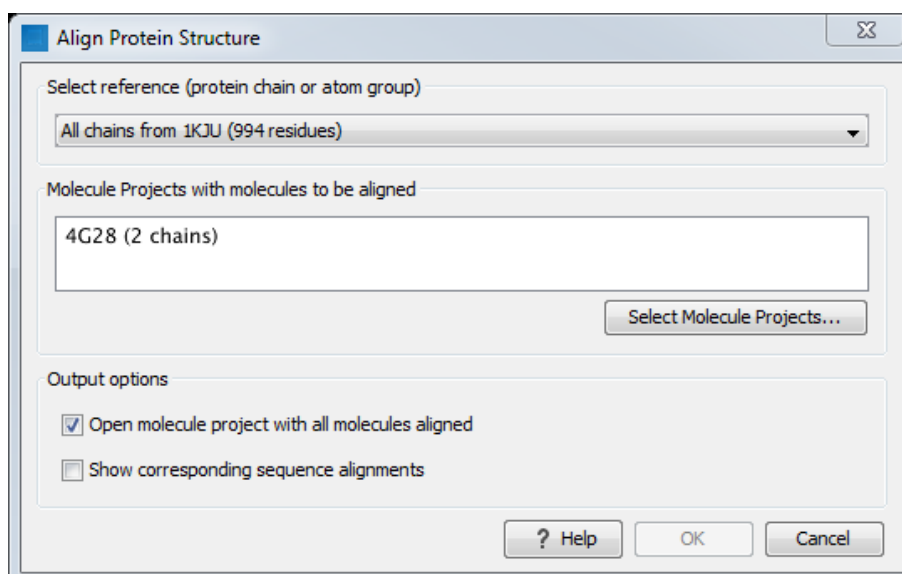


Figure 17.20: The Align Protein Structure dialog box.

chains and residue-containing atom groups in the current **Molecule Project**. If an atom group is selected, the structural alignment will be optimized in that area. The 'All chains from *Molecule Project* option will create a global alignment to all protein chains in the project, fitting e.g. a dimer to a dimer.

- **Molecule Projects with molecules to be aligned** One or more **Molecule Projects** containing protein chains may be selected.
- **Output options** The default output is a single **Molecule Project** containing all the input projects rotated onto the coordinate system of the reference. Several alignment statistics, including the RMSD, TM-score, and sequence identity, are added to the **History** of the output **Molecule Project**. Additionally, a sequence alignments of the aligned structures may be output, with the sequences linked to the 3D structure view.

17.5.1 Example: alignment of calmodulin

Calmodulin is a calcium binding protein. It is composed of two similar domains, each of which binds two calcium atoms. The protein is especially flexible, which can make structure alignment challenging. Here we will compare the calcium binding loops of two calmodulin crystal structures – PDB codes 1A29 and 4G28.

Initial global alignment The 1A29 project is opened and the Align Protein Structure dialog is filled out as in figure 17.20. Selecting "All chains from 1A29" tells the aligner to make the best possible global alignment, favoring no particular region. The output of the alignment is shown in figure 17.21. The blue chain is from 1A29, the brown chain is the corresponding calmodulin chain from 4G28 (a calmodulin-binding chain from the 4G28 file has been hidden from the view). Because calmodulin is so flexible, it is not possible to align both of its domains (enclosed in black boxes) at the same time. A good global alignment would require the brown protein to be

translated in one direction to match the N-terminal domain, and in the other direction to match the C-terminal domain (see black arrows).

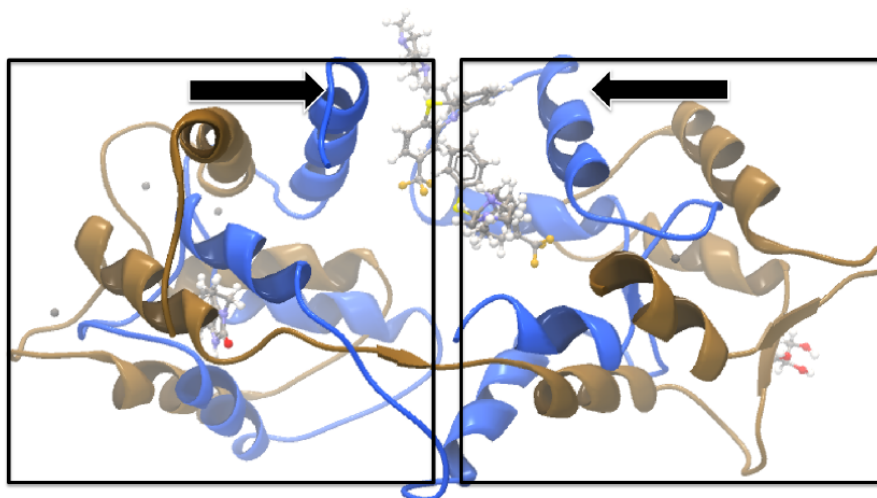


Figure 17.21: Global alignment of two calmodulin structures (blue and brown). The two domains of calmodulin (shown within black boxes) can undergo large changes in relative orientation. In this case, the different orientation of the domains in the blue and brown structures makes a good global alignment impossible: the movement required to align the brown structure onto the blue is shown by arrows – as the arrows point in opposite directions, improving the alignment of one domain comes at the cost of worsening the alignment of the other.

Focusing the alignment on the N-terminal domain To align only the N-terminal domain, we return to the 1A29 project and select the **Show Sequence** action from beneath the **Project Tree**. We highlight the first 62 residues, then convert them into an atom group by right-clicking on the "Current" selection in the **Project Tree** and choosing "Create Group from Selection" (figure 17.22). Using the new atom group as the reference in the alignment dialog leads to the alignment shown in figure 17.23. In addition to the original input proteins, the output now includes two Atom Groups, which contain the atoms on which the alignment was focused. The **History** of the output **Molecule Project** shows that the alignment has 0.9 Å RMSD over the 62 residues.

Aligning a binding site Two bound calcium atoms, one from each calmodulin structure, are shown in the black box of figure 17.23. We now wish to make an alignment that is as good as possible about these atoms so as to compare the binding modes. We return to the 1A29 project, right-click the calcium atom from the cofactors list in the **Project Tree** and select "Create Nearby Atoms Group". Using the new atom group as the reference in the alignment dialog leads to the alignment shown in figure 17.24.

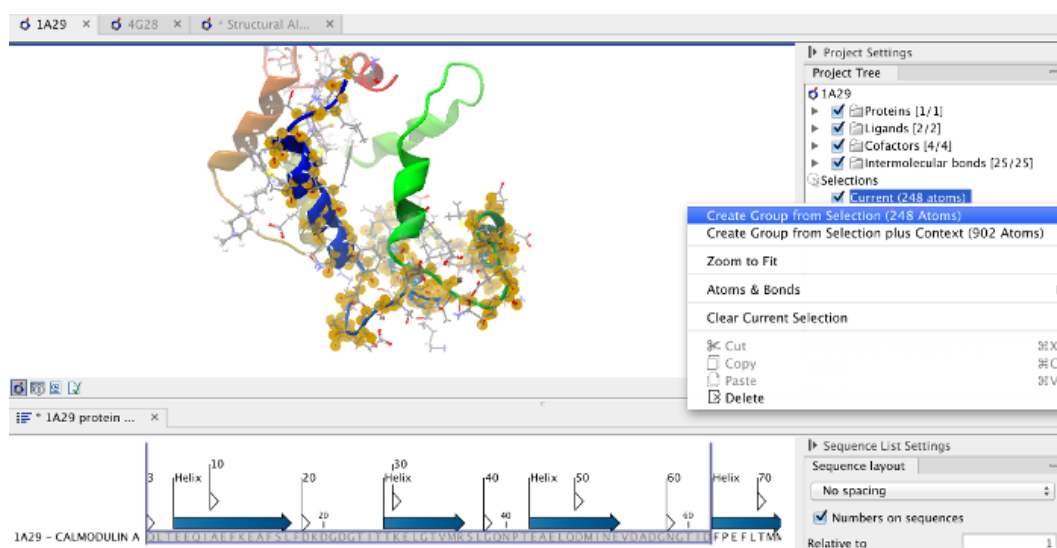


Figure 17.22: Creation of an atom group containing the N-terminal domain of calmodulin.

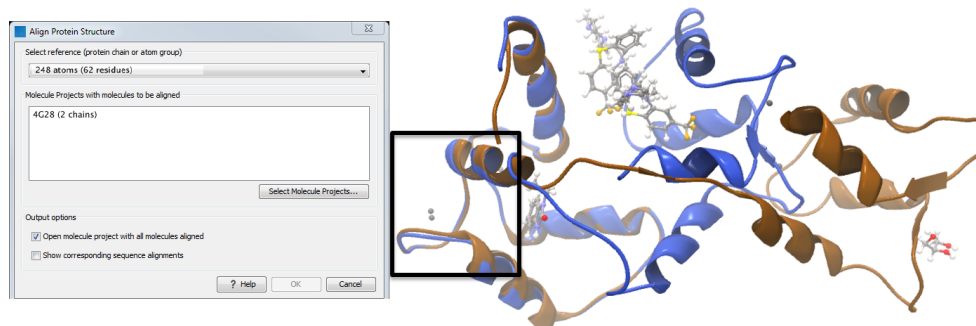


Figure 17.23: Alignment of the same two calmodulin proteins as in figure 17.21, but this time with a focus on the N-terminal domain. The blue and brown structures are now well-superimposed in the N-terminal region. The black box encloses two calcium atoms that are bound to the structures.

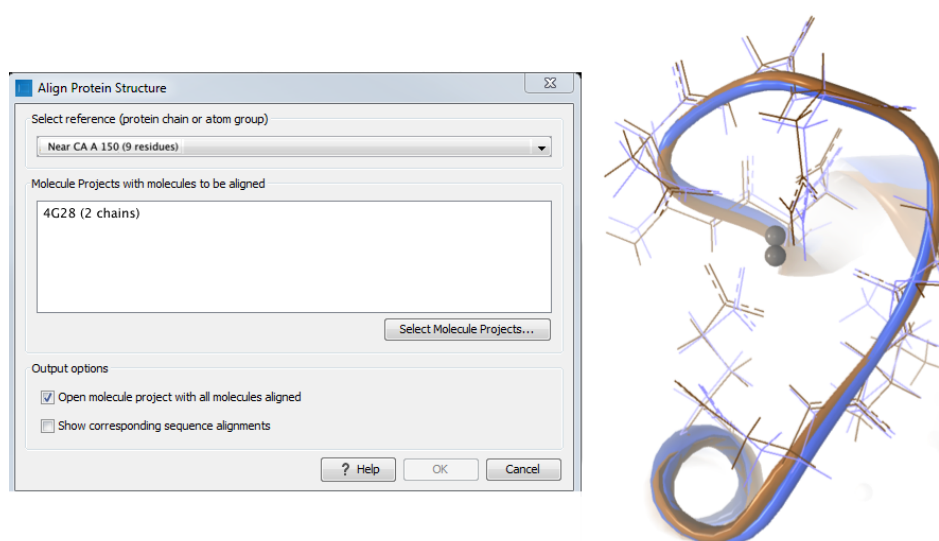


Figure 17.24: Alignment of the same two calmodulin domains as in figure 17.21, but this time with a focus on the calcium atom within the black box of figure 17.23. The calcium atoms are less than 1 Å apart – compatible with thermal motion encoded in the atoms' temperature factors.

17.5.2 The Align Protein Structure algorithm

Any approach to structure alignment must make a trade-off between alignment length and alignment accuracy. For example, is it better to align 200 amino acids at an RMSD of 3.0 Å or 150 amino acids at an RMSD of 2.5 Å? The Align Protein Structure algorithm determines the answer to this question by taking the alignment with the higher TM-score. For an alignment focused on a protein of length L , this is:

$$\text{TM-score} = \frac{1}{L} \sum_i \frac{1}{1 + \frac{d_i}{d(L)}^2}$$

where i runs over the aligned pairs of residues, d_i is the distance between the i^{th} such pair, and $d(L)$ is a normalization term that approximates the average distance between two randomly chosen points in a globular protein of length L [Zhang and Skolnick, 2004]. A perfect alignment has a TM-score of 1.0, and two proteins with a TM-score >0.5 are often said to show structural homology [Xu and Zhang, 2010].

The Align Protein Structure Algorithm attempts to find the *structure alignment* with the highest TM-score. This problem reduces to finding a *sequence alignment* that pairs residues in a way that results in a high TM-score. Several sequence alignments are tried including an alignment with the BLOSUM62 matrix, an alignment of secondary structure elements, and iterative refinements of these alignments.

The Align Protein Structure Algorithm is also capable of aligning entire protein complexes. To do this, it must determine the correct pairing of each chain in one complex with a chain in the other. This set of chain pairings is determined by the following procedure:


1. Make structure alignments between every chain in one complex and every chain in the other. Discard pairs of chains that have a TM-score of < 0.4
2. Find all pairs of structure alignments that are consistent with each other i.e. are achieved by approximately the same rotation
3. Use a heuristic to combine consistent pairs of structure alignments into a single alignment

The heuristic used in the last step is similar to that of MM-align [Mukherjee and Zhang, 2009], whereas the first two steps lead to both a considerable speed up and increased accuracy. The alignment of two 30S ribosome subunits, each with 20 protein chains, can be achieved in less than a minute (PDB codes 2QBD and 1FJG).

17.6 Generate Biomolecule

Protein structures imported from a PDB file show the tertiary structure of proteins, but not necessarily the biologically relevant form (the quaternary structure). Oftentimes, several copies of a protein chain need to arrange in a multi-subunit complex to form a functioning biomolecule. In some PDB files several copies of a biomolecule are present and in others only one chain from a multi-subunit complex is present. In many cases, PDB files have information about how the molecule structures in the file can form biomolecules.

When a PDB file with biomolecule information available has been either downloaded directly to the workbench using the *Search for PDB Structures at NCBI* or imported using *Import Molecules with 3D Coordinates*, the information can be used to generate biomolecule structures in *CLC Genomics Workbench*.

The "Generate Biomolecule" dialog is invoked from the Side Panel of a Molecule Project (figure 17.25). The button  is found in the Structure tools section below the Project Tree.

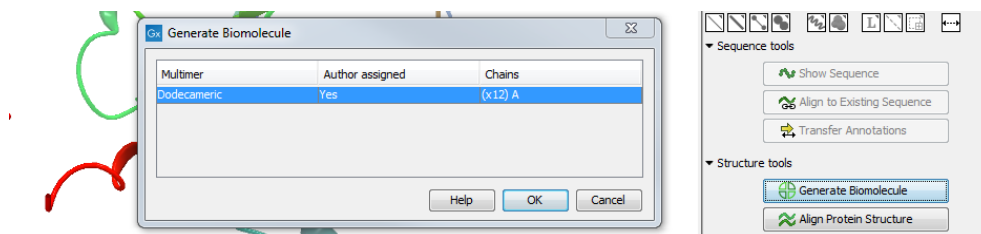


Figure 17.25: The *Generate Biomolecule* dialog lists all possibilities for biomolecules, as given in the PDB files imported to the Molecule Project. In this case, only one biomolecule option is available. The *Generate Biomolecule* button that invokes the dialog can be seen in the bottom right corner of the picture.

There can be more than one biomolecule description available from the imported PDB files. The biomolecule definitions have either been assigned by the crystallographer solving the protein structure (Author assigned = "Yes") or suggested by a software prediction tool (Author assigned = "No"). The third column lists which protein chains are involved in the biomolecule, and how many copies will be made.

Select the preferred biomolecule definition and click OK.

A new Molecule Project will open containing the molecules involved in the selected biomolecule (example in figure 17.26). If required by the biomolecule definition, copies are made of protein chains and other molecules, and the copies are positioned according to the biomolecule information given in the PDB file. The copies will in that case have "s1", "s2", "s3" etc. at the end of the molecule names seen in the Project Tree.

If the proteins in the Molecule Project already are present in their biomolecule form, the message "The biological unit is already shown" is displayed, when the "Generate Biomolecule" button is clicked.

If the PDB files imported or downloaded to a Molecule Project did not hold biomolecule information, the message "No biological unit is associated with this Molecule Project" is shown, when the Generate Biomolecule button is clicked.

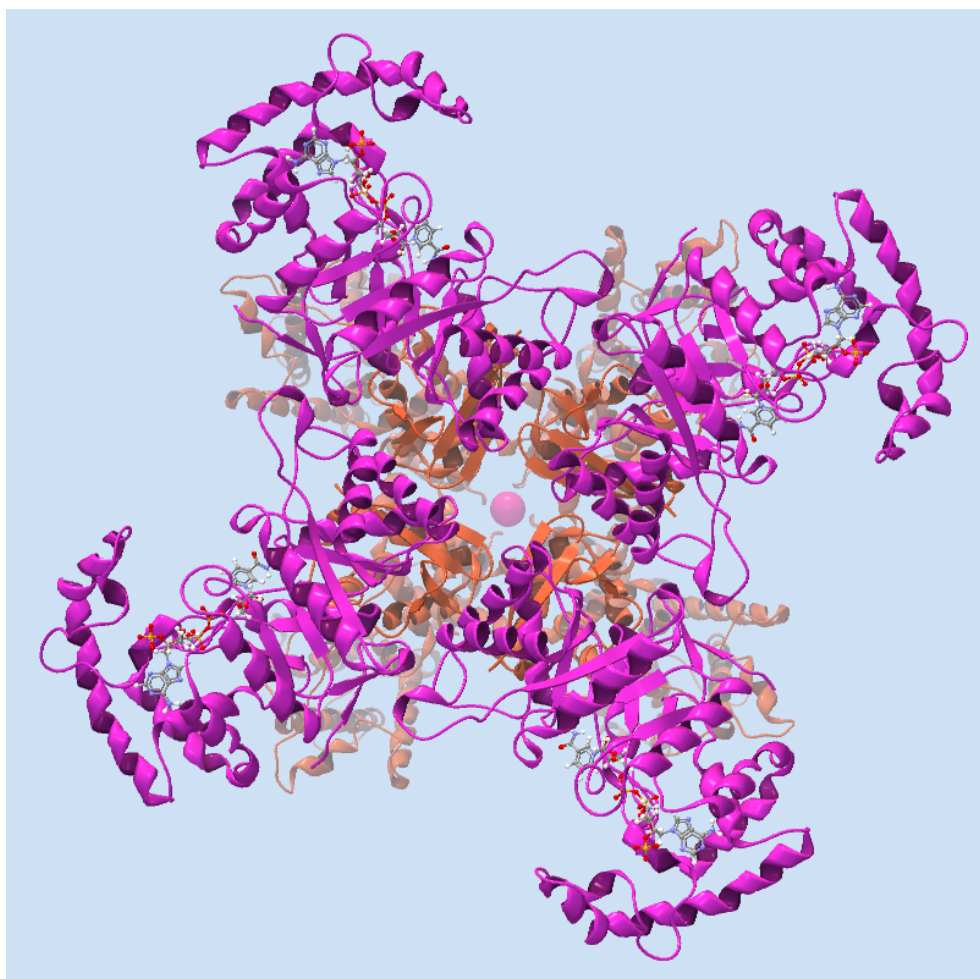


Figure 17.26: One of the biomolecules that can be generated after downloading the PDB 2R9R to **CLC Genomics Workbench**. It is a voltage gated potassium channel.

Chapter 18

General sequence analyses

Contents

18.1 Annotate with GFF/GTF/GVF file	463
18.2 Extract sequences	466
18.3 Shuffle sequence	467
18.4 Dot plots	469
18.4.1 Create dot plots	469
18.4.2 View dot plots	470
18.4.3 Bioinformatics explained: Dot plots	470
18.4.4 Bioinformatics explained: Scoring matrices	475
18.5 Local complexity plot	477
18.6 Sequence statistics	477
18.6.1 Bioinformatics explained: Protein statistics	480
18.7 Join Sequences	482
18.8 Pattern discovery	483
18.8.1 Pattern discovery search parameters	484
18.8.2 Pattern search output	485
18.9 Motif Search	485
18.9.1 Dynamic motifs	485
18.9.2 The Motif Search tool	487
18.9.3 Java regular expressions	489
18.10 Create motif list	490

CLC Genomics Workbench offers different kinds of sequence analyses that apply to both protein and DNA.

The analyses are described in this chapter.

18.1 Annotate with GFF/GTF/GVF file

Use **Annotate with GFF/GTF/GVF file** to add annotations from a GFF3, GTF or GVF file onto a sequence, or sequences in a sequence list. The names in the first column in the file must match

the names of the sequences to be annotated. If this is not the case, either the names in the annotation file, or the names of the sequences, must be updated.

Tools are available for renaming sequences or sequences in sequence lists:

- Rename Elements, described in section 37.14
- Rename Sequences in Lists, described in section 37.15

See <http://gmod.org/wiki/GFF3> for information about the GFF3 format and <https://mblab.wustl.edu/GTF22.html> for information on the GTF format.

Importing standard reference data

Before proceeding to use **Annotate with GFF/GTF/GVF file**, please refer to section 11. This covers how to download standard reference data, including annotations, provided by QIAGEN and other public sources using the Reference Data Manager, which is part of the Workbench. If the reference data you are interested in is available through the Reference Data Manager, it is usually easier to use that rather than taking the steps outlined in this section.

If the reference data you need is not available through the Reference Data Manager, and you are working with track-based data, please refer to section 7.2, rather than using **Annotate with GFF/GTF/GVF file**.

How annotations are applied

Annotations from each line in the annotation file are placed on the sequence with the name given in the first column. Special treatment is given to annotations of the types CDS, exon, mRNA, transcript and gene. For these, the following applies:

- A gene annotation is generated for each gene_id. The region annotated extends from the leftmost to the rightmost positions of all annotations that have the gene_id (gtf-style).
- CDS annotations that have the same transcriptID are joined to one CDS annotation (gtf-style). Similarly, CDS annotations that have the same parent are joined to one CDS annotation (gff-style).
- If there is more than one exon annotation with the same transcriptID these are joined to one mRNA annotation. If there is only one exon annotation with a particular transcriptID, and no CDS with this transcriptID, a transcript annotation is added instead of the exon annotation (gtf-style).
- Exon annotations that have the same mRNA as parent are joined to one mRNA annotation. Similarly, exon annotations that have the same transcript as parent, are joined to one transcript annotation (gff-style).

Note that genes and transcripts are linked by name only (not by position, ID etc).

Running the tool

To run the **Annotate with GFF/GTF/GVF file** tool, go to:

Tools | **Classical Sequence Analysis** (🖨️) | **Classical Sequence Analysis** (🖨️) | **General Sequence Analysis** (🖨️) | **Annotate with GFF/GTF/GVF file** (👉📄)

After selecting the sequence to annotate, the next step will look like that shown in figure 18.1.

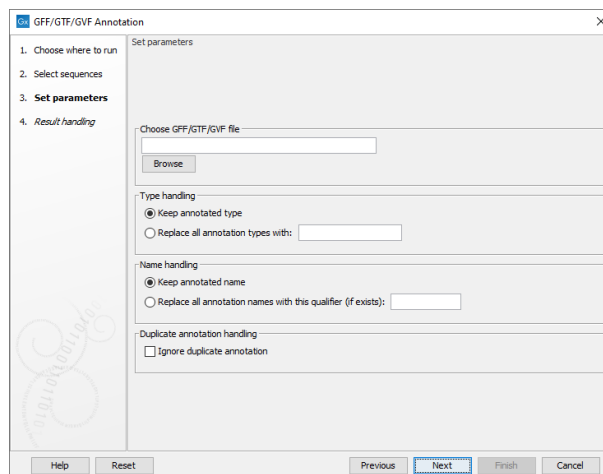


Figure 18.1: Select a GFF, GTF or GVR file by clicking on the Browse button.

Click on **Browse** to select a GFF, GTF or GVF file. After working through handling options, described below, your sequences will be annotated by the information from that file.

Name handling

Annotations are named in the following, prioritized way:

1. If one of the following qualifiers are present, it will be used for naming (prioritized):
 - (a) Name
 - (b) Gene_name
 - (c) Gene_ID
 - (d) Locus_tag
 - (e) ID
2. If none of these are found, the annotation type will be used as name.

You can overrule this naming convention by choosing **Replace all annotation names with this qualifier** and specifying another qualifier (see figure 18.2).

If you provide a qualiifier, it must be written *identically* to the corresponding qualifier name in the annotation file.

Transcript annotations are handled separately, since they inherit the name from the gene annotation.

Type handling

You can overrule feature types in the annotation file by choosing **Replace all annotation types with** and specifying a type to use.

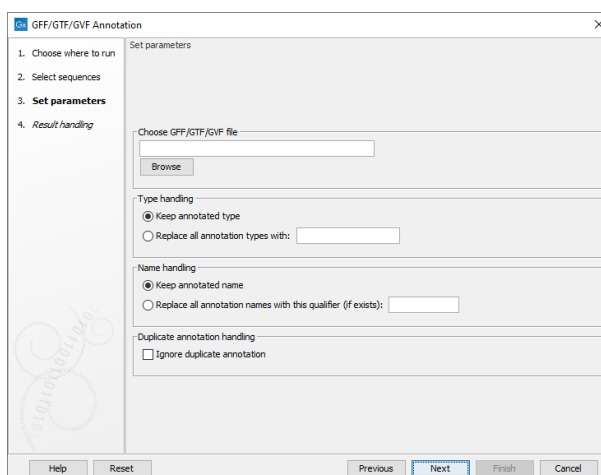


Figure 18.2: You can choose *Replace all annotation names with the specified qualifier*.

Ignore duplicate annotation

When the **Ignore duplicate annotation** option is checked, only one instance of duplicate annotations will be added to the sequence.

Create log

In the Result handling section of the wizard, check the **Create log** box results to create a log that includes information like the number of annotations found and if there are any that are could not be placed on the sequence. This information can help with troubleshooting when annotations are not added to a sequence when they were expected to be.

18.2 Extract sequences

Extract Sequences extracts all the sequences from any of the element types below into a sequence list or into individual sequence elements:

- Alignments (📄)
- BLAST result (📄) For BLAST results, the sequence hits are extracted but not the original query sequence or the consensus sequence.
- BLAST overview tables (📄)
- Contigs and read mappings (📄) For mappings, only the read sequences are extracted. Reference and consensus sequences are not extracted using this tool.
- Read mapping tables (📄)
- Read mapping tracks (📄)
- RNA-Seq mapping results (📄)
- Sequence lists (📄) See further notes below about running this tool on sequence lists.

If only a **subset** of the sequences are of interest, create an element containing just this subset first, and then run Extract Sequences on this. See the documentation for the relevant element types for further details. For example, for extracting a subset of a mapping, see section 22.7.6.

Paired reads are extracted in accordance with the read group settings, which are specified during the original import of the reads. If the orientation has since been changed (for example using the Element Info tab for the sequence list), the read group information will be modified and reads will be extracted as specified by the modified read group. The default read group orientation is forward-reverse.

Extracting sequences from sequence lists: As all sequences will be extracted, the main reason to run this tool on a sequence list would be if you wished to create individual sequence elements from each sequence in the list. This is somewhat uncommon. If your aim is to create a list containing a subset of the sequences from another list, this can be done directly from the table view of sequence lists (see section 15.1.3), or using Split Sequence List (see section 37.12).

Running Extract Sequences

Launch **Extract Sequences** by going to:

Tools | Classical Sequence Analysis (📁) | General Sequence Analysis (📁) | Extract Sequences (🔍)

After selecting the elements to extract sequences from, you are offered the choice of extracting them to individual sequence elements or to a sequence list (figure 18.3). For most data types, a sequence list will be the best choice.

Below these options, the number of sequences that will be extracted is reported.

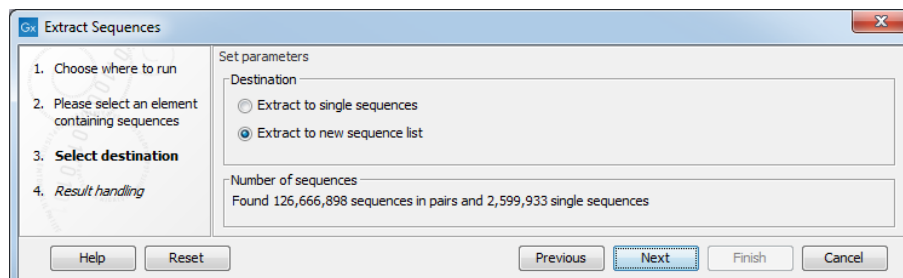


Figure 18.3: Extracted sequences can be put into a new sequence list or split into individual sequence elements.

18.3 Shuffle sequence

In some cases, it is beneficial to shuffle a sequence, for example, when for statistical analyses when comparing an alignment score with the distribution of scores of shuffled sequences.

Shuffling a sequence removes all annotations that relate to the residues. To launch the **Shuffle Sequence** tool, go to:

Tools | Classical Sequence Analysis (📁) | General Sequence Analysis (📁) | Shuffle Sequence (🔍)

Use the arrows to add or remove sequences or sequence lists from the selected elements list.

Click **Next** to determine how the shuffling should be performed.

In this step, shown in figure 18.4:

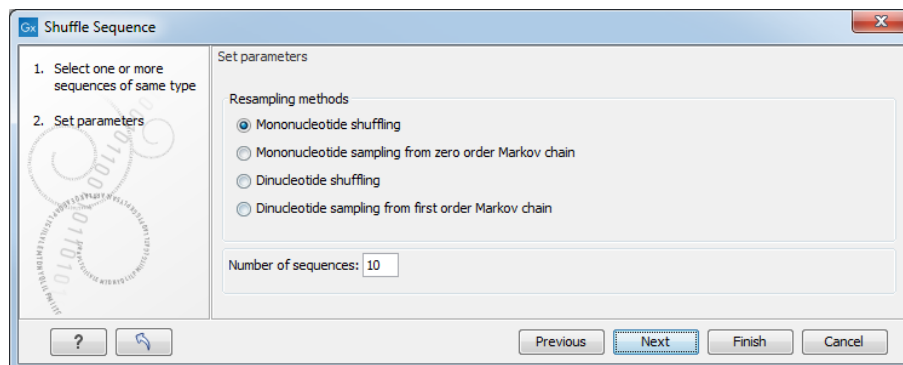


Figure 18.4: Parameters for shuffling.

For nucleotides, the following parameters can be set:

- **Mononucleotide shuffling.** Shuffle method generating a sequence of the exact same mononucleotide frequency
- **Dinucleotide shuffling.** Shuffle method generating a sequence of the exact same dinucleotide frequency
- **Mononucleotide sampling from zero order Markov chain.** Resampling method generating a sequence of the same expected mononucleotide frequency.
- **Dinucleotide sampling from first order Markov chain.** Resampling method generating a sequence of the same expected dinucleotide frequency.

For proteins, the following parameters can be set:

- **Single amino acid shuffling.** Shuffle method generating a sequence of the exact same amino acid frequency.
- **Single amino acid sampling from zero order Markov chain.** Resampling method generating a sequence of the same expected single amino acid frequency.
- **Dipeptide shuffling.** Shuffle method generating a sequence of the exact same dipeptide frequency.
- **Dipeptide sampling from first order Markov chain.** Resampling method generating a sequence of the same expected dipeptide frequency.

For further details of these algorithms, see [Clote et al., 2005]. In addition to the shuffle method, you can specify the number of randomized sequences to output.

Click **Finish** to start the tool.

This will open a new view in the **View Area** displaying the shuffled sequence. The new sequence is not saved automatically. To save the sequence, drag it into the **Navigation Area** or press ctrl + S (⌘ + S on Mac) to activate a save dialog.

18.4 Dot plots

Dot plots provide a powerful visual comparison of two sequences. Dot plots can also be used to compare regions of similarity within a sequence.

A dot plot is a simple, yet intuitive way of comparing two sequences, either DNA or protein, and is probably the oldest way of comparing two sequences [Maizel and Lenk, 1981]. A dot plot is a 2 dimensional matrix where each axis of the plot represents one sequence. By sliding a fixed size window over the sequences and making a sequence match by a dot in the matrix, a diagonal line will emerge if two identical (or very homologous) sequences are plotted against each other. Dot plots can also be used to visually inspect sequences for direct or inverted repeats or regions with low sequence complexity. Various smoothing algorithms can be applied to the dot plot calculation to avoid noisy background of the plot. Moreover, various substitution matrices can be applied in order to take the evolutionary distance of the two sequences into account.

18.4.1 Create dot plots

To create a dot plot, go to:

Tools | Classical Sequence Analysis (📁) | General Sequence Analysis (📁) | Create Dot Plot (🔗)

In the dialog that opens, select a sequence and click **Next** to adjust dot plot parameters (figure 18.5).

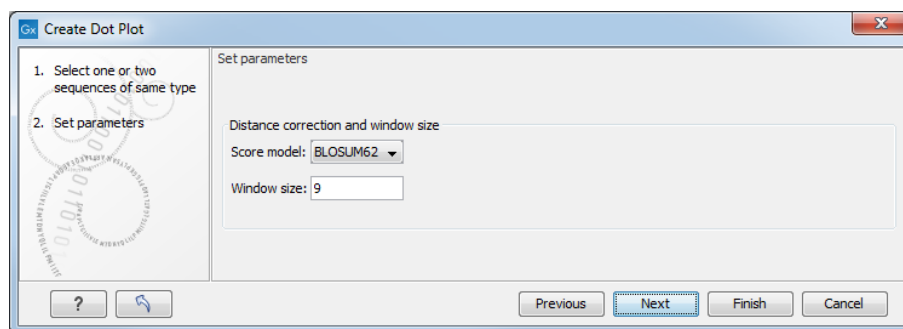


Figure 18.5: Setting the dot plot parameters.

There are two parameters for calculating the dot plot:

- **Distance correction (only valid for protein sequences)** In order to treat evolutionary transitions of amino acids, a distance correction measure can be used when calculating the dot plot. These distance correction matrices (substitution matrices) take into account the likeliness of one amino acid changing to another.
- **Window size** A residue by residue comparison (window size = 1) would undoubtedly result in a very noisy background due to a lot of similarities between the two sequences of interest. For DNA sequences the background noise will be even more dominant as a match between only four nucleotide is very likely to happen. Moreover, a residue by residue comparison (window size = 1) can be very time consuming and computationally demanding. Increasing the window size will make the dot plot more 'smooth'.

Note! Calculating dot plots takes up a considerable amount of memory in the computer. Therefore, you will see a warning message if the sum of the number of nucleotides/amino acids

in the sequences is higher than 8000. If you insist on calculating a dot plot with more residues the Workbench may shut down, but still allowing you to save your work first. However, this depends on your computer's memory configuration.

Click **Finish** to start the tool.

18.4.2 View dot plots

A view of a dot plot can be seen in figure 18.6. You can select **Zoom in** (🔍) in the Toolbar and click the dot plot to zoom in to see the details of particular areas.

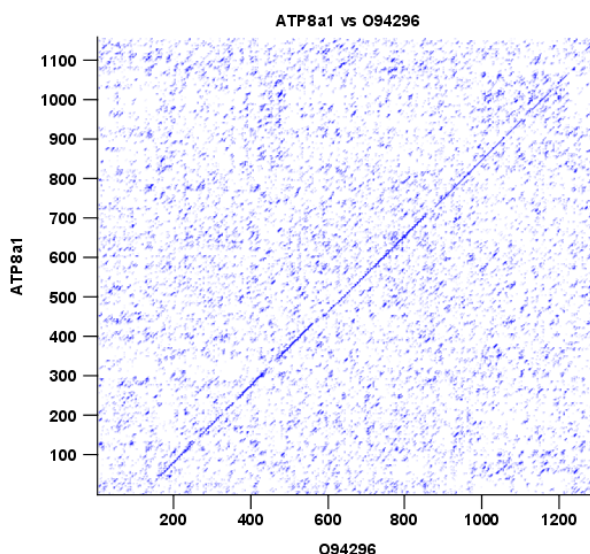


Figure 18.6: A view is opened showing the dot plot.

The **Side Panel** to the right let you specify the dot plot preferences. The gradient color box can be adjusted to get the appropriate result by dragging the small pointers at the top of the box. Moving the slider from the right to the left lowers the thresholds which can be directly seen in the dot plot, where more diagonal lines will emerge. You can also choose another color gradient by clicking on the gradient box and choose from the list.

Adjusting the sliders above the gradient box is also practical, when producing an output for printing where too much background color might not be desirable. By crossing one slider over the other (the two sliders change side) the colors are inverted, allowing for a white background (figure 18.7).

18.4.3 Bioinformatics explained: Dot plots

Dot plots are two-dimensional plots where the x-axis and y-axis each represents a sequence and the plot itself shows a comparison of these two sequences by a calculated score for each position of the sequence. If a window of fixed size on one sequence (one axis) match to the other sequence a dot is drawn at the plot. Dot plots are one of the oldest methods for comparing two sequences [Maizel and Lenk, 1981].

The scores that are drawn on the plot are affected by several issues.

- Scoring matrix for distance correction.

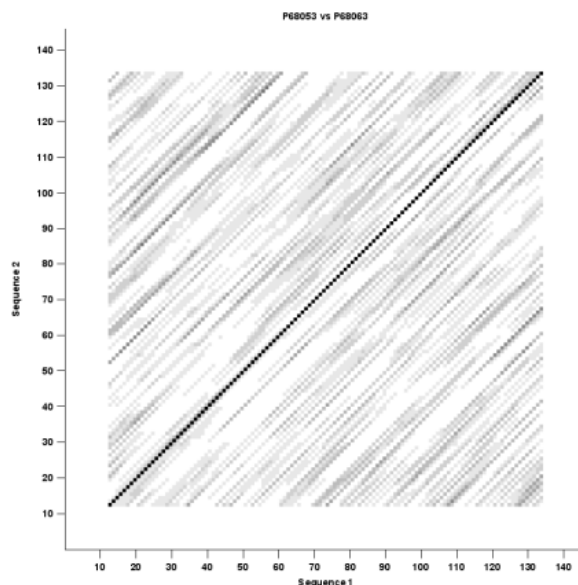


Figure 18.7: Dot plot with inverted colors, practical for printing.

Scoring matrices (BLOSUM and PAM) contain substitution scores for every combination of two amino acids. Thus, these matrices can only be used for dot plots of protein sequences.

- Window size

The single residue comparison (bit by bit comparison (window size = 1)) in dot plots will undoubtedly result in a noisy background of the plot. You can imagine that there are many successes in the comparison if you only have four possible residues like in nucleotide sequences. Therefore you can set a window size which is smoothing the dot plot. Instead of comparing single residues it compares subsequences of length set as window size. The score is now calculated with respect to aligning the subsequences.

- Threshold

The dot plot shows the calculated scores with colored threshold. Hence you can better recognize the most important similarities.

Examples and interpretations of dot plots

Contrary to simple sequence alignments dot plots can be a very useful tool for spotting various evolutionary events which may have happened to the sequences of interest.

Below is shown some examples of dot plots where sequence insertions, low complexity regions, inverted repeats etc. can be identified visually.

Similar sequences The most simple example of a dot plot is obtained by plotting two homologous sequences of interest. If very similar or identical sequences are plotted against each other a diagonal line will occur.

The dot plot in figure 18.8 shows two related sequences of the Influenza A virus nucleoproteins infecting ducks and chickens. Accession numbers from the two sequences are: DQ232610 and DQ023146. Both sequences can be retrieved directly from <http://www.ncbi.nlm.nih.gov/gquery/gquery.fcgi>.

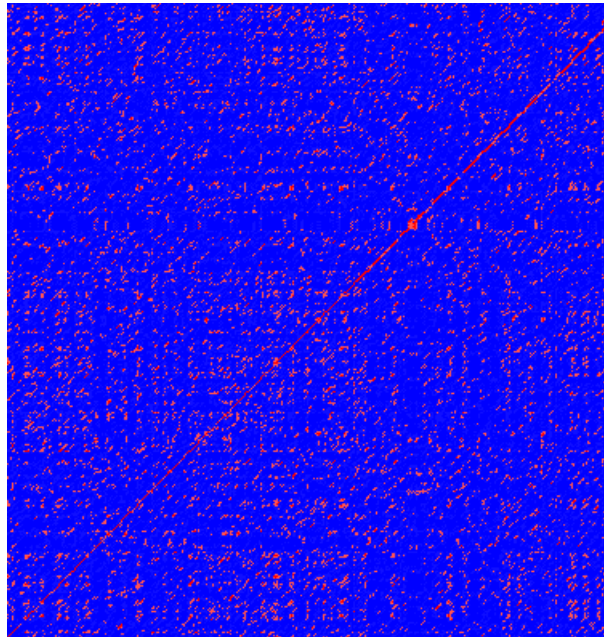


Figure 18.8: Dot plot of DQ232610 vs. DQ023146 (*Influenza A virus nucleoproteins*) showing and overall similarity

Repeated regions Sequence repeats can also be identified using dot plots. A repeat region will typically show up as lines parallel to the diagonal line.

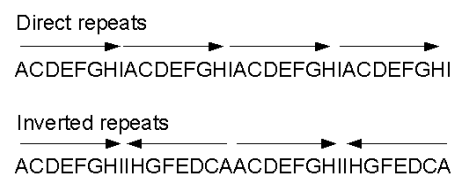


Figure 18.9: Direct and inverted repeats shown on an amino acid sequence generated for demonstration purposes.

If the dot plot shows more than one diagonal in the same region of a sequence, the regions depending to the other sequence are repeated. In figure 18.10 you can see a sequence with repeats.

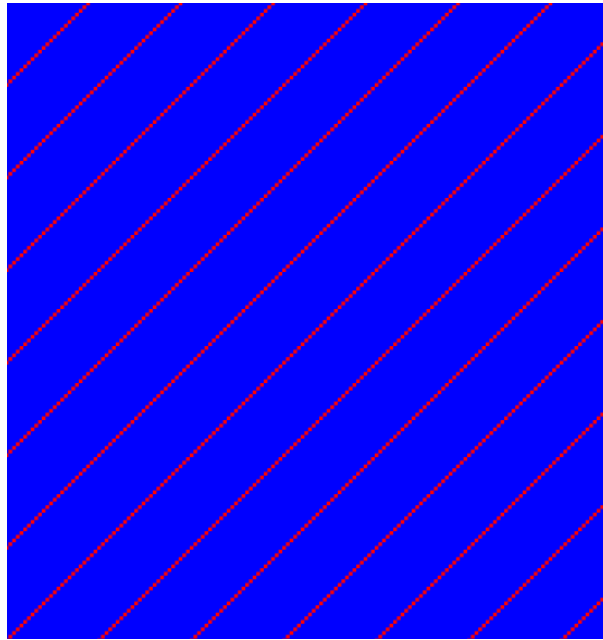


Figure 18.10: *The dot plot of a sequence showing repeated elements. See also figure 18.9.*

Frame shifts Frame shifts in a nucleotide sequence can occur due to insertions, deletions or mutations. Such frame shifts can be visualized in a dot plot as seen in figure 18.11. In this figure, three frame shifts for the sequence on the y-axis are found.

1. Deletion of nucleotides
2. Insertion of nucleotides
3. Mutation (out of frame)

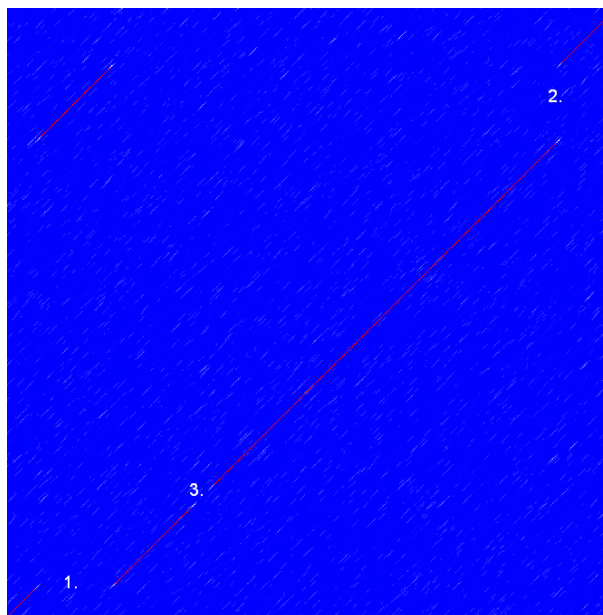


Figure 18.11: *This dot plot show various frame shifts in the sequence. See text for details.*

Sequence inversions In dot plots you can see an inversion of sequence as contrary diagonal to the diagonal showing similarity. In figure 18.12 you can see a dot plot (window length is 3) with an inversion.

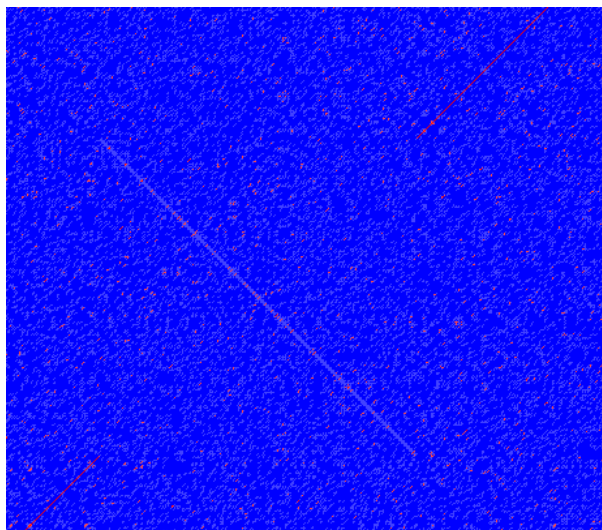


Figure 18.12: The dot plot showing an inversion in a sequence. See also figure 18.9.

Low-complexity regions Low-complexity regions in sequences can be found as regions around the diagonal all obtaining a high score. Low complexity regions are calculated from the redundancy of amino acids within a limited region [Wootton and Federhen, 1993]. These are most often seen as short regions of only a few different amino acids. In the middle of figure 18.13 is a square shows the low-complexity region of this sequence.

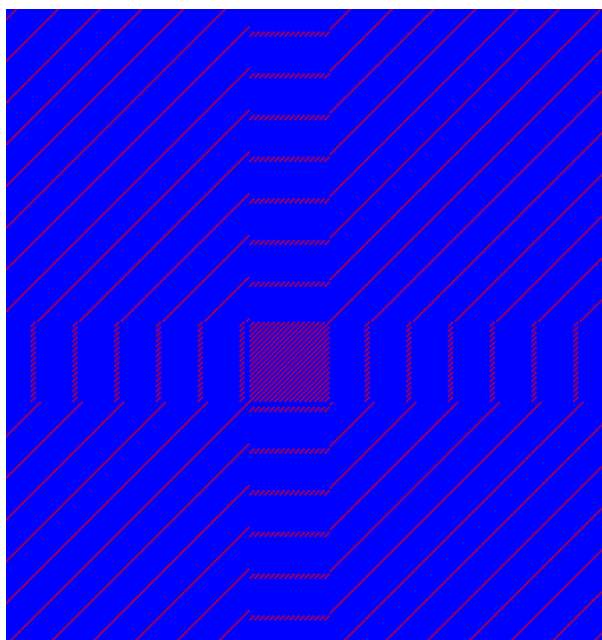


Figure 18.13: The dot plot showing a low-complexity region in the sequence. The sequence is artificial and low complexity regions do not always show as a square.

18.4.4 Bioinformatics explained: Scoring matrices

Biological sequences have evolved throughout time and evolution has shown that not all changes to a biological sequence is equally likely to happen. Certain amino acid substitutions (change of one amino acid to another) happen often, whereas other substitutions are very rare. For instance, tryptophan (W) which is a relatively rare amino acid, will only – on very rare occasions – mutate into a leucine (L).

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

Table 18.1: **The BLOSUM62 matrix.** A tabular view of the BLOSUM62 matrix containing all possible substitution scores [Henikoff and Henikoff, 1992].

Based on evolution of proteins it became apparent that these changes or substitutions of amino acids can be modeled by a scoring matrix also refereed to as a substitution matrix. See an example of a scoring matrix in table 18.1. This matrix lists the substitution scores of every single amino acid. A score for an aligned amino acid pair is found at the intersection of the corresponding column and row. For example, the substitution score from an arginine (R) to a lysine (K) is 2. The diagonal show scores for amino acids which have not changed. Most substitutions changes have a negative score. Only rounded numbers are found in this matrix.

The two most used matrices are the BLOSUM [Henikoff and Henikoff, 1992] and PAM [Dayhoff and Schwartz, 1978].

Different scoring matrices

• PAM

The first PAM matrix (Point Accepted Mutation) was published in 1978 by Dayhoff et al. The PAM matrix was build through a global alignment of related sequences all having sequence similarity above 85% [Dayhoff and Schwartz, 1978]. A PAM matrix shows the probability that any given amino acid will mutate into another in a given time interval. As an example, PAM1 gives that one amino acid out of a 100 will mutate in a given time interval. In the other end of the scale, a PAM256 matrix, gives the probability of 256 mutations in a 100 amino acids (see figure 18.14).

There are some limitation to the PAM matrices which makes the BLOSUM matrices somewhat more attractive. The dataset on which the initial PAM matrices were build is very old by now, and the PAM matrices assume that all amino acids mutate at the same rate - this is not a correct assumption.

- **BLOSUM**

In 1992, 14 years after the PAM matrices were published, the BLOSUM matrices (BLOcks Substitution Matrix) were developed and published [Henikoff and Henikoff, 1992].

Henikoff et al. wanted to model more divergent proteins, thus they used locally aligned sequences where none of the aligned sequences share less than 62% identity. This resulted in a scoring matrix called BLOSUM62. In contrast to the PAM matrices the BLOSUM matrices are calculated from alignments without gaps emerging from the BLOCKS database <http://blocks.fhcrc.org/>.

Sean Eddy recently wrote a paper reviewing the BLOSUM62 substitution matrix and how to calculate the scores [Eddy, 2004].

Use of scoring matrices Deciding which scoring matrix you should use in order of obtain the best alignment results is a difficult task. If you have no prior knowledge on the sequence the BLOSUM62 is probably the best choice. This matrix has become the *de facto* standard for scoring matrices and is also used as the default matrix in BLAST searches. The selection of a "wrong" scoring matrix will most probable strongly influence on the outcome of the analysis. In general a few rules apply to the selection of scoring matrices.

- For closely related sequences choose BLOSUM matrices created for highly similar alignments, like BLOSUM80. You can also select low PAM matrices such as PAM1.
- For distant related sequences, select low BLOSUM matrices (for example BLOSUM45) or high PAM matrices such as PAM250.

The BLOSUM matrices with low numbers correspond to PAM matrices with high numbers. (See figure 18.14) for correlations between the PAM and BLOSUM matrices. To summarize, if you want to find distant related proteins to a sequence of interest using BLAST, you could benefit of using BLOSUM45 or similar matrices.

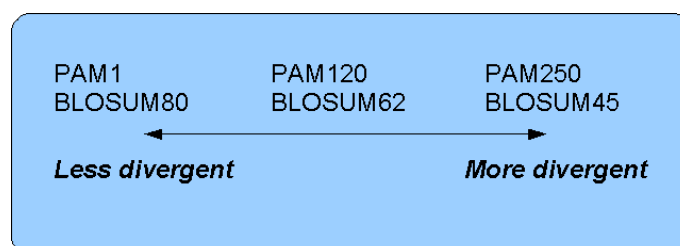


Figure 18.14: Relationship between scoring matrices. The BLOSUM62 has become a *de facto* standard scoring matrix for a wide range of alignment programs. It is the default matrix in BLAST.

Other useful resources BLOKS database

<http://blocks.fhcrc.org/>

NCBI help site

http://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs

18.5 Local complexity plot

In *CLC Genomics Workbench* it is possible to calculate local complexity for both DNA and protein sequences. The local complexity is a measure of the diversity in the composition of amino acids within a given range (window) of the sequence. The K2 algorithm is used for calculating local complexity [Wootton and Federhen, 1993]. To conduct a complexity calculation do the following:

Tools | Classical Sequence Analysis (📁) | **General Sequence Analysis** (📁) | **Create Complexity Plot** (📈)

This opens a dialog. In **Step 1** you can use the arrows to change, remove and add DNA and protein sequences in the **Selected Elements** window.

When the relevant sequences are selected, clicking **Next** takes you to **Step 2**. This step allows you to adjust the window size from which the complexity plot is calculated. Default is set to 11 amino acids and the number should always be odd. The higher the number, the less volatile the graph.

Figure 18.15 shows an example of a local complexity plot.

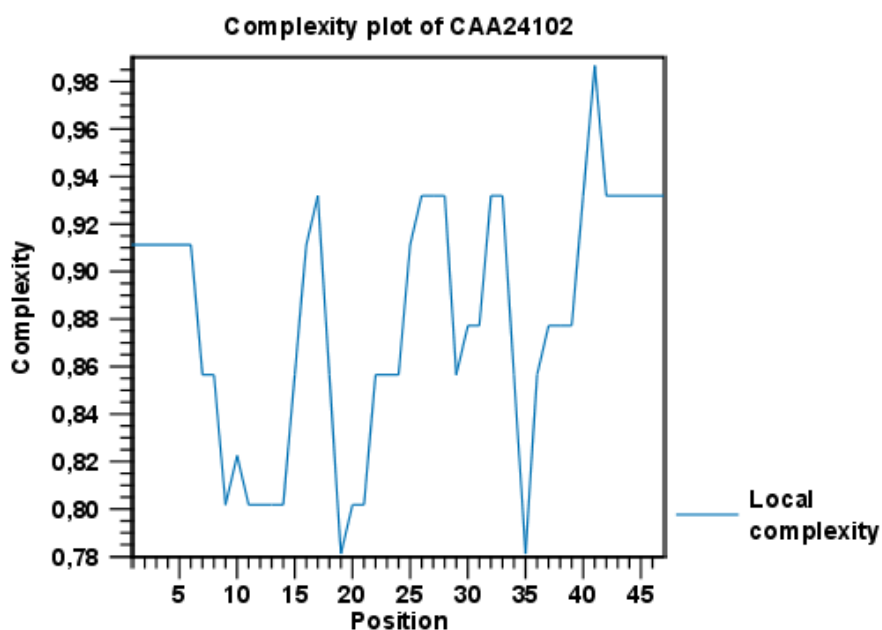


Figure 18.15: An example of a local complexity plot.

Click **Finish** to start the tool. The values of the complexity plot approaches 1.0 as the distribution of amino acids become more complex.

See section B in the appendix for information about the graph view.

18.6 Sequence statistics

CLC Genomics Workbench can produce an output with many relevant statistics for protein sequences. Some of the statistics are also relevant to produce for DNA sequences. Therefore, this section deals with both types of statistics. The required steps for producing the statistics are the same.

To create a statistic for the sequence, do the following:

Tools | Classical Sequence Analysis (📁) | General Sequence Analysis (📁) | Create Sequence Statistics (📊)

Select one or more sequence(s) or/and one or more sequence list(s). **Note!** You cannot create statistics for DNA and protein sequences at the same time, they must be run separately.

Next (figure 18.16), the dialog offers to adjust the following parameters:

- **Individual statistics layout.** If more sequences were selected in **Step 1**, this function generates separate statistics report for each sequence.
- **Comparative statistics layout.** If more sequences were selected in **Step 1**, this function generates statistics with comparisons between the sequences.

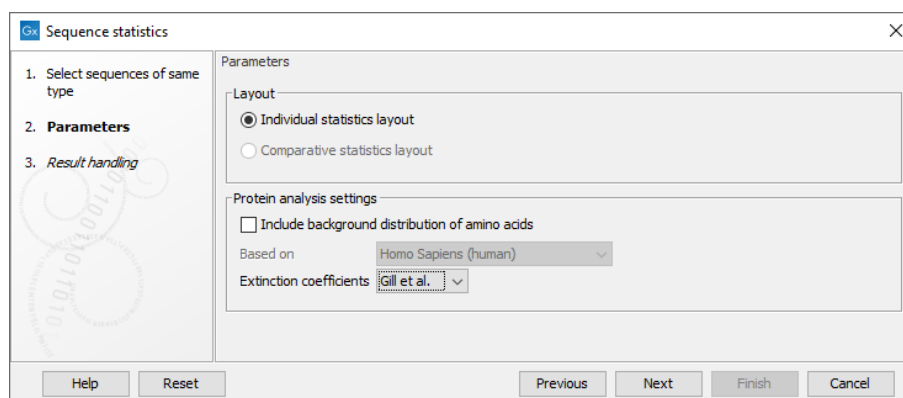


Figure 18.16: Setting parameters for the Sequence statistics tool.

For protein sequences, you can choose to include Background distribution of amino acids. If this box is ticked, an extra column with amino acid distribution of the chosen species, is included in the table output. (The distributions are calculated from UniProt www.uniprot.org version 6.0, dated September 13 2005.)

You can also choose between two different sets of values for calculation of extinction coefficients:

- [Gill and von Hippel, 1989]: Ext(Cystine) = 120, Ext(Tyr) = 1280 and Ext(Trp) = 5690
- [Pace et al., 1995]: Ext(Cystine) = 125, Ext(Tyr) = 1490 and Ext(Trp) = 5500

Read more about calculation of extinction coefficients in section section 18.6.1.

Click **Finish** to start the tool. An example of protein sequence statistics is shown in figure 18.17.

Nucleotide sequence statistics are generated using the same dialog as used for protein sequence statistics. However, the output of Nucleotide sequence statistics is less extensive than that of the protein sequence statistics.

Note! The headings of the tables change depending on whether you calculate individual or comparative sequence statistics.

The output of protein sequence statistics includes:

- **Sequence Information:**

1.1 Sequence information	
Sequence type	Protein
Length	147aa
Organism	Mus musculus
Name	HBB0_MOUSE
Description	RecName: Full=Hemoglobin subunit beta-H0; AltName: Full=Beta-H0-globin; AltName: Full=Hemoglobin beta-H0 chain
Modification Date	23-JAN-2007
Weight	16.384 kDa
Isoelectric point	9.08
Aliphatic index	95.578

Figure 18.17: Example of protein sequence statistics.

- Sequence type
- Length
- Organism
- Name
- Description
- Modification Date
- Weight. This is calculated like this: $\sum_{units\ in\ sequence} (weight(unit)) - links * weight(H_2O)$ where `links` is the sequence length minus one and `units` are amino acids. The atomic composition is defined the same way.
- Isoelectric point
- Aliphatic index
- Amino acid counts, frequencies
- Annotation counts

The output of nucleotide sequence statistics include:

- General statistics:
 - Sequence type
 - Length
 - Organism
 - Name
 - Description
 - Modification Date
 - Weight (calculated as single-stranded and double-stranded DNA)
- Annotation table
- Nucleotide distribution table

If nucleotide sequences are used as input, and these are annotated with CDS, a section on codon statistics for coding regions is included. This represents statistics for all codons; however, only codons that contribute with amino acids to the translated sequence will be counted.

A short description of the different areas of the statistical output is given in section [18.6.1](#).

18.6.1 Bioinformatics explained: Protein statistics

Every protein holds specific and individual features which are unique to that particular protein. Features such as isoelectric point or amino acid composition can reveal important information of a novel protein. Many of the features described below are calculated in a simple way.

- **Molecular weight** The molecular weight is the mass of a protein or molecule. The molecular weight is simply calculated as the sum of the atomic mass of all the atoms in the molecule.

The weight of a protein is usually represented in Daltons (Da).

A calculation of the molecular weight of a protein does not usually include additional post-translational modifications. For native and unknown proteins it tends to be difficult to assess whether posttranslational modifications such as glycosylations are present on the protein, making a calculation based solely on the amino acid sequence inaccurate. The molecular weight can be determined very accurately by mass-spectrometry in a laboratory.

- **Isoelectric point** The isoelectric point (pI) of a protein is the pH where the proteins has no net charge. The pI is calculated from the pKa values for 20 different amino acids. At a pH below the pI, the protein carries a positive charge, whereas if the pH is above pI the proteins carry a negative charge. In other words, pI is high for basic proteins and low for acidic proteins. This information can be used in the laboratory when running electrophoretic gels. Here the proteins can be separated, based on their isoelectric point.

- **Aliphatic index** The aliphatic index of a protein is a measure of the relative volume occupied by aliphatic side chain of the following amino acids: alanine, valine, leucine and isoleucine. An increase in the aliphatic index increases the thermostability of globular proteins. The index is calculated by the following formula.

$$\text{Aliphatic index} = X(\text{Ala}) + a * X(\text{Val}) + b * X(\text{Leu}) + b * (X)\text{Ile}$$

$X(\text{Ala})$, $X(\text{Val})$, $X(\text{Ile})$ and $X(\text{Leu})$ are the amino acid compositional fractions. The constants a and b are the relative volume of valine (a=2.9) and leucine/isoleucine (b=3.9) side chains compared to the side chain of alanine [Ikai, 1980].

- **Estimated half-life** The half life of a protein is the time it takes for the protein pool of that particular protein to be reduced to the half. The half life of proteins is highly dependent on the presence of the N-terminal amino acid, thus overall protein stability [Bachmair et al., 1986, Gonda et al., 1989, Tobias et al., 1991]. The importance of the N-terminal residues is generally known as the 'N-end rule'. The N-end rule and consequently the N-terminal amino acid, simply determines the half-life of proteins. The estimated half-life of proteins have been investigated in mammals, yeast and *E. coli* (see Table 18.2). If leucine is found N-terminally in mammalian proteins the estimated half-life is 5.5 hours.
- **Extinction coefficient** This measure indicates how much light is absorbed by a protein at a particular wavelength. The extinction coefficient is measured by UV spectrophotometry, but can also be calculated. The amino acid composition is important when calculating the extinction coefficient. The extinction coefficient is calculated from the absorbance of cysteine, tyrosine and tryptophan.

Two values are reported. The first value, "Non-reduced cysteines", is computed assuming that all cysteine residues appear as half cystines, meaning they form di-sulfide bridges to other cysteines:

Amino acid	Mammalian	Yeast	E. coli
Ala (A)	4.4 hour	>20 hours	>10 hours
Cys (C)	1.2 hours	>20 hours	>10 hours
Asp (D)	1.1 hours	3 min	>10 hours
Glu (E)	1 hour	30 min	>10 hours
Phe (F)	1.1 hours	3 min	2 min
Gly (G)	30 hours	>20 hours	>10 hours
His (H)	3.5 hours	10 min	>10 hours
Ile (I)	20 hours	30 min	>10 hours
Lys (K)	1.3 hours	3 min	2 min
Leu (L)	5.5 hours	3 min	2 min
Met (M)	30 hours	>20 hours	>10 hours
Asn (N)	1.4 hours	3 min	>10 hours
Pro (P)	>20 hours	>20 hours	?
Gln (Q)	0.8 hour	10 min	>10 hours
Arg (R)	1 hour	2 min	2 min
Ser (S)	1.9 hours	>20 hours	>10 hours
Thr (T)	7.2 hours	>20 hours	>10 hours
Val (V)	100 hours	>20 hours	>10 hours
Trp (W)	2.8 hours	3 min	2 min
Tyr (Y)	2.8 hours	10 min	2 min

Table 18.2: **Estimated half life.** Half life of proteins where the N-terminal residue is listed in the first column and the half-life in the subsequent columns for mammals, yeast and *E. coli*.

$$\text{Ext}(Protein) = \frac{\text{count}(Cys)}{2} \cdot \text{Ext}(Cys) + \text{count}(Tyr) \cdot \text{Ext}(Tyr) + \text{count}(Trp) \cdot \text{Ext}(Trp).$$

The second value, "Reduced cysteines", assumes that no di-sulfide bonds are formed:

$$\text{Ext}(Protein) = \text{count}(Tyr) \cdot \text{Ext}(Tyr) + \text{count}(Trp) \cdot \text{Ext}(Trp).$$

The extinction coefficient values of the three important amino acids at different wavelengths are found in [Gill and von Hippel, 1989] or in [Pace et al., 1995]. At 280nm the extinction coefficients are

- [Gill and von Hippel, 1989]: Ext(Cystine) = 120, Ext(Tyr) = 1280 and Ext(Trp) = 5690
- [Pace et al., 1995]: Ext(Cystine) = 125, Ext(Tyr) = 1490 and Ext(Trp) = 5500

This equation is only valid under the following conditions:

- pH 6.5
- 6.0 M guanidium hydrochloride
- 0.02 M phosphate buffer




Knowing the extinction coefficient, the absorbance (optical density) can be calculated using the following formula: $\text{Absorbance}(Protein) = \frac{\text{Ext}(Protein)}{\text{Molecular weight}}$

- **Atomic composition** Amino acids are indeed very simple compounds. All 20 amino acids consist of combinations of only five different atoms. The atoms which can be found in these simple structures are: Carbon, Nitrogen, Hydrogen, Sulfur, Oxygen. The atomic composition of a protein can for example be used to calculate the precise molecular weight of the entire protein.
- **Total number of negatively charged residues (Asp + Glu)** At neutral pH, the fraction of negatively charged residues provides information about the location of the protein. Intracellular proteins tend to have a higher fraction of negatively charged residues than extracellular proteins.
- **Total number of positively charged residues (Arg + Lys)** At neutral pH, nuclear proteins have a high relative percentage of positively charged amino acids. Nuclear proteins often bind to the negatively charged DNA, which may regulate gene expression or help to fold the DNA. Nuclear proteins often have a low percentage of aromatic residues [Andrade et al., 1998].
- **Amino acid distribution** Amino acids are the basic components of proteins. The amino acid distribution in a protein is simply the percentage of the different amino acids represented in a particular protein of interest. Amino acid composition is generally conserved through family-classes in different organisms which can be useful when studying a particular protein or enzymes across species borders. Another interesting observation is that amino acid composition variate slightly between proteins from different subcellular localizations. This fact has been used in several computational methods, used for prediction of subcellular localization.
- **Annotation table** This table provides an overview of all the different annotations associated with the sequence and their incidence.
- **Dipeptide distribution** This measure is simply a count, or frequency, of all the observed adjacent pairs of amino acids (dipeptides) found in the protein. It is only possible to report neighboring amino acids. Knowledge on dipeptide composition have previously been used for prediction of subcellular localization.

18.7 Join Sequences

CLC Genomics Workbench can join several nucleotide or protein sequences into one sequence. This feature can for example be used to construct "supergenes" for phylogenetic inference by joining several disjoint genes into one. Note, that when sequences are joined, all their annotations are carried over to the new spliced sequence.

Two (or more) sequences can be joined using the **Join Sequences** tool, available at:

Tools | Classical Sequence Analysis  | **General Sequence Analyses**  | **Join Sequences** 

This opens the dialog shown in figure 18.18.

Use the arrows to add or remove sequences from the selected elements list.

In the next wizard step, you can define the order in which the sequences should be joined (figure 18.19).

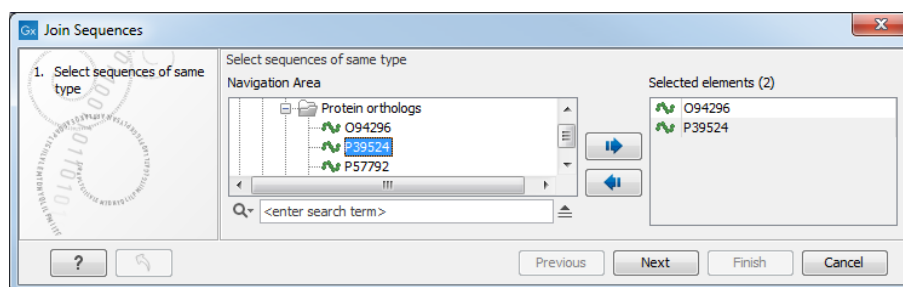


Figure 18.18: Selecting two sequences to be joined.

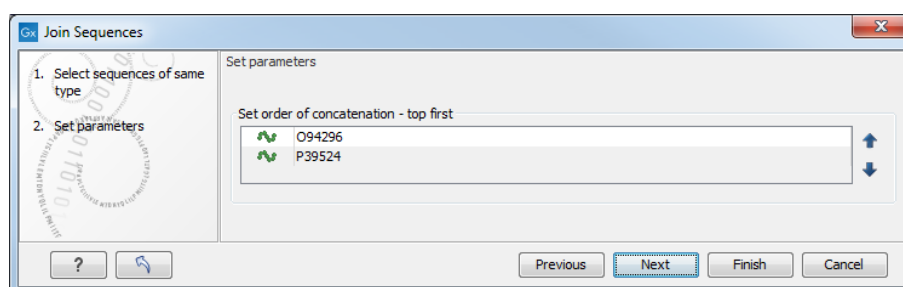


Figure 18.19: Setting the order in which sequences are joined.

In step 2 you can change the order in which the sequences will be joined. Select a sequence and use the arrows to move the selected sequence up or down.

Click **Finish** to start the tool.

The result is shown in figure 18.20.

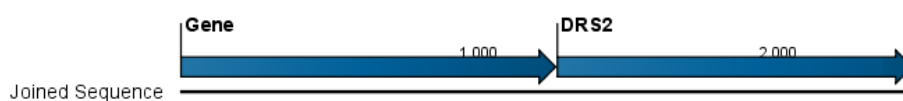


Figure 18.20: The result of joining sequences is a new sequence containing the annotations of the joined sequences (they each had a HBB annotation).

18.8 Pattern discovery

With *CLC Genomics Workbench* you can perform pattern discovery on both DNA and protein sequences. Advanced hidden Markov models can help to identify unknown sequence patterns across single or even multiple sequences.

In order to search for unknown patterns:

Tools | Classical Sequence Analysis (📁) | General Sequence Analysis (📁) | Pattern Discovery (🔍)

Choose one or more sequence(s) or sequence list(s). You can perform the analysis on several DNA or several protein sequences at a time. If the analysis is performed on several sequences at a time the method will search for patterns which is common between all the sequences. Annotations will be added to all the sequences and a view is opened for each sequence.

Click **Next** to adjust parameters (see figure 18.21).

In order to search unknown sequences with an already existing model:

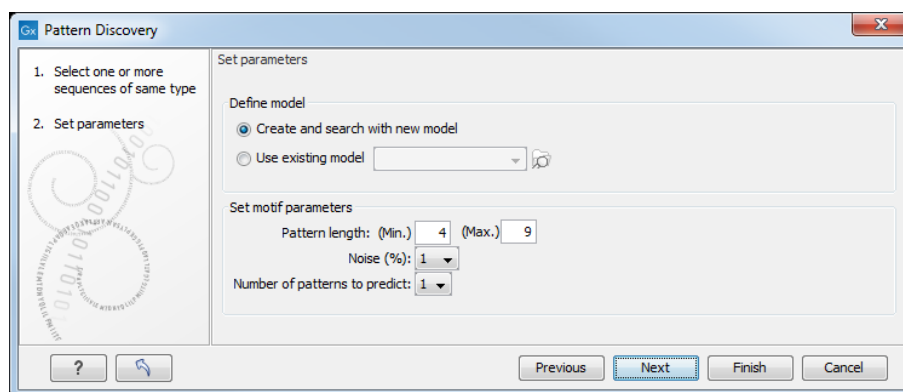


Figure 18.21: Setting parameters for the pattern discovery. See text for details.

Select to use an already existing model which is seen in figure 18.21. Models are represented with the following icon in the **Navigation Area**

18.8.1 Pattern discovery search parameters

Various parameters can be set prior to the pattern discovery. The parameters are listed below and a screenshot of the parameter settings can be seen in figure 18.21.

- **Create and search with new model.** This will create a new HMM model based on the selected sequences. The found model will be opened after the run and presented in a table view. It can be saved and used later if desired.
- **Use existing model.** It is possible to use already created models to search for the same pattern in new sequences.
- **Minimum pattern length.** Here, the minimum length of patterns to search for, can be specified.
- **Maximum pattern length.** Here, the maximum length of patterns to search for, can be specified.
- **Noise (%).** Specify noise-level of the model. This parameter has influence on the level of degeneracy of patterns in the sequence(s). The noise parameter can be 1,2,5 or 10 percent.
- **Number of different kinds of patterns to predict.** Number of iterations the algorithm goes through. After the first iteration, we force predicted pattern-positions in the first run to be member of the background: In that way, the algorithm finds new patterns in the second iteration. Patterns marked 'Pattern1' have the highest confidence. The maximal iterations to go through is 3.
- **Include background distribution.** For protein sequences it is possible to include information on the background distribution of amino acids from a range of organisms.

Click **Finish** to start the tool. This will open a view showing the patterns found as annotations on the original sequence (see figure 18.22). If you have selected several sequences, a corresponding number of views will be opened.



Figure 18.22: Sequence view displaying two discovered patterns.

18.8.2 Pattern search output

If the analysis is performed on several sequences at a time the method will search for patterns in the sequences and open a new view for each of the sequences, in which a pattern was discovered. Each novel pattern will be represented as an annotation of the type **Region**. More information on each found pattern is available through the tool-tip, including detailed information on the position of the pattern and quality scores.

It is also possible to get a tabular view of all found patterns in one combined table. Then each found pattern will be represented with various information on obtained scores, quality of the pattern and position in the sequence.

A table view of emission values of the actual used HMM model is presented in a table view. This model can be saved and used to search for a similar pattern in new or unknown sequences.

18.9 Motif Search

Known motifs in nucleotide or peptide sequences can be searched for using a specific sequence or, more flexibly, by using a regular expression.

There are two ways to search for known motifs:

- **Search in an open sequence** Common motifs and custom motifs can be quickly scanned for and visualized on a sequence while working with that sequence interactively. See section 18.9.1 for details.
- **Use the Motif Search tool** A more refined and systematic search for motifs can be performed using the **Motif Search** tool. This generates a table and can optionally add annotations to the sequences. See section 18.9.2 for details.

18.9.1 Dynamic motifs

In the Side Panel settings of a sequence that is open for viewing, there is a **Motifs** palette (figure 18.23). Using these options, motifs can be added to or removed from the list, as well as moved up and down in the list. All listed motifs are searched for, and the number of instances found is reported in brackets by the motif name.

If the box by a particular motif is checked, that motif will be highlighted on the sequence itself (figure 18.24). By default, a motif is shown as a faded arrow, where the direction of the arrow indicates the strand of the motif. Hovering the mouse cursor over a motif on the sequence reveals information about the motif (figure 18.25).

To add **Labels** to the motif, select the **Flag** or **Stacked** option. They will put the name of the motif as a flag above the sequence. The stacked option will stack the labels when there is more than one motif so that all labels are shown.

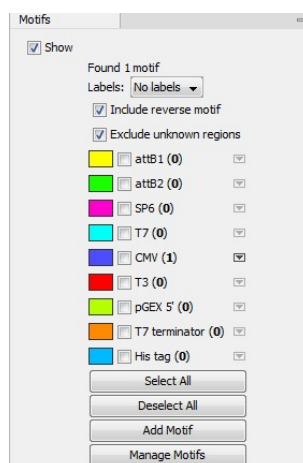


Figure 18.23: The Motifs palette of the Side Panel of an open sequence. A single instance of the CMV motif has been detected.

```

      380      400      420
CCCATTGACGTC AATGGGAGTTTGT TTTGGCACCAAAATCAACGGGACTTTCC
      440      460
AAAATGTCGTAACAACTCCGCCCCATTGACGCAAAATGGGCGGTAGGCGTGTAC
      480      500      520
GGTGGGAGGTCTATATAAGCAGAGCTCGTTTAGTGAACCGTCAGATCGCCTGG
  
```

Figure 18.24: When the box next to a motif type is checked, any instances of that motif in a sequence will be highlighted in the view.

```

3CCCCATTGACGCAAAATGGGCGGTAGGCGTGTACGGTGGGAGG
  
```

/motif=CGCAAAATGGGCGGTAGGCGTG, list index: 5 (CMV);
 /type=Simple
 /description=CMV promoter primer

Figure 18.25: Hover the mouse cursor over a motif region on the sequence to reveal a tool tip with information about the motif.

Below the labels option there are two options for controlling the way the sequence should be searched for motifs:

- **Include reverse motifs.** This will also find motifs on the negative strand (only available for nucleotide sequences)
- **Exclude matches in N-regions for simple motifs.** The motif search handles ambiguous characters in the way that two residues are different if they do not have any residues in common. For example: For nucleotides, *N* matches any character and *R* matches A,G. For proteins, *X* matches any character and *Z* matches E,Q. Genome sequence often have large regions with unknown sequence. These regions are very often padded with N's. Ticking this checkbox will not display hits found in N-regions and if a one residue in a motif matches to an N, it will be treated as a mismatch.

The list of motifs shown in figure 18.23 is a pre-defined list that is included with the workbench, but you can define your own set of motifs to use instead. In order to do this, you can either

launch the Create Motif List tool from the Navigation Area or using the **Add Motif** button in the Side Panel (see section 18.10). Once your list of custom motif(s) is saved, you can click the **Manage Motifs** button in the Side Panel, which will bring up the dialog shown in figure 18.26.

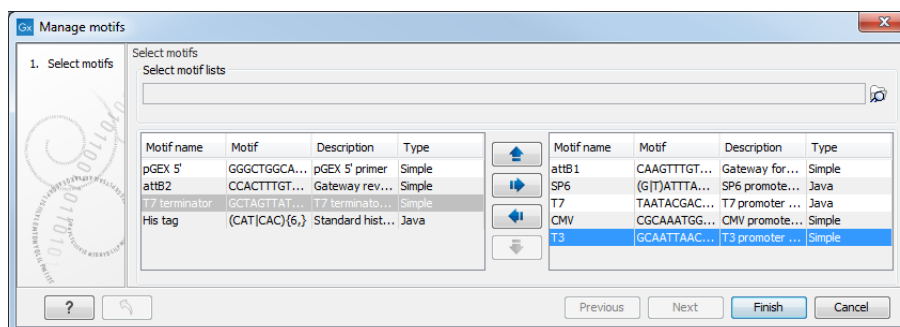


Figure 18.26: Managing the motifs to display

At the top, select a motif list by clicking the **Browse** button. When the motif list is selected, its motifs are listed in the panel in the left-hand side of the dialog. The right-hand side panel contains the motifs that will be listed in the **Side Panel** when you click **Finish**.

See section 18.9.2 for a non-interactive option for detecting motifs.

18.9.2 The Motif Search tool

The **Motif Search** tool supports searching for motifs in nucleotide or peptide sequences, including in alignments. Motifs to search for can be provided as simple text, as regular expressions or in lists, and a match accuracy for a successful match can be defined. Searches on the negative strand of nucleotide sequences is optional. Results can be added as annotation to the sequences, and a table of results can also be output.

To start **Motif Search** tool, go to:

Tools | Classical Sequence Analysis (🔍) | General Sequence Analysis (🔍) | Motif Search (🔍)

In the launch wizard, select the sequences, sequence lists, or alignments from the Navigation Area.

Search options are provided in the "Motif Search Parameters" wizard step (see figure 18.27).

The Motif Search options are:

- **Motif types.** Choose what kind of motif to be used:
 - Simple motif. Choosing this option means that you enter a simple motif, e.g. ATGATGNNATG.
 - Java regular expression. See section 18.9.3.
 - Prosite regular expression. For proteins, you can enter different protein patterns from the PROSITE database (protein patterns using regular expressions and describing specific amino acid sequences). The PROSITE database contains a great number of patterns and have been used to identify related proteins (see <https://prosite.expasy.org/cgi-bin/prosite/prosite-list.pl>).

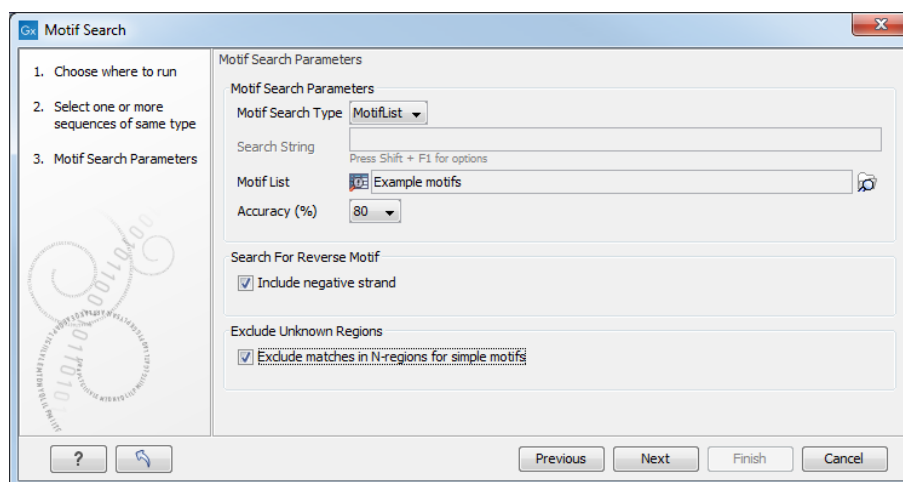


Figure 18.27: Specifying the options for the Motif Search tool.

- Use motif list. Clicking the small button (📁) will allow you to select a saved motif list (see section 18.10).
- **Motif.** If you choose to search with a simple motif, you should enter a literal string as your motif. Ambiguous amino acids and nucleotides are allowed. Example; ATGATGNNATG. If your motif type is Java regular expression, you should enter a regular expression according to the syntax rules described in section 18.9.3. Press **Shift + F1** key for options. For proteins, you can search with a Prosite regular expression and you should enter a protein pattern from the PROSITE database.
- **Accuracy.** If you search with a simple motif, you can adjust the accuracy of the motif to the match on the sequence. If you type in a simple motif and let the accuracy be 80%, the motif search algorithm runs through the input sequence and finds all subsequences of the same length as the simple motif such that the fraction of identity between the subsequence and the simple motif is at least 80%. A motif match is added to the sequence as an annotation with the exact fraction of identity between the subsequence and the simple motif. If you use a list of motifs, the accuracy applies only to the simple motifs in the list.
- **Search for reverse motif.** This enables searching on the negative strand on nucleotide sequences.
- **Exclude unknown regions.** Genome sequence often have large regions with unknown sequence. These regions are very often padded with N's. Ticking this checkbox will not display hits found in N-regions. Motif search handles ambiguous characters in the way that two residues are different if they do not have any residues in common. For example: For nucleotides, *N* matches any character and *R* matches A,G. For proteins, *X* matches any character and *Z* matches E,Q.

Click **Next** to adjust how to handle the results and then click **Finish**. There are multiple types of results that can be produced:

- **Create report.** This will create a report with summary information about motifs found.
- **Create table.** This will create an overview table of all the motifs found for all the input sequences.

- **Add annotations to sequences.** This will add an annotation to the sequence when a motif is found (an example is shown in figure 18.28). For details on viewing annotations see section 15.3.1.

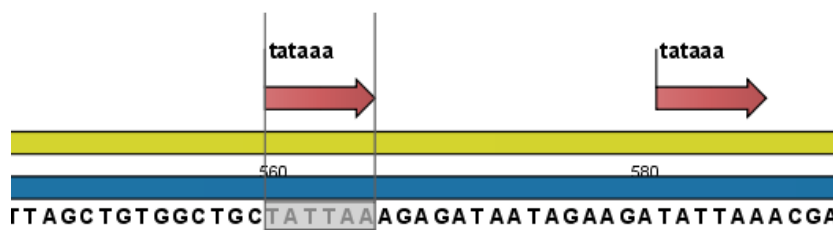


Figure 18.28: Sequence view displaying the pattern found. The search string was 'tataaa'.

18.9.3 Java regular expressions

A regular expressions is a string that describes or matches a set of strings, according to certain syntax rules. They are usually used to give a concise description of a set, without having to list all elements. The simplest form of a regular expression is a literal string. The syntax used for the regular expressions is the Java regular expression syntax (see <https://java.sun.com/docs/books/tutorial/essential/regex/index.html>). Below is listed some of the most important syntax rules which are also shown in the help pop-up when you press Shift + F1:

`[A-Z]` will match the characters A through Z (Range). You can also put single characters between the brackets: The expression `[AGT]` matches the characters A, G or T.

`[A-D[M-P]]` will match the characters A through D and M through P (Union). You can also put single characters between the brackets: The expression `[AG[M-P]]` matches the characters A, G and M through P.

`[A-M&&[H-P]]` will match the characters between A and M lying between H and P (Intersection). You can also put single characters between the brackets. The expression `[A-M&&[HGTDA]]` matches the characters A through M which is H, G, T, D or A.

`[^A-M]` will match any character except those between A and M (Excluding). You can also put single characters between the brackets: The expression `[^AG]` matches any character except A and G.

`[A-Z&&[^M-P]]` will match any character A through Z except those between M and P (Subtraction). You can also put single characters between the brackets: The expression `[A-P&&[^CG]]` matches any character between A and P except C and G.

The symbol `.` matches any character.

`X{n}` will match a repetition of an element indicated by following that element with a numerical value or a numerical range between the curly brackets. For example, `ACG{2}` matches the string `ACGG` and `(ACG){2}` matches `ACGACG`.

`X{n,m}` will match a certain number of repetitions of an element indicated by following that element with

two numerical values between the curly brackets. The first number is a lower limit on the number of repetitions and the second number is an upper limit on the number of repetitions. For example, `ACT{1,3}` matches `ACT`, `ACTT` and `ACTTT`.

`X{n,}` represents a repetition of an element at least n times. For example, `(AC){2,}` matches all strings `ACAC`, `ACACAC`, `ACACACAC`,...

The symbol `^` restricts the search to the beginning of your sequence. For example, if you search through a sequence with the regular expression `^AC`, the algorithm will find a match if `AC` occurs in the beginning of the sequence.

The symbol `$` restricts the search to the end of your sequence. For example, if you search through a sequence with the regular expression `GT$`, the algorithm will find a match if `GT` occurs in the end of the sequence.

Examples

The expression `[ACG][^AC]G{2}` matches all strings of length 4, where the first character is A,C or G and the second is any character except A,C and the third and fourth character is G. The expression `G.[^A]$` matches all strings of length 3 in the end of your sequence, where the first character is C, the second any character and the third any character except A.

18.10 Create motif list

CLC Genomics Workbench offers advanced and versatile options to create lists of sequence patterns or known motifs, represented either by a literal string or a regular expression.

A motif list can be created using:

Tools | Classical Sequence Analysis (📁) | General Sequence Analysis (📁) | Create Motif List (🛠️)

Click on the **Add** (+) button at the bottom of the view. This will open a dialog shown in figure 18.29.

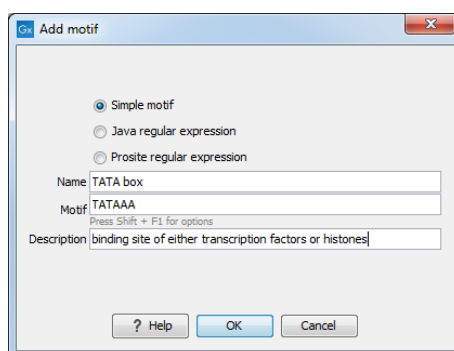




Figure 18.29: Entering a new motif in the list.


In this dialog, you can enter the following information:


- **Name.** The name of the motif. In the result of a motif search, this name will appear as the name of the annotation and in the result table.
- **Motif.** The actual motif. See section 18.9.2 for more information about the syntax of motifs.


- **Description.** You can enter a description of the motif. In the result of a motif search, the description will appear in the result table and will be added as a note to the annotation on the sequence (visible in the **Annotation table**  or by placing the mouse cursor on the annotation).
- **Type.** You can enter three different types of motifs: Simple motifs, java regular expressions or PROSITE regular expression. Read more in section [18.9.2](#).

The motif list can contain a mix of different types of motifs. This is practical because some motifs can be described with the simple syntax, whereas others need the more advanced regular expression syntax.

Instead of manually adding motifs, you can **Import From Fasta File** . This will show a dialog where you can select a fasta file on your computer and use this to create motifs. This will automatically take the name, description and sequence information from the fasta file, and put it into the motif list. The motif type will be "simple". Note that reformatting Prosite file into FASTA format for import will fail, as only simple motifs can be imported this way and regular expressions are not supported.

Besides adding new motifs, you can also edit and delete existing motifs in the list. To edit a motif, either double-click the motif in the list, or select and click the **Edit** () button at the bottom of the view.

To delete a motif, select it and press the Delete key on the keyboard. Alternatively, click **Delete** () in the **Tool bar**.

Save the motif list in the **Navigation Area**, and you will be able to use for Motif Search () (see section [18.9](#)).

Chapter 19

Nucleotide analyses

Contents

19.1 Convert DNA to RNA	492
19.2 Convert RNA to DNA	493
19.3 Reverse complements of sequences	493
19.4 Translation of DNA or RNA to protein	494
19.5 Find open reading frames	496

CLC Genomics Workbench offers different kinds of sequence analyses, which only apply to DNA and RNA.

19.1 Convert DNA to RNA

Use the **Convert DNA to RNA** tool to convert a DNA sequence into an RNA sequence, substituting the T residues (Thymine) for U residues (Uracil). It is available at:

Tools | Classical Sequence Analysis (📁) | Nucleotide Analysis (📁) | Convert DNA to RNA (🔗)

This opens the dialog displayed in figure 19.1:

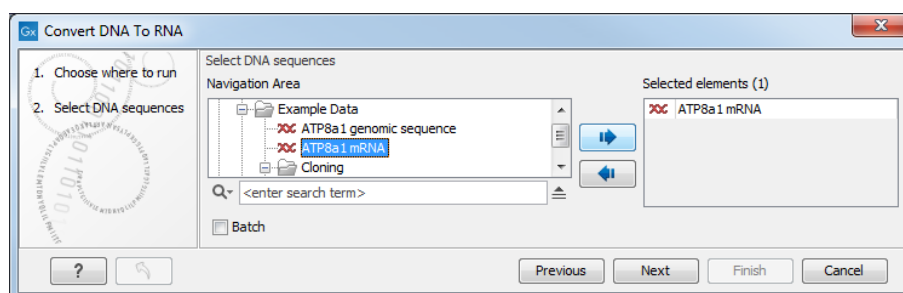


Figure 19.1: Translating DNA to RNA.

Use the arrows to add or remove sequences or sequence lists from the selected elements list.

You can select multiple DNA sequences and sequence lists for conversion. If a sequence list contains RNA sequences, those sequences will not be converted.

Click **Finish** to start the tool.

19.2 Convert RNA to DNA

Use the **Convert RNA to DNA** tool to convert an RNA sequence into DNA, substituting the U residues (Uracil) for T residues (Thymine). It is available at:

Tools | Classical Sequence Analysis (📁) | **Nucleotide Analysis** (📁) | **Convert RNA to DNA** (🔗)

This opens the dialog displayed in figure 19.2:

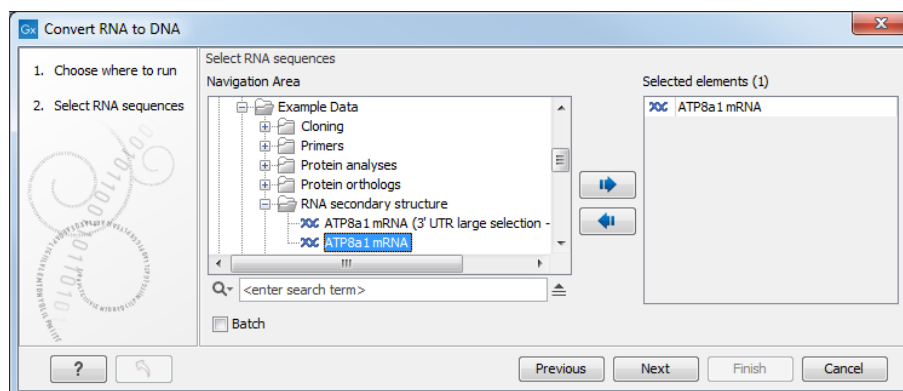


Figure 19.2: Translating RNA to DNA.

Use the arrows to add or remove sequences or sequence lists from the selected elements list.

You can select multiple RNA sequences and sequence lists for conversion. If a selected sequence list contains DNA sequences, those sequences will not be converted.

Click **Finish** to start the tool.

This will open a new view in the **View Area** displaying the new DNA sequence. The new sequence is not saved automatically. To save the sequence, drag it into the **Navigation Area** or press Ctrl + S (⌘ + S on Mac) to activate a save dialog.

19.3 Reverse complements of sequences

CLC Genomics Workbench is able to create the reverse complement of a nucleotide sequence. By doing that, a new sequence is created which also has all the annotations reversed since they now occupy the opposite strand of their previous location.

To quickly obtain the reverse complement of a sequence or part of a sequence, you may select a region on the negative strand and open it in a new view:

right-click a selection on the negative strand | Open selection in New View (📄)

By doing that, the sequence will be reversed. This is only possible when the double stranded view option is enabled. It is possible to copy the selection and paste it in a word processing program or an e-mail. To obtain a reverse complement of an entire sequence:

Tools | Classical Sequence Analysis (📁) | **Nucleotide Analysis** (📁) | **Reverse Complement Sequence** (🔗)

This opens the dialog displayed in figure 19.3:

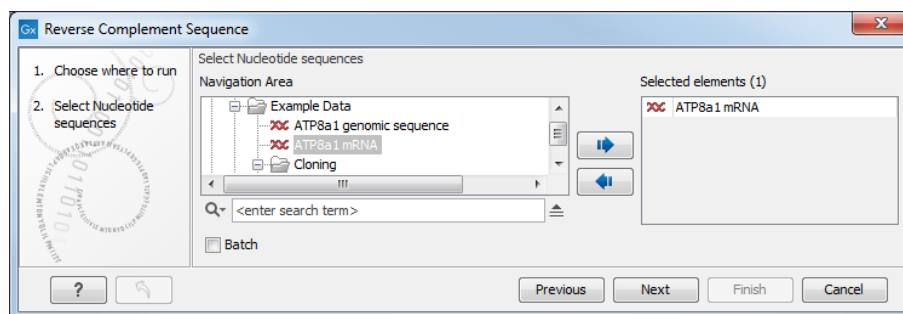


Figure 19.3: Creating a reverse complement sequence.

Use the arrows to add or remove sequences or sequence lists from the selected elements list.

Click **Finish** to start the tool.

This will open a new view in the **View Area** displaying the reverse complement of the selected sequence. The new sequence is not saved automatically. To save the sequence, drag it into the **Navigation Area** or press Ctrl + S (⌘ + S on Mac) to activate a save dialog.

19.4 Translation of DNA or RNA to protein

To translate a nucleotide sequence into a protein sequence use the **Translate to Protein** tool, available at:

Tools | Classical Sequence Analysis (🔍) | **Nucleotide Analysis** (📄) | **Translate to Protein** (🔍)

This opens the dialog displayed in figure 19.4:

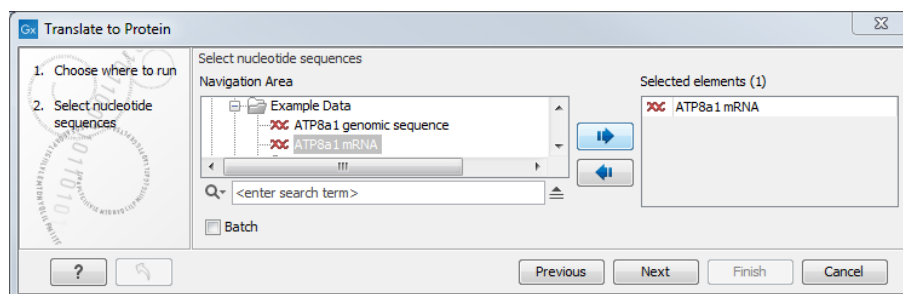


Figure 19.4: Choosing sequences for translation.

Use the arrows to add or remove sequences or sequence lists from the selected elements list.

Clicking **Next** generates the dialog seen in figure 19.5:

Here you have the following options:

Reading frames If you wish to translate the whole sequence, you must specify the reading frame for the translation. If you select e.g. two reading frames, two protein sequences are generated.

Translate CDS You can choose to translate regions marked by and CDS or ORF annotation. This will generate a protein sequence for each CDS or ORF annotation on the sequence. The

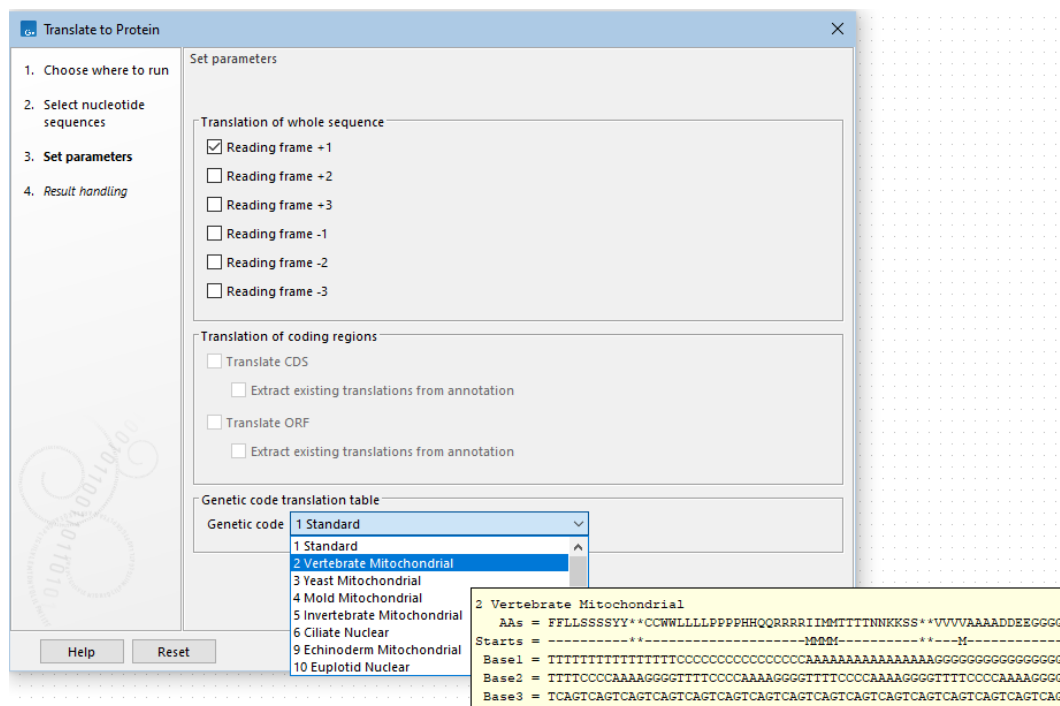


Figure 19.5: Configure the translation options. Hover the mouse cursor over a genetic code option to reveal a tooltip containing the relevant translation table.

"Extract existing translations from annotation" allows to list the amino acid CDS sequence shown in the tool tip annotation (e.g. interstate from NCBI download) and does therefore not represent a translation of the actual nt sequence.

Genetic code Specify the genetic code to use. Hover the mouse cursor over an item in this list to reveal a tooltip containing the relevant translation table (figure 19.5). The translation tables are sourced from the NCBI (<https://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi>).

Stop codons result in an asterisk being inserted in the protein sequence at the corresponding position.

Click **Finish** to start the tool. The newly created protein is shown, but is not saved automatically.

To save a protein sequence, drag it into the **Navigation Area** or press Ctrl + S (⌘ + S on Mac) to activate a save dialog.

The name for a coding region translation consists of the name of the input sequence followed by the annotation type and finally the annotation name.

Translate part of a nucleotide sequence If you want to make separate translations of *all* the coding regions of a nucleotide sequence, you can check the option: "Translate CDS/ORF..." in the translation dialog (see figure 19.5).

If you want to translate a *specific* coding region, which is annotated on the sequence, use the following procedure:

Open the nucleotide sequence | right-click the ORF or CDS annotation | Translate CDS/ORF... (👉)

A dialog opens to offer you the following choices (figure 19.6):

- **Select a genetic code translation table** Translates the ORF/CDS to protein using the selected translation table. Hover the mouse cursor over an item in this list to reveal a tooltip containing the relevant translation table (figure 19.5). The translation tables are sourced from the NCBI (<https://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi>).
- **Extract existing translation from annotation** Translates the ORF/CDS to protein using existing translation information available in the annotation.

Choose the option needed and click **Translate**.

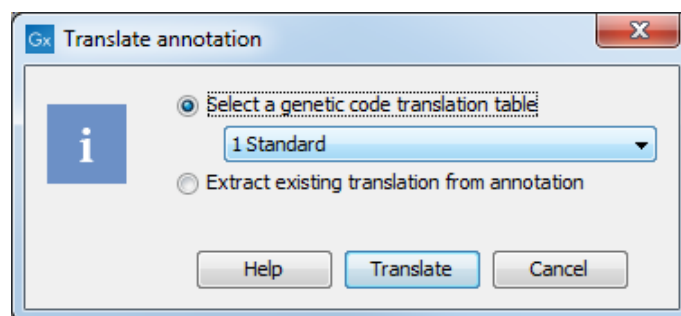


Figure 19.6: Choosing how to translate CDS or ORF annotations.

The CDS and ORF annotations are colored yellow as default.

19.5 Find open reading frames

Find Open Reading Frames identifies open reading frames (ORFs) in sequences, and can be used as a rudimentary gene finder.

During translation of a transcript, protein is generated from the first start codon to the stop codon, internal start codons are translated to their respective amino acids. **Find Open Reading Frames** correspondingly always reports ORFs using the first possible start codon and ignores internal start codons.

Identified ORFs are shown as annotations on the sequence. Different genetic codes are available, but it is also possible to manually specify start codons.

In one analysis, **Find Open Reading Frames** can process a maximum of 100,000 sequences or 50 million base pairs. Sequences may be provided to the tool as individual sequences or as sequence lists.

To run **Find Open Reading Frames**, go to:

Tools | Classical Sequence Analysis (📁) | **Nucleotide Analysis** (📁) | **Find Open Reading Frames** (🔍)

This opens the dialog displayed in figure 19.7

Use the arrows to add or remove sequences or sequence lists from the selected elements list.

Next, specify which parameters should be used (figure 19.8)

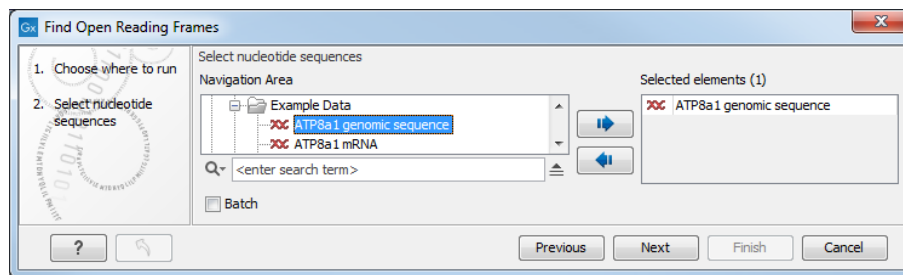


Figure 19.7: Select a sequence or a sequence list as input.

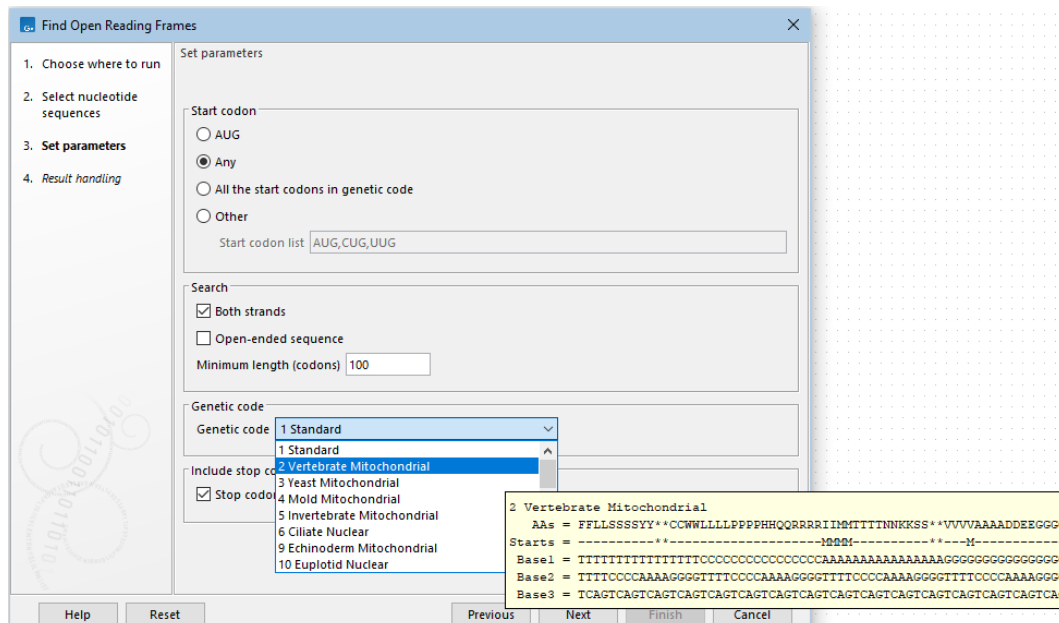


Figure 19.8: Configure the options for finding open reading frames. Hover the mouse cursor over a genetic code option to reveal a tooltip containing the relevant translation table.

- **Start codon**

- **AUG** Most commonly used start codon. When selected, only AUG (or ATG) codons are used as start codons.
- **Any** Any codon can be used as the start codon. For identification of the open reading frames, the first possible codon in the same reading frame as the stop codon is used as the start codon.
- **All the start codons in genetic code** Select to use the start codons that are specific to the genetic code specified under **Genetic code**.
- **Other** Identifies open reading frames that start with one of the codons provided in the start codon list.

- **Both strands** Find reading frames on both strands.

- **Open-ended sequence** Allow ORFs to extend up to the sequence start or end not considering the sequence context. This can be relevant when only a fragment of a sequence is analyzed, and there may be up- or downstream start and stop codons that are not included in the sequence. When predicting the open reading frames, stop codons are always used, but a given start codon is only used if it is the first one after the last stop codon. Start codons

Chapter 20

Protein analyses

Contents

20.1 Protein charge	499
20.2 Antigenicity	501
20.3 Hydrophobicity	502
20.3.1 Hydrophobicity graphs along sequence	503
20.3.2 Bioinformatics explained: Protein hydrophobicity	505
20.4 Download Pfam Database	507
20.5 Pfam domain search	507
20.6 Find and Model Structure	510
20.6.1 Create structure model	511
20.6.2 Model structure	512
20.7 Secondary structure prediction	518
20.8 Protein report	519
20.9 Reverse translation from protein into DNA	522
20.9.1 Bioinformatics explained: Reverse translation	523
20.10 Proteolytic cleavage detection	525
20.10.1 Bioinformatics explained: Proteolytic cleavage	527

CLC Genomics Workbench offers a number of analyses of proteins as described in this chapter.

Note that the SignalP and TMHMM plugin allows you to predict signal peptides. For more information, please read the plugin manual at https://resources.qiagenbioinformatics.com/manuals/signalpandtmhmm/current/User_Manual.pdf.

The TMHMM plugin allows you to predict transmembrane helix. For more information, please read the plugin manual at http://resources.qiagenbioinformatics.com/manuals/tmhmm/current/Tmhmm_User_Manual.pdf.

20.1 Protein charge

In *CLC Genomics Workbench* you can create a graph in the electric charge of a protein as a function of pH. This is particularly useful for finding the net charge of the protein at a given pH.

This knowledge can be used e.g. in relation to isoelectric focusing on the first dimension of 2D-gel electrophoresis. The isoelectric point (pI) is found where the net charge of the protein is zero. The calculation of the protein charge does not include knowledge about any potential post-translational modifications the protein may have.

The pKa values reported in the literature may differ slightly, thus resulting in different looking graphs of the protein charge plot compared to other programs.

In order to calculate the protein charge:

Tools | Classical Sequence Analysis (📁) | Protein Analysis (📁) | Create Protein Charge Plot (📄)

This opens the dialog displayed in figure 20.1:

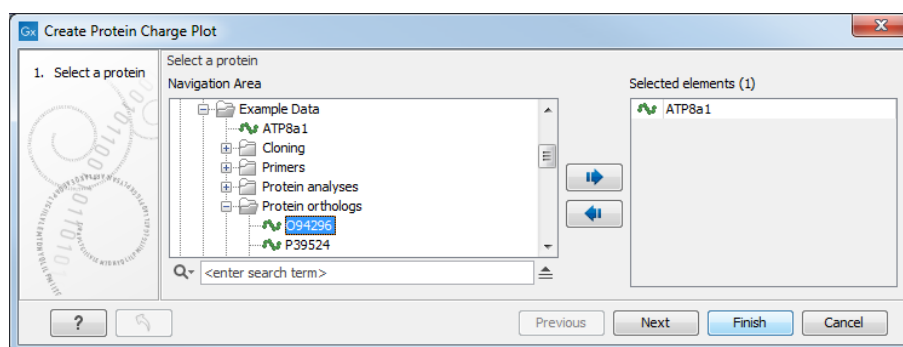


Figure 20.1: Choosing protein sequences to calculate protein charge.

If a sequence was selected before running the tool, the sequence will be listed in the **Selected Elements** pane of the dialog. Use the arrows to add or remove sequences or sequence lists from the selected elements.

You can perform the analysis on several protein sequences at a time. This will result in one output graph showing protein charge graphs for the individual proteins.

Click **Finish** to start the tool.

Figure 20.2 shows the electrical charges for three proteins. In the **Side Panel** to the right, you can modify the layout of the graph.

See section B in the appendix for information about the graph view.

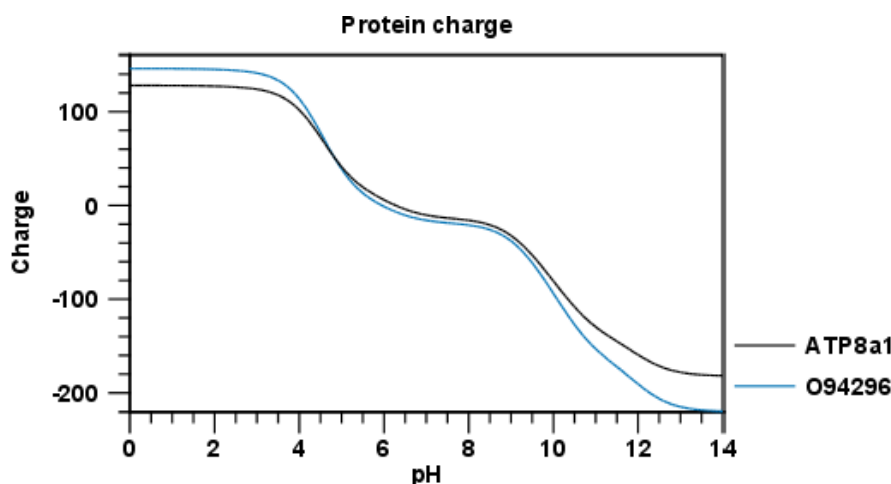


Figure 20.2: View of the protein charge.

20.2 Antigenicity

CLC Genomics Workbench can help to identify antigenic regions in protein sequences in different ways, using different algorithms. The algorithms provided in the Workbench, merely plot an index of antigenicity over the sequence.

Two different methods are available:

- [Welling et al., 1985] Welling et al. used information on the relative occurrence of amino acids in antigenic regions to make a scale which is useful for prediction of antigenic regions. This method is better than the Hopp-Woods scale of hydrophobicity which is also used to identify antigenic regions.
- A semi-empirical method for prediction of antigenic regions has been developed [Kolaskar and Tongaonkar, 1990]. This method also includes information of surface accessibility and flexibility and at the time of publication the method was able to predict antigenic determinants with an accuracy of 75%.

Note! Similar results from the two methods can not always be expected as the two methods are based on different training sets.

Displaying the antigenicity for a protein sequence in a plot is done in the following way:

Tools | Classical Sequence Analysis (📁) | Protein Analysis (📁) | Create Antigenicity Plot (📊)

This opens a dialog. The first step allows you to add or remove sequences. If you had already selected sequences in the Navigation Area before running the tool, these will be listed in the **Selected Elements** pane. Clicking **Next** takes you through to **Step 2**, which is displayed in figure 20.3.

The **Window size** is the width of the window where, the antigenicity is calculated. The wider the window, the less volatile the graph. You can choose from a number of antigenicity scales. Click **Finish** to start the tool. The result can be seen in figure 20.4.

See section B in the appendix for information about the graph view.

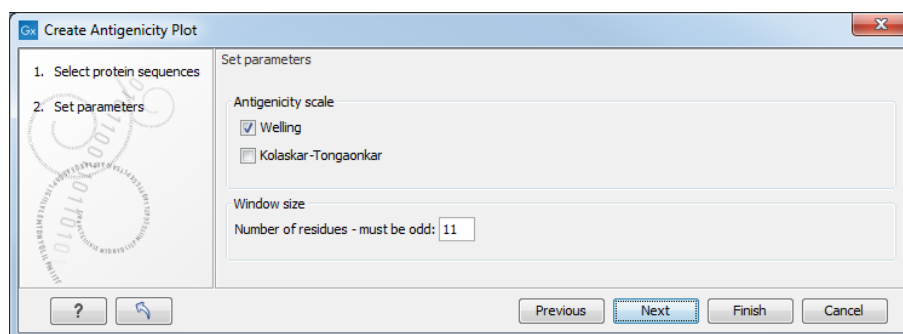


Figure 20.3: Step two in the *Antigenicity Plot* allows you to choose different antigenicity scales and the window size.

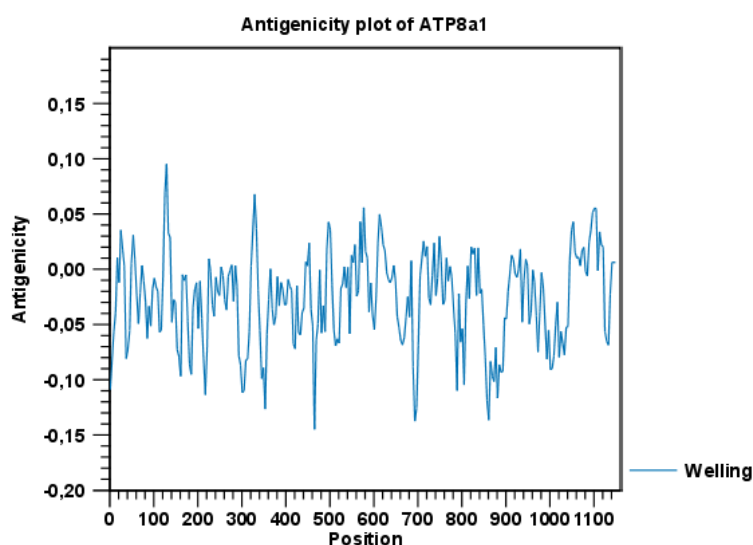


Figure 20.4: The result of the antigenicity plot calculation and the associated Side Panel.

The level of antigenicity is calculated on the basis of the different scales. The different scales add different values to each type of amino acid. The antigenicity score is then calculated as the sum of the values in a 'window', which is a particular range of the sequence. The window length can be set from 5 to 25 residues. The wider the window, the less fluctuations in the antigenicity scores.

Antigenicity graphs along the sequence can be displayed using the **Side Panel**. The functionality is similar to hydrophobicity (see section 20.3.1).

20.3 Hydrophobicity

CLC Genomics Workbench can calculate the hydrophobicity of protein sequences in different ways, using different algorithms (see section 20.3.2). Furthermore, hydrophobicity of sequences can be displayed as hydrophobicity plots and as graphs along sequences. In addition, CLC Genomics Workbench can calculate hydrophobicity for several sequences at the same time, and for alignments.

Displaying the hydrophobicity for a protein sequence in a plot is done in the following way:

Tools | Classical Sequence Analysis (📁) | **Protein Analysis** (📁) | **Create Hydrophobicity Plot** (📊)

This opens a dialog. The first step allows you to add or remove sequences. If you had already selected a sequence in the Navigation Area, this will be shown in the **Selected Elements**. Clicking **Next** takes you through to **Step 2**, which is displayed in figure 20.5.

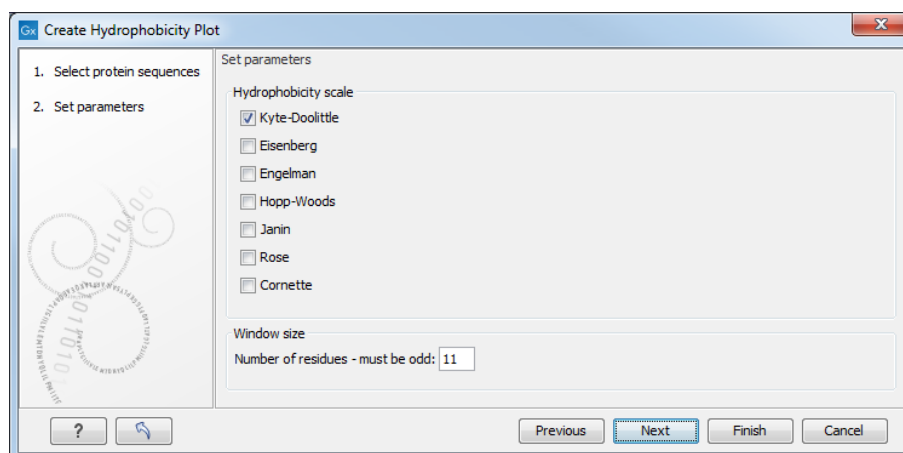


Figure 20.5: Step two in the Hydrophobicity Plot allows you to choose hydrophobicity scale and the window size.

The **Window size** is the width of the window where the hydrophobicity is calculated. The wider the window, the less volatile the graph. You can choose from a number of hydrophobicity scales which are further explained in section 20.3.2. Click **Finish** to start the tool. The result can be seen in figure 20.6.

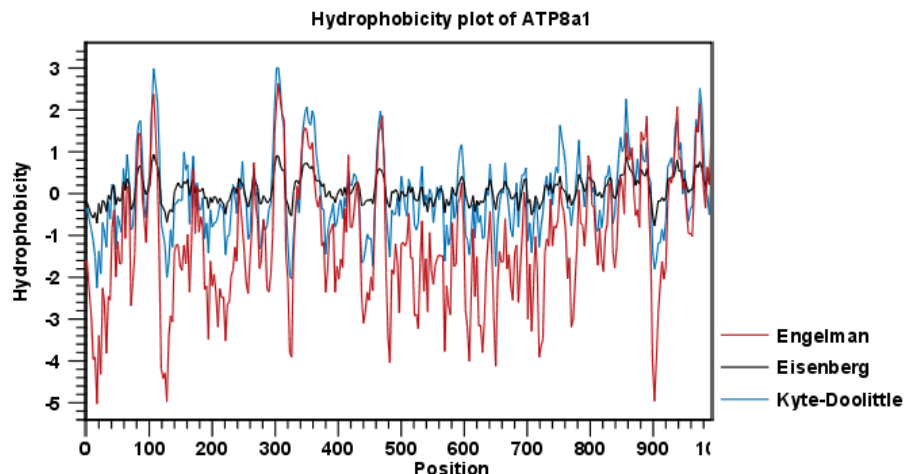


Figure 20.6: The result of the hydrophobicity plot calculation and the associated Side Panel.

See section B in the appendix for information about the graph view.

20.3.1 Hydrophobicity graphs along sequence

Hydrophobicity graphs along sequence can be displayed easily by activating the calculations from the **Side Panel** for a sequence. Simply right-click or double click on a protein sequence in the Navigation Area, and choose

Show | Sequence | open Protein info in Side Panel

These actions result in the view displayed in figure 20.7.



Figure 20.7: The different available scales in Protein info.

The level of hydrophobicity is calculated on the basis of the different scales. The different scales add different values to each type of amino acid. The hydrophobicity score is then calculated as the sum of the values in a 'window', which is a particular range of the sequence. The window length can be set from 5 to 25 residues. The wider the window, the less fluctuations in the hydrophobicity scores. (For more about the theory behind hydrophobicity, see 20.3.2).

In the following we will focus on the different ways that the Workbench offers to display the hydrophobicity scores. We use Kyte-Doolittle to explain the display of the scores, but the different options are the same for all the scales. Initially there are three options for displaying the hydrophobicity scores. You can choose one, two or all three options by selecting the boxes (figure 20.8).

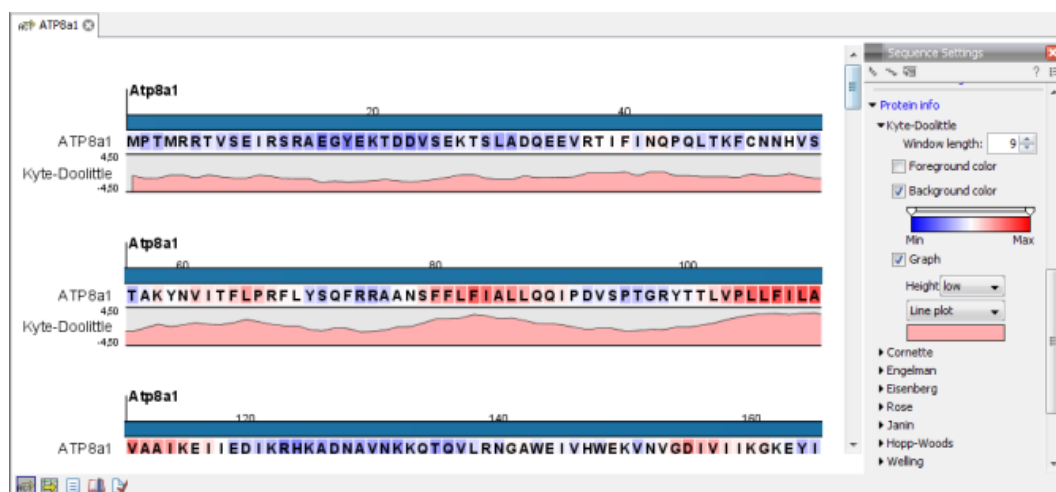


Figure 20.8: The different ways of displaying the hydrophobicity scores, using the Kyte-Doolittle scale.

Coloring the letters and their background. When choosing coloring of letters or coloring of their background, the color red is used to indicate high scores of hydrophobicity. A 'color-slider' allows you to amplify the scores, thereby emphasizing areas with high (or low, blue) levels of hydrophobicity. The color settings mentioned are default settings. By clicking the color bar just below the color slider you get the option of changing color settings.

Graphs along sequences. When selecting graphs, you choose to display the hydrophobicity scores underneath the sequence. This can be done either by a line-plot or bar-plot, or by coloring. The latter option offers you the same possibilities of amplifying the scores as applies for coloring of letters. The different ways to display the scores when choosing 'graphs' are displayed in figure 20.8. Notice that you can choose the height of the graphs underneath the sequence.

20.3.2 Bioinformatics explained: Protein hydrophobicity

Calculation of hydrophobicity is important to the identification of various protein features. This can be membrane spanning regions, antigenic sites, exposed loops or buried residues. Usually, these calculations are shown as a plot along the protein sequence, making it easy to identify the location of potential protein features.

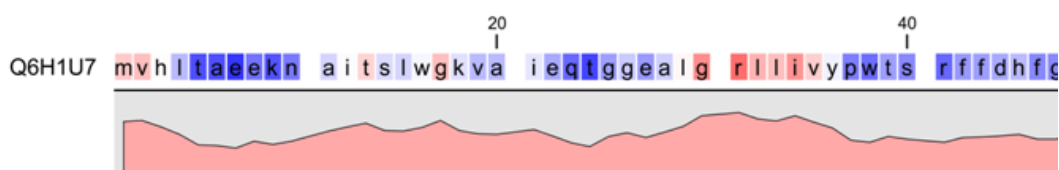


Figure 20.9: Plot of hydrophobicity along the amino acid sequence. Hydrophobic regions on the sequence have higher numbers according to the graph below the sequence, furthermore hydrophobic regions are colored on the sequence. Red indicates regions with high hydrophobicity and blue indicates regions with low hydrophobicity.

The hydrophobicity is calculated by sliding a fixed size window (of an odd number) over the protein sequence. At the central position of the window, the average hydrophobicity of the entire window is plotted (see figure 20.9).

Hydrophobicity scales Several hydrophobicity scales have been published for various uses. Many of the commonly used hydrophobicity scales are described below.

- **Kyte-Doolittle scale.** The Kyte-Doolittle scale is widely used for detecting hydrophobic regions in proteins. Regions with a positive value are hydrophobic. This scale can be used for identifying both surface-exposed regions as well as transmembrane regions, depending on the window size used. Short window sizes of 5-7 generally work well for predicting putative surface-exposed regions. Large window sizes of 19-21 are well suited for finding transmembrane domains if the values calculated are above 1.6 [Kyte and Doolittle, 1982]. These values should be used as a rule of thumb and deviations from the rule may occur.
- **Engelman scale.** The Engelman hydrophobicity scale, also known as the GES-scale, is another scale which can be used for prediction of protein hydrophobicity [Engelman et al., 1986]. As the Kyte-Doolittle scale, this scale is useful for predicting transmembrane regions in proteins.
- **Eisenberg scale.** The Eisenberg scale is a normalized consensus hydrophobicity scale which shares many features with the other hydrophobicity scales [Eisenberg et al., 1984].
- **Hopp-Woods scale.** Hopp and Woods developed their hydrophobicity scale for identification of potentially antigenic sites in proteins. This scale is basically a hydrophilic index where

apolar residues have been assigned negative values. Antigenic sites are likely to be predicted when using a window size of 7 [Hopp and Woods, 1983].

- **Cornette scale.** Cornette *et al.* computed an optimal hydrophobicity scale based on 28 published scales [Cornette *et al.*, 1987]. This optimized scale is also suitable for prediction of alpha-helices in proteins.
- **Rose scale.** The hydrophobicity scale by Rose *et al.* is correlated to the average area of buried amino acids in globular proteins [Rose *et al.*, 1985]. This results in a scale which is not showing the helices of a protein, but rather the surface accessibility.
- **Janin scale.** This scale also provides information about the accessible and buried amino acid residues of globular proteins [Janin, 1979].
- **Welling scale.** Welling *et al.* used information on the relative occurrence of amino acids in antigenic regions to make a scale which is useful for prediction of antigenic regions. This method is better than the Hopp-Woods scale of hydrophobicity which is also used to identify antigenic regions.
- **Kolaskar-Tongaonkar.** A semi-empirical method for prediction of antigenic regions has been developed [Kolaskar and Tongaonkar, 1990]. This method also includes information of surface accessibility and flexibility and at the time of publication the method was able to predict antigenic determinants with an accuracy of 75%.
- **Surface Probability.** Display of surface probability based on the algorithm by [Emini *et al.*, 1985]. This algorithm has been used to identify antigenic determinants on the surface of proteins.
- **Chain Flexibility.** Display of backbone chain flexibility based on the algorithm by [Karplus and Schulz, 1985]. It is known that chain flexibility is an indication of a putative antigenic determinant.

Many more scales have been published throughout the last three decades. Even though more advanced methods have been developed for prediction of membrane spanning regions, the simple and very fast calculations are still highly used.

aa	aa	Kyte-Doolittle	Hopp-Woods	Cornette	Eisenberg	Rose	Janin	Engelman (GES)
A	Alanine	1.80	-0.50	0.20	0.62	0.74	0.30	1.60
C	Cysteine	2.50	-1.00	4.10	0.29	0.91	0.90	2.00
D	Aspartic acid	-3.50	3.00	-3.10	-0.90	0.62	-0.60	-9.20
E	Glutamic acid	-3.50	3.00	-1.80	-0.74	0.62	-0.70	-8.20
F	Phenylalanine	2.80	-2.50	4.40	1.19	0.88	0.50	3.70
G	Glycine	-0.40	0.00	0.00	0.48	0.72	0.30	1.00
H	Histidine	-3.20	-0.50	0.50	-0.40	0.78	-0.10	-3.00
I	Isoleucine	4.50	-1.80	4.80	1.38	0.88	0.70	3.10
K	Lysine	-3.90	3.00	-3.10	-1.50	0.52	-1.80	-8.80
L	Leucine	3.80	-1.80	5.70	1.06	0.85	0.50	2.80
M	Methionine	1.90	-1.30	4.20	0.64	0.85	0.40	3.40
N	Asparagine	-3.50	0.20	-0.50	-0.78	0.63	-0.50	-4.80
P	Proline	-1.60	0.00	-2.20	0.12	0.64	-0.30	-0.20
Q	Glutamine	-3.50	0.20	-2.80	-0.85	0.62	-0.70	-4.10
R	Arginine	-4.50	3.00	1.40	-2.53	0.64	-1.40	-12.3
S	Serine	-0.80	0.30	-0.50	-0.18	0.66	-0.10	0.60
T	Threonine	-0.70	-0.40	-1.90	-0.05	0.70	-0.20	1.20
V	Valine	4.20	-1.50	4.70	1.08	0.86	0.60	2.60
W	Tryptophan	-0.90	-3.40	1.00	0.81	0.85	0.30	1.90
Y	Tyrosine	-1.30	-2.30	3.20	0.26	0.76	-0.40	-0.70

Table 20.1: *Hydrophobicity scales. This table shows seven different hydrophobicity scales which are generally used for prediction of e.g. transmembrane regions and antigenicity.*

Other useful resources

AAindex: Amino acid index database

<http://www.genome.ad.jp/dbget/aaindex.html>

20.4 Download Pfam Database

To run **Pfam Domain Search** you must first download the Pfam database. The Pfam database can be downloaded using:

Tools | Classical Sequence Analysis (📁) | Protein Analysis (📁) | Download Pfam Database (📁)

This element is designed for use with Pfam Domain Search. If you open it directly from the Navigation Area, only the element history is accessible.

20.5 Pfam domain search

Pfam Domain Search searches for domains in protein sequences using the Pfam database [Bate-man et al., 2004], a large collection of multiple sequence alignments and hidden Markov models (HMMs) covering many common protein domains. It can add **Region** annotations on the input sequences where domains were found (figure 20.11) and it can output a table listing the domains found.

Why search against Pfam? Many proteins have a unique combination of domains, which can be responsible for the catalytic activities of enzymes. Annotating sequences based on pairwise alignment methods by simply transferring annotation from a known protein to the unknown partner does not take domain organization into account [Galperin and Koonin, 1998]. For example, a

protein may be annotated incorrectly as an enzyme if the pairwise alignment only finds a regulatory domain.

After the Pfam database has been downloaded (see section 20.4), start **Pfam Domain Search** by going to:

Tools | Classical Sequence Analysis (📁) | Protein Analysis (📁) | Pfam Domain Search (➡)

By selecting several input sequences, you can perform the analysis on all these at once. Options can be configured (figure 20.10).

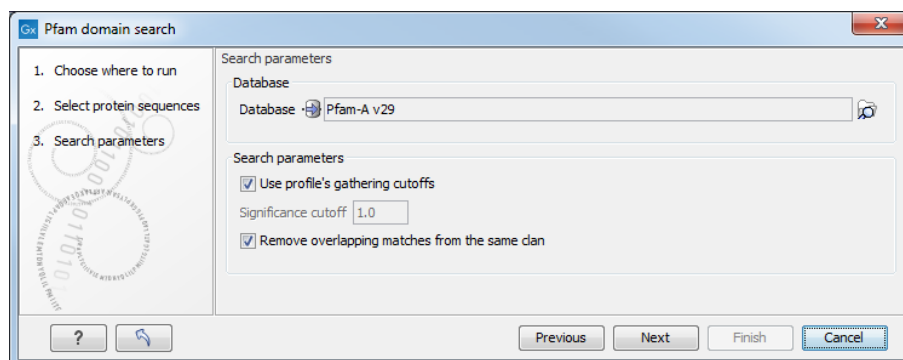


Figure 20.10: Setting parameters for Pfam Domain Search.

Pfam Domain Search options

- **Database** Choose the database to use when searching for Pfam domains.
- **Significance cutoff:**
 - **Use profile's gathering cutoffs** Use cutoffs specifically assigned to each family by the curator instead of manually assigning the **Significance cutoff**.
 - **Significance cutoff** The E-value (expectation value) describes the number of hits one would expect to see by chance when searching a database of a particular size. Essentially, a hit with a low E-value is more significant than a hit with a high E-value. By lowering the significance threshold the domain search will become more specific and less sensitive, i.e. fewer hits will be reported but the reported hits will be more significant on average.
- **Remove overlapping matches from the same clan** Perform post-processing of the results where overlaps between hits are resolved by keeping the hit with the smallest E-value.

If annotations were added but are not initially visible on your sequences, check under the "Annotation types" tab of the side panel settings to ensure the **Region** annotation type has been checked.

Detailed information for each domain annotation is available in the annotation tool tip as well as in the Annotation Table view of the sequence list.

The domain search is performed using the `hmmsearch` tool from the HMMER3 package version 3.4 <http://hmmerr.org/>. Detailed information about the scores in the Region annotations added can be found in the HMMER User Guide <http://eddylib.org/software/hmmerr/Userguide.pdf>.

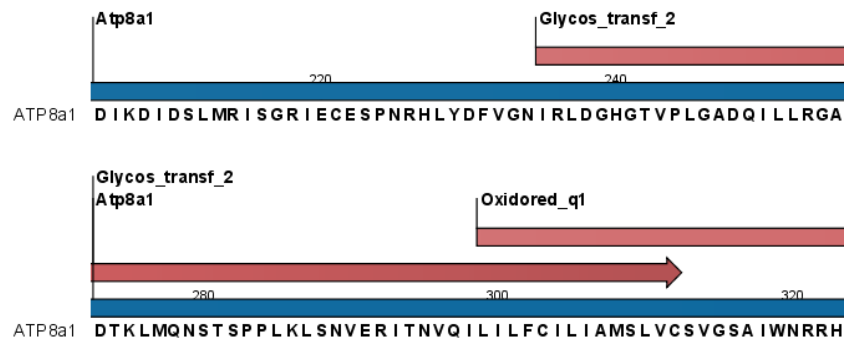


Figure 20.11: Annotations (in red) that were added by the Pfam search tool.

Individual domain annotations can be removed manually, if desired. See section [15.3.5](#).

20.6 Find and Model Structure

This tool is used to find suitable protein structures for representing a given protein sequence. From the resulting table, a structure model (homology model) of the sequence can be created by one click, using one of the found protein structures as template.

To run the *Find and Model Structure* tool, go to:

Tools | Sequence Analysis  | **Find and Model Structure** 

Note: Before running the tool, a protein structure sequence database must be downloaded and installed using the 'Download Find Structure Database' tool (see section 32.5.4).

In the tool wizard step 1, select the amino acid sequence to use as query from the Navigation Area.

In step 2, specify if the output table should be opened or saved.

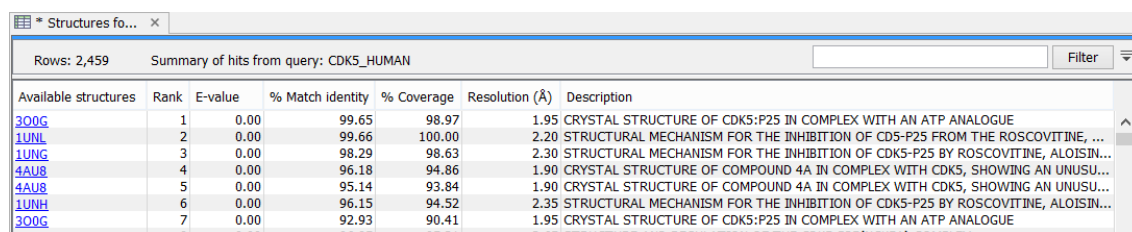
The Find and Model Structure tool carries out the following steps, to find and rank available structures representing the query sequence:

Input: Query protein sequence

1. BLAST against protein structure sequence database
2. Filter away low quality hits
3. Rank the available structures

Output: Table listing available structures

In the output table (figure 20.12), the column named "Available Structures" contains links that will invoke a menu with the options to either create a structure model of the query sequence or just download the structure. This is further described in section 20.6.1. The remaining columns contain additional information originating from the PDB file or from the BLAST search.



Available structures	Rank	E-value	% Match identity	% Coverage	Resolution (Å)	Description
3O0G	1	0.00	99.65	98.97	1.95	CRYSTAL STRUCTURE OF CDK5:P25 IN COMPLEX WITH AN ATP ANALOGUE
1UNL	2	0.00	99.66	100.00	2.20	STRUCTURAL MECHANISM FOR THE INHIBITION OF CD5-P25 FROM THE ROSCOVITINE, ...
1UNG	3	0.00	98.29	98.63	2.30	STRUCTURAL MECHANISM FOR THE INHIBITION OF CDK5-P25 BY ROSCOVITINE, ALOISIN...
4A08	4	0.00	96.18	94.86	1.90	CRYSTAL STRUCTURE OF COMPOUND 4A IN COMPLEX WITH CDK5, SHOWING AN UNUSU...
4A08	5	0.00	95.14	93.84	1.90	CRYSTAL STRUCTURE OF COMPOUND 4A IN COMPLEX WITH CDK5, SHOWING AN UNUSU...
1UNH	6	0.00	96.15	94.52	2.35	STRUCTURAL MECHANISM FOR THE INHIBITION OF CDK5-P25 BY ROSCOVITINE, ALOISIN...
3O0G	7	0.00	92.93	90.41	1.95	CRYSTAL STRUCTURE OF CDK5:P25 IN COMPLEX WITH AN ATP ANALOGUE

Figure 20.12: Table output from *Find and Model Structure*.

The three steps carried out by the *Find and Model Structure* tool are described in short below.

BLAST against protein structure sequence database A local BLAST search is carried out for the query sequence against the protein structure sequence database (see section 32.5.4).

BLAST hits with E-value > 0.0001 are rejected and a maximum of 2500 BLAST hits are retrieved. Read more about BLAST in section 16.5.

Filter away low quality hits From the list of BLAST hits, entries are rejected based on the following rules:

- PDB structures with a resolution lower than 4 Å are removed since they cannot be expected to represent a trustworthy atomistic model.
- BLAST hits with an identity to the query sequence lower than 20 % are removed since they most likely would result in inaccurate models.

Rank the available structures For the resulting list of available structures, each structure is scored based on its homology to the query sequence, and the quality of the structure itself. The *Template quality score* is used to rank the structures in the table, and the rank of each structure is shown in the "Rank" column (see figure 20.12). Read more about the *Template quality score* in section 20.6.2.

20.6.1 Create structure model

Clicking on a link in the "Available structures" column will show a menu with three options:

- Download and Open
- Download and Create Model
- Help

The "Download and Open" option will do the following:

1. **Download and import** the PDB file containing the structure.
2. **Create an alignment** between the query and structure sequences.
3. **Open a 3D view** (Molecule Project) with the molecules from the PDB file and open the created sequence alignment. The sequence originating from the structure will be linked to the structure in the 3D view, so that selections on the sequence will show up on the structure (see section 17.4).

The "Download and Create Model" option will do the following:

1. **Download and import** the PDB file containing the structure.
2. **Generate a biomolecule** involving the protein chain to be modeled. Biomolecule information available in the template PDB file is used (see section 17.6). If several biomolecules involving the chain are available, the first one is applied.
3. **Create an alignment** between the query and structure sequences.
4. **Create a model structure** by mapping the query sequence onto the structure based on the sequence alignment (see section 20.6.2). If multiple copies of the template protein chain have been made to generate a biomolecule, all copies are modeled at the same time.

5. **Open a 3D view** (a Molecule Project) with the structure model shown in both backbone and wireframe representation. The model is colored by temperature (see figure 20.13), to indicate local model uncertainty (see section 20.6.2). Other molecules from the template PDB file are shown in orange or yellow coloring. The created sequence alignment is also opened and linked with the 3D views so that selections on the model sequence will show up on the model structure (see section 17.4).

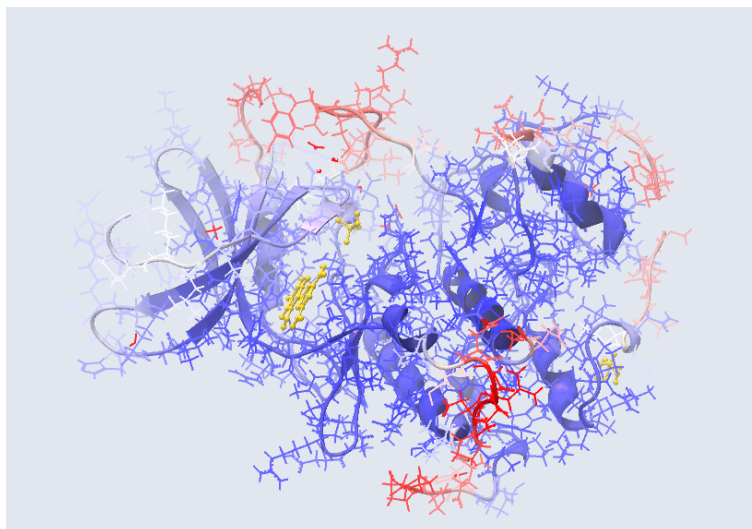


Figure 20.13: *Structure Model of CDK5_HUMAN. The atoms and backbone are colored by temperature, showing uncertain structure in red and well defined structure in blue.*

The template structure is also available from the Proteins category in the Project Tree, but hidden in the initial view. The initial view settings are saved on the Molecule Project as "Initial visualization", and can always be reapplied from the View Settings menu (☰) found in the bottom right corner of the Molecule Project (see section 4.6).

If you have problems viewing 3D structures, please check your system matches the requirements for 3D Viewers. See section 1.3.

20.6.2 Model structure

Protein coloring to visualize local structural uncertainties

The default coloring scheme for modeled structures in *CLC Genomics Workbench* is "Color by Temperature". This coloring indicates the uncertainty or disorder of each atom position in the structure.

For crystal structures, the temperature factor (also called the B-factor) is given in the PDB file as a measure of the uncertainty or disorder of each atom position. The temperature factor has the unit \AA^2 , and is typically in the range [0, 100].

The temperature color scale ranges from blue (0) over white (50) to red (100) (see section 17.3.1).

For structure models created in *CLC Genomics Workbench*, the temperature factor assigned to each atom combines three sources of positional uncertainty:

- **PDB Temp.** The atom position uncertainty for the template structure, represented by the temperature factor of the backbone atoms in the template structure.
- **P(alignment)** The probability that the alignment of a residue in the query sequence to a particular position on the structure is correct.
- **Clash?** It is evaluated if atoms in the structure model seem to clash, thereby indicating a problem with the model.

The three aspects are combined to give a temperature value between zero and 100, as illustrated in figure 20.14 and 20.15.

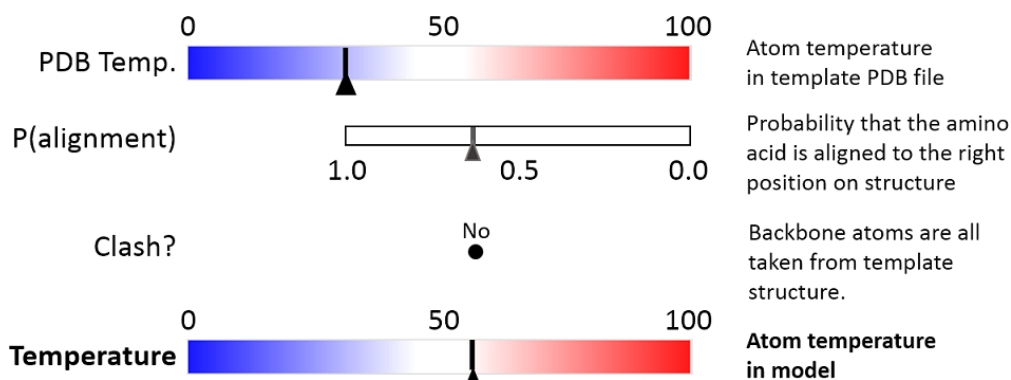


Figure 20.14: Evaluation of temperature color for backbone atoms in structure models.

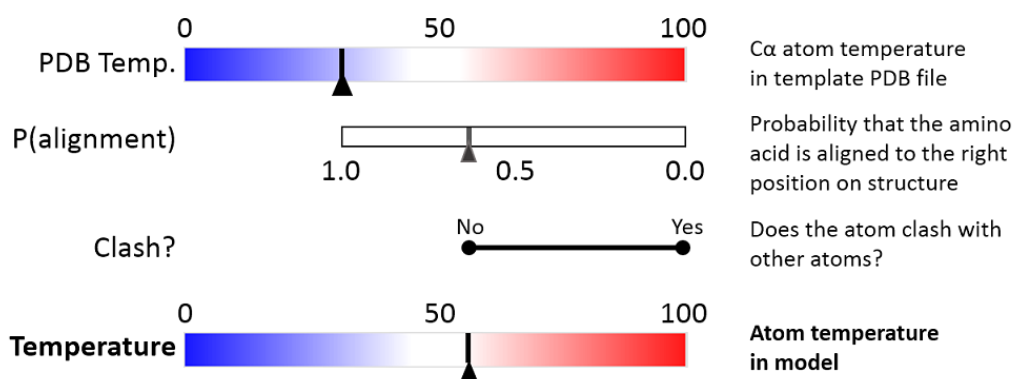


Figure 20.15: Evaluation of temperature color for side chain atoms in structure models.

When holding the mouse over an atom, the Property Viewer in the Side Panel will show various information about the atom. For atoms in structure models, the contributions to the assigned temperature are listed as seen in figure 20.16.

Note: For NMR structures, the temperature factor is set to zero in the PDB file, and the "Color by Temperature" will therefore suggest that the structure is more well determined than is actually the case.

P(alignment) Alignment error is one of the largest causes of model inaccuracy, particularly when the model is built from a template sharing low sequence identity (e.g. lower than 60%). Misaligning a single amino acid by one position will cause a ca. 3.5 Å shift of its atoms from their true positions.

Visualization	Property viewer
Atom selection (1 atom)	
Molecule	Model (CDK5_HUMAN)
Residue	LEU 98 (A)
Name	CB (Carbon)
Hybridization	SP3
Charge	0.00
Source: Modeled	
Temperature	99.15
PDB Temp.	21.85
P(alignment)	0.01
Clash?	No
Occupancy	0.00

Figure 20.16: Information displayed in the Side Panel Property viewer for a modeled atom.

The estimate of the probability that two amino acids are correctly aligned, $P(\text{alignment})$, is obtained by averaging over all the possible alignments between two sequences, similar to [Knudsen and Miyamoto, 2003].

This allows local alignment uncertainty to be detected even in similar sequences. For example the position of the D in this alignment:

```
Template  GGACDAEDRSTRSTACE---GG
Target    GGACD---RSTRSTACEKLMGG
```

is uncertain, because an alternative alignment is as likely:

```
Template  GGACDAEDRSTRSTACE---GG
Target    GGAC---DRSTRSTACEKLMGG
```

Clash? Clashes are evaluated separately for each atom in a side chain. If the atom is considered to clash, it will be assigned a temperature of 100.

Note: Clashes within the modeled protein chain as well as with all other molecules in the downloaded PDB file (except water) are considered.

Ranking structures

The protein sequence of the gene affected by the variant (the query sequence) is BLASTed against the protein structure sequence database (section 32.5.4).

A *template quality score* is calculated for the available structures found for the query sequence. The purpose of the score is to rank structures considering both their quality and their homology to the query sequence.

The five descriptors contributing to the score are:

- E-value
- % Match identity
- % Coverage
- Resolution (of crystal structure)

- Free R-value (R_{free} of crystal structure)

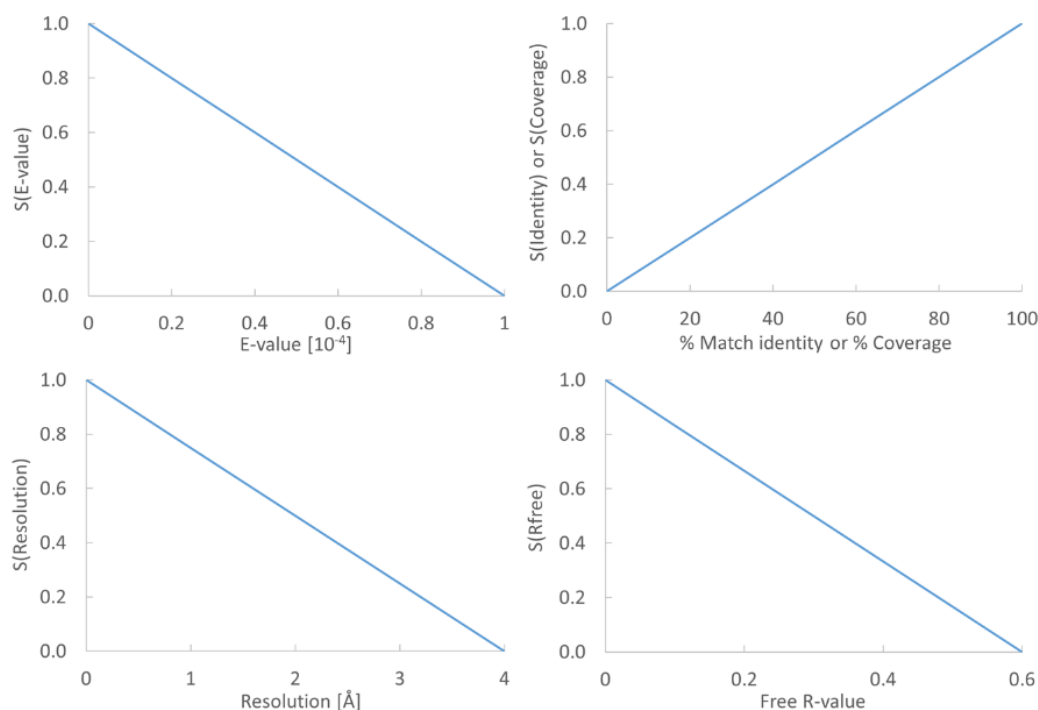


Figure 20.17: From the E-value, % Match identity, % Coverage, Resolution, and Free R-value, the contributions to the "Template quality score" are determined from the linear functions shown in the graphs.

Each of the five descriptors are scaled to $[0,1]$, based on the linear functions seen in figure 20.17. The five scaled descriptors are combined into the *template quality score*, weighting them to emphasize homology over structure qualities.

$$\text{Template quality score} = 3 \cdot S_{\text{E-value}} + 3 \cdot S_{\text{Identity}} + 1.5 \cdot S_{\text{Coverage}} + S_{\text{Resolution}} + 0.5 \cdot S_{\text{Rfree}}$$

E-value is a measure of the quality of the match returned from the BLAST search. You can read more about BLAST and E-values in section 16.5.

% Match identity is the identity between the query sequence and the BLAST hit in the matched region. It is evaluated as

$$\% \text{ Match identity} = 100\% \cdot (\text{Identity in BLAST alignment}) / L_B$$

where L_B is the length of the BLAST alignment of the matched region, as indicated in figure 20.18, and "Identity in BLAST alignment" is the number of identical positions in the matched region.

% Coverage indicates how much of the query sequence has been covered by a given BLAST hit (see figure 20.18). It is evaluated as

$$\% \text{ Coverage} = 100\% \cdot (L_B - L_G) / L_Q$$

where L_G is the total length of gaps in the BLAST alignment and L_Q is the length of the query sequence.

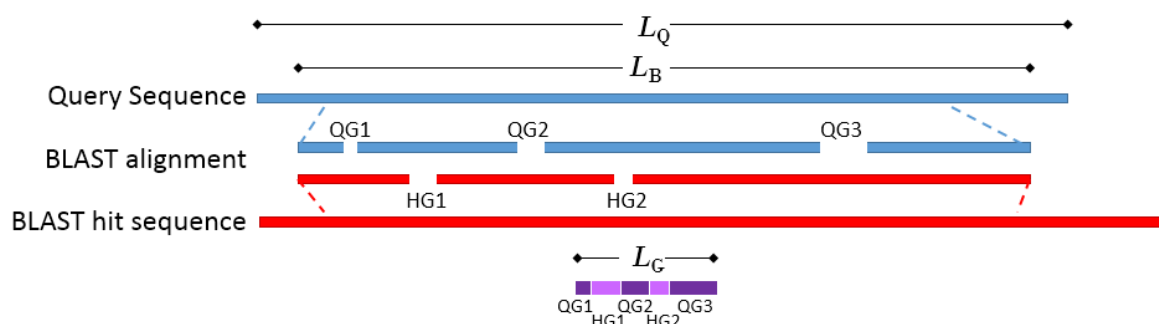


Figure 20.18: Schematic of a query sequence matched to a BLAST hit. L_Q is the length of the query sequence, L_B is the length of the BLAST alignment of the matched region, QG1-3 are gaps in the matched region of the query sequence, HG1-2 are gaps in the matched region of the BLAST hit sequence, L_G is the total length of gaps in the BLAST alignment.

The **resolution** of a crystal structure is related to the size of structural features that can be resolved from the raw experimental data.

R_{free} is used to assess possible overmodeling of the experimental data.

Resolution and R_{free} are only given for crystal structures. NMR structures will therefore usually be ranked lower than crystal structures. Likewise, structures where R_{free} has not been given will tend to receive a lower rank. This often coincides with structures of older date.

How a model structure is created

A structure model is created by mapping the query sequence onto the template structure based on a sequence alignment (see figure 20.19):

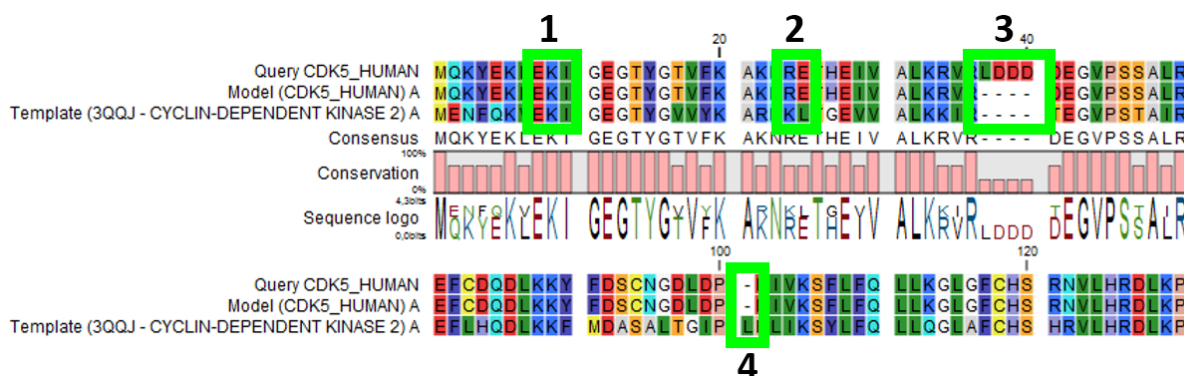


Figure 20.19: Sequence alignment mapping query sequence (Query CDK5_HUMAN) to the structure with sequence "Template(3QQJ - CYCLIN-DEPENDENT KINASE 2)", producing a structure with sequence "Model(CDK5_HUMAN)". Examples are highlighted: 1. Identical amino acids, 2. Amino acid changes, 3. Amino acids in query sequence not aligned to a position on the template structure, and 4. Amino acids on the template structure, not aligned to query sequence.

- For identical amino acids (example 1 in figure 20.19) => Copy atom positions from the PDB file. If the side chain is missing atoms in the PDB file, the side chain is rebuilt (section 20.6.2).

- For amino acid changes (example 2 in figure 20.19) => Copy backbone atom positions from the PDB file. Model side chain atom positions to match the query sequence (section 20.6.2).
- For amino acids in the query sequence not aligned to a position on the template structure (example 3 in figure 20.19) => No atoms are modeled. The model backbone will have a gap at this position and a "Structure modeling" issue is raised (see section 17.1.4).
- For amino acids on the template structure, not aligned to the query sequence (example 4 in figure 20.19) => The residues are deleted from the structure and a "Structure modeling" issue is raised (see section 17.1.4).

How side chains are modeled

Amino acid side chains tend to assume one of a discrete number of "rotamer" conformations. The rotamers used in *CLC Genomics Workbench* have been calculated from a non-redundant set of high-resolution crystal structures.

Side chains are modeled using a heat bath Monte Carlo simulated annealing algorithm, similar to the OPUS-Rota method [Lu et al., 2008]. The algorithm consists of approximately 100 cycles of simulation. In a single cycle, rotamers are selected for each side chain with a probability according to their energy. As the simulation proceeds, the selection increasingly favors the rotamers with the lowest energy, and the algorithm converges.

A local minimization of the modeled side chains is then carried out, to reduce unfavorable interactions with the surroundings.

Calculating the energy of a side chain rotamer

The total energy is composed of several terms:

- Statistical potential: This score accounts for interactions between the given side chain and the local backbone, and is estimated from a database of high-resolution crystal structures. It depends only on the rotamer and the local backbone dihedral angles ϕ and ψ .
- Atom interaction potential: This score is used to evaluate the interaction between a given side chain atom and its surroundings.
- Disulfide potential: Only applies to cysteines. It follows the form used in the RASP program [Miao et al., 2011] and serves to allow disulfide bridges between cysteine residues. It penalizes deviations from ideal disulfide geometry. A distance filter is applied to determine if the disulfide potential should be used, and when it is applied the atom interaction potential between the two sulfur atoms is turned off. Note that disulfide bridges are not formed between separate chains.

Note: The atom interaction potential considers interactions within the modeled protein chain as well as with all other molecules in the downloaded PDB file (except water).

Local minimization of side chain

After applying a side chain rotamer from the library to the backbone, a local minimization may be carried out for rotations around single bonds in the side chain.

The potential to minimize with respect to bond rotation is composed of the following terms:

- Atom interaction potential: Same as for calculating the energy of a rotamer.
- Disulfide potential: Same as for calculating the energy of a rotamer.
- Harmonic potential: This penalizes small deviations from ideal rotamers according to a harmonic potential. This is motivated by the concept of a rotamer representing a minimum energy state for a residue without external interactions.

20.7 Secondary structure prediction

An important issue when trying to understand protein function is to know the actual structure of the protein. Many questions that are raised by molecular biologists are directly targeted at protein structure. The alpha-helix forms a coiled rod like structure whereas a beta-sheet show an extended sheet-like structure. Some proteins are almost devoid of alpha-helices such as chymotrypsin (PDB_ID: 1AB9) whereas others like myoglobin (PDB_ID: 101M) have a very high content of alpha-helices.

With *CLC Genomics Workbench* one can predict the secondary structure of proteins very fast. Predicted elements are alpha-helix, beta-sheet (same as beta-strand) and other regions.

Based on extracted protein sequences from the Protein Data Bank (<https://www.rcsb.org/>) a hidden Markov model (HMM) was trained and evaluated for performance. Machine learning methods have shown superior when it comes to prediction of secondary structure of proteins [Rost, 2001]. By far the most common structures are Alpha-helices and beta-sheets which can be predicted, and predicted structures are automatically added to the query as annotation which later can be edited.

In order to predict the secondary structure of proteins:

Tools | Classical Sequence Analysis (📁) | Protein Analysis (📁) | Predict secondary structure (🌀)

This opens the dialog displayed in figure 20.20:

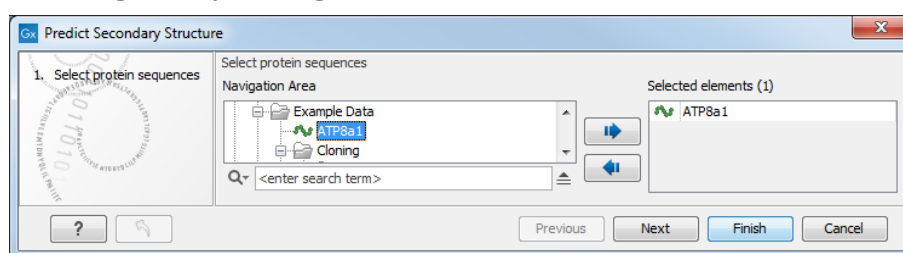


Figure 20.20: Choosing one or more protein sequences for secondary structure prediction.

If a sequence was selected before running the tool, that sequence will be listed in the **Selected Elements** pane of the dialog. Use the arrows to add or remove sequences or sequence lists from the selected elements.

You can perform the analysis on several protein sequences at a time. This will add annotations to all the sequences and open a view for each sequence.

Click **Finish** to start the tool.

After running the prediction as described above, the protein sequence will show predicted alpha-helices and beta-sheets as annotations on the original sequence (see figure 20.21).

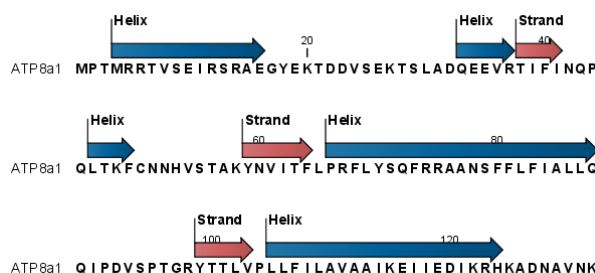


Figure 20.21: Alpha-helices and beta-strands shown as annotations on the sequence.

Each annotation will carry a tooltip note saying that the corresponding annotation is predicted with *CLC Genomics Workbench*. Additional notes can be added through the **Edit Annotation** (🔧) right-click mouse menu. See section 15.3.2.

Undesired alpha-helices or beta-sheets can be removed through the **Delete Annotation** (🗑️) right-click mouse menu. See section 15.3.5.

20.8 Protein report

CLC Genomics Workbench is able to produce protein reports, a collection of some of the protein analyses described elsewhere in this manual.

To create a protein report do the following:

Tools | Classical Sequence Analysis (🔧) | Protein Analysis (📊) | Create Protein Report (📄)

This opens a dialog where you can choose which proteins to create a report for. If you had already selected a sequence in the Navigation Area before running the tool, that sequence will be listed in the **Selected Elements** pane. Use the arrows to add and remove elements from the list of selected elements.

In the next dialog, you can choose which analyses you want to include in the report. The following list shows which analyses are available and explains where to find more details.

- **Sequence statistics.** Will produce a section called Protein statistics, as described in section 18.6.1.
- **Protein charge plot.** Plot of charge as function of pH, see section 20.1.
- **Hydrophobicity plot.** See section 20.3.
- **Complexity plot.** See section 18.5.
- **Dot plot.** See section 18.4.
- **Secondary structure prediction.** See section 20.7.
- **Pfam domain search.** See section 20.5.

- **BLAST against NCBI databases.** See section 16.1.1.

When you have selected the relevant analyses, click **Next**. In the following dialogs, adjust the parameters for the different analyses you selected. The parameters are explained in more details in the relevant chapters or sections (mentioned in the list above).

For sequence statistics:

- **Individual Statistics Layout.** **Comparative** is disabled because reports are generated for one protein at a time.
- **Include Background Distribution of Amino Acids.** Includes distributions from different organisms. Background distributions are calculated from UniProt www.uniprot.org version 6.0, dated September 13 2005.

For hydrophobicity plots:

- **Hydrophobicity scales.** Lets you choose between different scales.
- **Window size.** Width of window on sequence (it must be an odd number).

For complexity plots:

- **Window size.** Width of window on sequence (must be odd).

For dot plots:

- **Score model.** Different scoring matrices.
- **Window size.** Width of window on sequence.

For Pfam domain search:

- **Database and search type** lets you choose different databases and specify the search for full domains or fragments.
- **Significance cutoff** lets you set your E-value.

For BLAST against NCBI databases:

- **Program** lets you choose between different BLAST programs.
- **Database** lets you limit your search to a particular database.
- **Genetic code** lets you choose a genetic code for the sequence or the database.

Also set the BLAST parameters as explained in section 16.1.1.

An example of Protein report can be seen in figure 20.22.

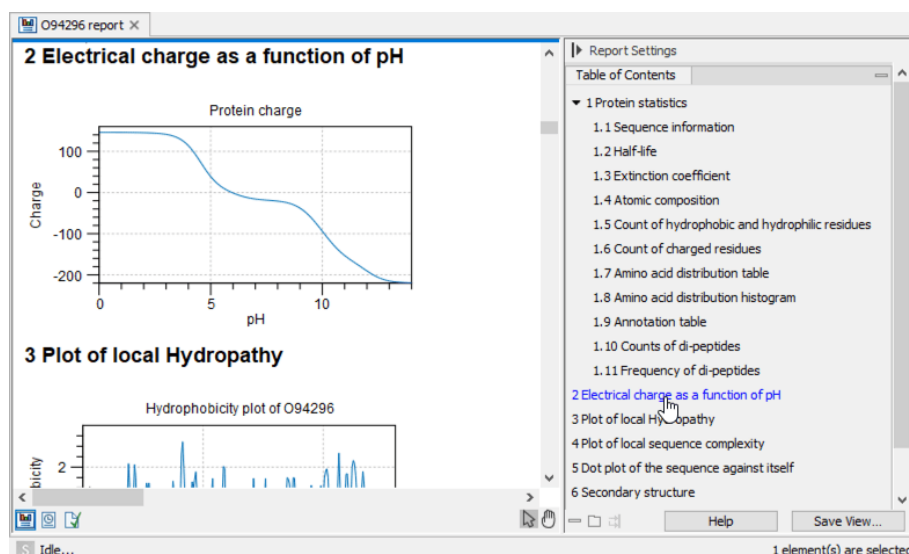


Figure 20.22: A protein report. There is a Table of Contents in the Side Panel that makes it easy to browse the report.

By double clicking a graph in the output, this graph is shown in a different view (CLC Genomics Workbench generates another tab). The report output and the new graph views can be saved by dragging the tab into the **Navigation Area**.

The content of the tables in the report can be copy/pasted out of the program and e.g. into Microsoft Excel. You can also **Export** (📎) the report in Excel format.

20.9 Reverse translation from protein into DNA

A protein sequence can be back-translated into DNA using *CLC Genomics Workbench*. Due to degeneracy of the genetic code every amino acid could translate into several different codons (only 20 amino acids but 64 different codons). Thus, the program offers a number of choices for determining which codons should be used. These choices are explained in this section. For background information see section 20.9.1.

In order to make a reverse translation:

Tools | Classical Sequence Analysis (🔍) | Protein Analysis (🧬) | Reverse Translate (🔄)

This opens the dialog displayed in figure 20.23:

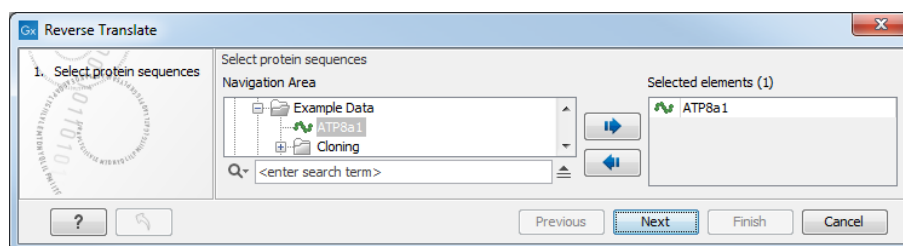


Figure 20.23: Choosing a protein sequence for reverse translation.

If a sequence was selected before running the tool, that sequence will be listed in the **Selected Elements** pane of the dialog. Use the arrows to add or remove sequences or sequence lists from the selected elements. You can translate several protein sequences at a time.

Adjust the parameters for the translation in the dialog shown in figure 20.24.

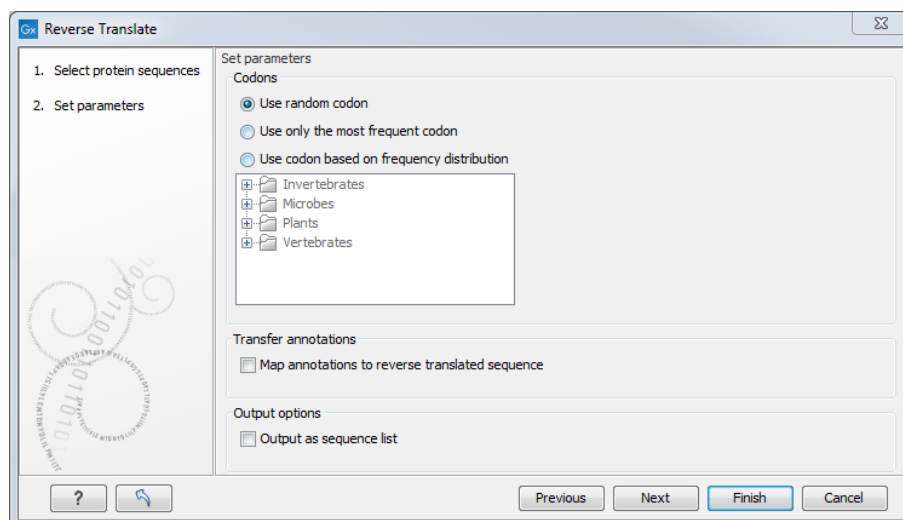


Figure 20.24: Choosing parameters for the reverse translation.

- **Use random codon.** This will randomly back-translate an amino acid to a codon assuming the genetic code to be 1, but without using the codon frequency tables. Every time you perform the analysis you will get a different result.
- **Use only the most frequent codon.** On the basis of the selected translation table, this parameter/option will assign the codon that occurs most often. When choosing this option,

the results of performing several reverse translations will always be the same, contrary to the other two options.

- **Use codon based on frequency distribution.** This option is a mix of the other two options. The selected translation table is used to attach weights to each codon based on its frequency. The codons are assigned randomly with a probability given by the weights. A more frequent codon has a higher probability of being selected. Every time you perform the analysis, you will get a different result. This option yields a result that is closer to the translation behavior of the organism (assuming you choose an appropriate codon frequency table).
- **Map annotations to reverse translated sequence.** If this checkbox is checked, then all annotations on the protein sequence will be mapped to the resulting DNA sequence. In the tooltip on the transferred annotations, there is a note saying that the annotation derives from the original sequence.

The **Codon Frequency Table** is used to determine the frequencies of the codons. Select a frequency table from the list that fits the organism you are working with. A translation table of an organism is created on the basis of counting all the codons in the coding sequences. Every codon in a **Codon Frequency Table** has its own count, frequency (per thousand) and fraction which are calculated in accordance with the occurrences of the codon in the organism. The tables provided were made using Codon Usage database <https://www.kazusa.or.jp/codon/> that was built on The NCBI-GenBank Flat File Release 160.0 [June 15 2007]. You can customize the list of codon frequency tables for your installation, see Appendix L.

Click **Finish** to start the tool. The newly created nucleotide sequence is shown, and if the analysis was performed on several protein sequences, there will be a corresponding number of views of nucleotide sequences.

20.9.1 Bioinformatics explained: Reverse translation

In all living cells containing hereditary material such as DNA, a transcription to mRNA and subsequent a translation to proteins occur. This is of course simplified but is in general what is happening in order to have a steady production of proteins needed for the survival of the cell. In bioinformatics analysis of proteins it is sometimes useful to know the ancestral DNA sequence in order to find the genomic localization of the gene. Thus, the translation of proteins back to DNA/RNA is of particular interest, and is called reverse translation or back-translation.

The Genetic Code In 1968 the Nobel Prize in Medicine was awarded to Robert W. Holley, Har Gobind Khorana and Marshall W. Nirenberg for their interpretation of the Genetic Code (<http://nobelprize.org/medicine/laureates/1968/>). The Genetic Code represents translations of all 64 different codons into 20 different amino acids. Therefore it is no problem to translate a DNA/RNA sequence into a specific protein. But due to the degeneracy of the genetic code, several codons may code for only one specific amino acid. This can be seen in the table below. After the discovery of the genetic code it has been concluded that different organism (and organelles) have genetic codes which are different from the "standard genetic code". Moreover, the amino acid alphabet is no longer limited to 20 amino acids. The 21st amino acid, selenocysteine, is encoded by an 'UGA' codon which is normally a stop codon. The

discrimination of a selenocysteine over a stop codon is carried out by the translation machinery. Selenocysteines are very rare amino acids.

The table below shows the Standard Genetic Code which is the default translation table.

TTT F Phe	TCT S Ser	TAT Y Tyr	TGT C Cys
TTC F Phe	TCC S Ser	TAC Y Tyr	TGC C Cys
TTA L Leu	TCA S Ser	TAA * Ter	TGA * Ter
TTG L Leu i	TCG S Ser	TAG * Ter	TGG W Trp
CTT L Leu	CCT P Pro	CAT H His	CGT R Arg
CTC L Leu	CCC P Pro	CAC H His	CGC R Arg
CTA L Leu	CCA P Pro	CAA Q Gln	CGA R Arg
CTG L Leu i	CCG P Pro	CAG Q Gln	CGG R Arg
ATT I Ile	ACT T Thr	AAT N Asn	AGT S Ser
ATC I Ile	ACC T Thr	AAC N Asn	AGC S Ser
ATA I Ile	ACA T Thr	AAA K Lys	AGA R Arg
ATG M Met i	ACG T Thr	AAG K Lys	AGG R Arg
GTT V Val	GCT A Ala	GAT D Asp	GGT G Gly
GTC V Val	GCC A Ala	GAC D Asp	GGC G Gly
GTA V Val	GCA A Ala	GAA E Glu	GGA G Gly
GTG V Val	GCG A Ala	GAG E Glu	GGG G Gly

Solving the ambiguities of reverse translation A particular protein follows from the translation of a DNA sequence whereas the reverse translation need not have a specific solution according to the Genetic Code. The Genetic Code is degenerate which means that a particular amino acid can be translated into more than one codon. Hence there are ambiguities of the reverse translation.

In order to solve these ambiguities of reverse translation you can define how to prioritize the codon selection, e.g:

- Choose a codon randomly.
- Select the most frequent codon in a given organism.
- Randomize a codon, but with respect to its frequency in the organism.

As an example we want to translate an alanine to the corresponding codon. Four different codons can be used for this reverse translation; GCU, GCC, GCA or GCG. By picking either one by random choice we will get an alanine.

The most frequent codon, coding for an alanine in *E. coli* is GCG, encoding 33.7% of all alanines. Then comes GCC (25.5%), GCA (20.3%) and finally GCU (15.3%). The data are retrieved from the Codon usage database, see below. Always picking the most frequent codon does not necessarily give the best answer.

By selecting codons from a distribution of calculated codon frequencies, the DNA sequence obtained after the reverse translation, holds the correct (or nearly correct) codon distribution. It

should be kept in mind that the obtained DNA sequence is not necessarily identical to the original one encoding the protein in the first place, due to the degeneracy of the genetic code.

In order to obtain the best possible result of the reverse translation, one should use the codon frequency table from the correct organism or a closely related species. The codon usage of the mitochondrial chromosome are often different from the native chromosome(s), thus mitochondrial codon frequency tables should only be used when working specifically with mitochondria.

Other useful resources

The Genetic Code at NCBI:

<http://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi?mode=c>

Codon usage database:

<http://www.kazusa.or.jp/codon/>

Wikipedia on the genetic code

http://en.wikipedia.org/wiki/Genetic_code

20.10 Proteolytic cleavage detection

Given a protein sequence, *CLC Genomics Workbench* detects proteolytic cleavage sites in accordance with detection parameters and shows the detected sites as annotations on the sequence as well as in a table below the sequence view.

Detection of proteolytic cleavage sites is initiated by:

Tools | Classical Sequence Analysis (📁) | Protein Analysis (📁) | Proteolytic Cleavage (🔪)

If you are connected to a server, the first wizard step will ask you where you would like to run the analysis. Next, you must provide one or more sequences figure 20.25.

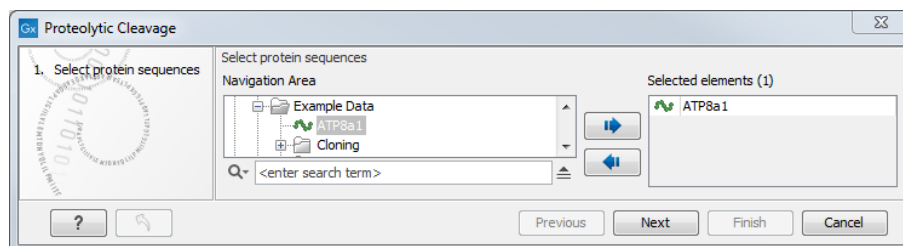


Figure 20.25: Choosing a protein sequence for proteolytic cleavage.

In the second dialog, you can select proteolytic cleavage enzymes. Presently, the list contains the enzymes shown in figure 20.26. The full list of enzymes and their cleavage patterns can be seen in Appendix, section D.

You can then set parameters for the detection. This limits the number of detected cleavages (figure 20.27).

- **Min. and max. number of cleavage sites.** Certain proteolytic enzymes cleave at many positions in the amino acid sequence. For instance proteinase K cleaves at nine different amino acids, regardless of the surrounding residues. Thus, it can be very useful to limit the number of actual cleavage sites before running the analysis.

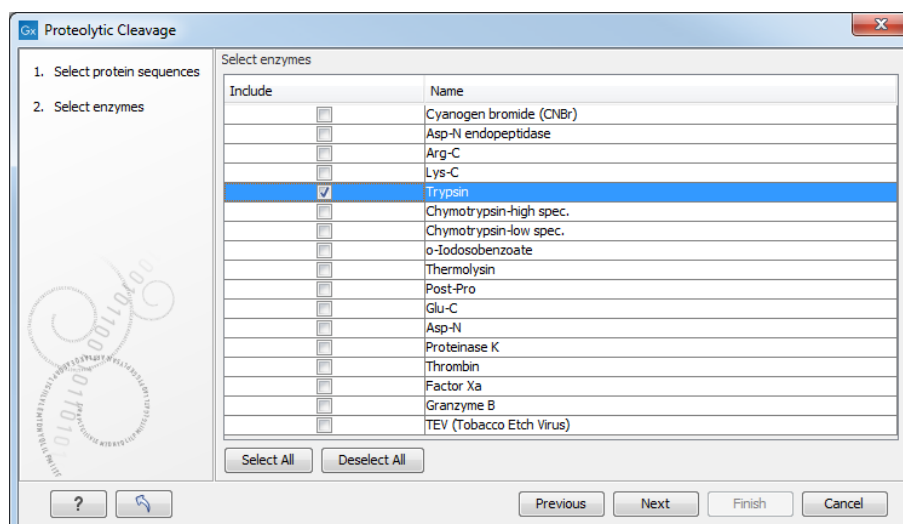


Figure 20.26: Setting parameters for proteolytic cleavage detection.

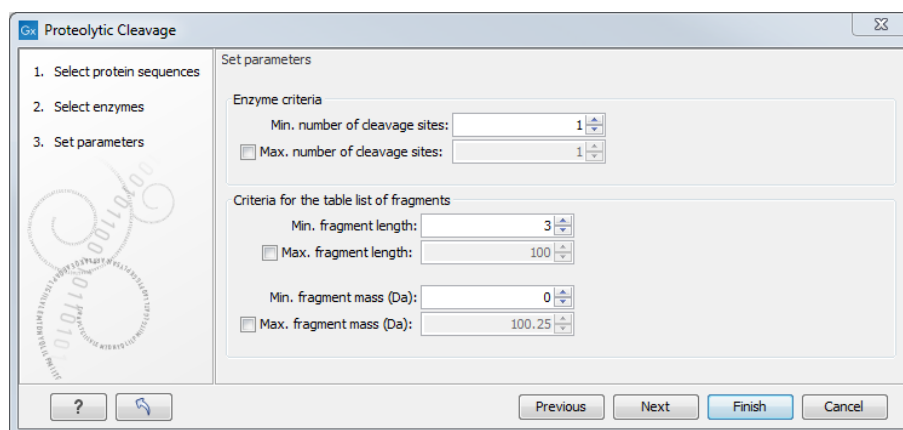


Figure 20.27: Setting parameters for proteolytic cleavage detection.

- **Min. and max. fragment length** Likewise, it is possible to limit the output to only display sequence fragments between a chosen length. Both a lower and upper limit can be chosen.
- **Min. and max. fragment mass** The molecular weight is not necessarily directly correlated to the fragment length as amino acids have different molecular masses. For that reason it is also possible to limit the search for proteolytic cleavage sites to mass-range.

For example, if you have one protein sequence but you only want to show which enzymes cut between two and four times. Then you should select "The enzymes has more cleavage sites than 2" and select "The enzyme has less cleavage sites than 4". In the next step you should simply select all enzymes. This will result in a view where only enzymes which cut 2,3 or 4 times are presented.

Click **Finish** to start the tool. The result of the detection is displayed in figure 20.28.

Depending on the settings in the program, the output of the proteolytic cleavage site detection will display two views on the screen. The top view shows the actual protein sequence with the predicted cleavage sites indicated by small arrows. If no labels are found on the arrows they can be enabled by setting the labels in the "annotation layout" in the preference panel. The bottom view shows a text output of the detection, listing the individual fragments and information on

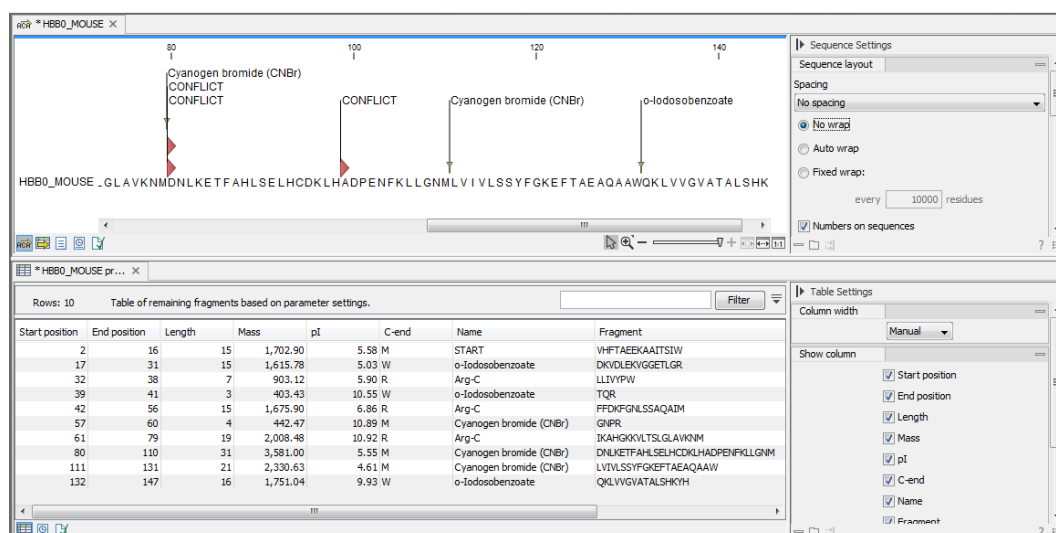


Figure 20.28: The result of the proteolytic cleavage detection.

these.

20.10.1 Bioinformatics explained: Proteolytic cleavage

Proteolytic cleavage is basically the process of breaking the peptide bonds between amino acids in proteins. This process is carried out by enzymes called peptidases, proteases or proteolytic cleavage enzymes.

Proteins often undergo proteolytic processing by specific proteolytic enzymes (proteases/peptidases) before final maturation of the protein. Proteins can also be cleaved as a result of intracellular processing of, for example, misfolded proteins. Another example of proteolytic processing of proteins is secretory proteins or proteins targeted to organelles, which have their signal peptide removed by specific signal peptidases before release to the extracellular environment or specific organelle.

Below a few processes are listed where proteolytic enzymes act on a protein substrate.

- N-terminal methionine residues are often removed after translation.
- Signal peptides or targeting sequences are removed during translocation through a membrane.
- Viral proteins that were translated from a monocistronic mRNA are cleaved.
- Proteins or peptides can be cleaved and used as nutrients.
- Precursor proteins are often processed to yield the mature protein.

Proteolytic cleavage of proteins has shown its importance in laboratory experiments where it is often useful to work with specific peptide fragments instead of entire proteins.

Proteases also have commercial applications. As an example proteases can be used as detergents for cleavage of proteinaceous stains in clothing.

The general nomenclature of cleavage site positions of the substrate were formulated by Schechter and Berger, 1967-68 [Schechter and Berger, 1967], [Schechter and Berger, 1968]. They designate the cleavage site between P1-P1', incrementing the numbering in the N-terminal direction of the cleaved peptide bond (P2, P3, P4, etc..). On the carboxyl side of the cleavage site the numbering is incremented in the same way (P1', P2', P3' etc.). This is visualized in figure 20.29.

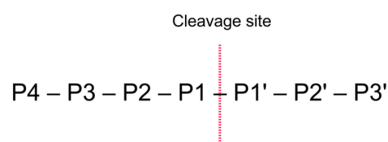


Figure 20.29: Nomenclature of the peptide substrate. The substrate is cleaved between position P1-P1'.

Proteases often have a specific recognition site where the peptide bond is cleaved. As an example trypsin only cleaves at lysine or arginine residues, but it does not matter (with a few exceptions) which amino acid is located at position P1'(carboxyterminal of the cleavage site). Another example is trombin which cleaves if an arginine is found in position P1, but not if a D or E is found in position P1' at the same time. (See figure 20.30).

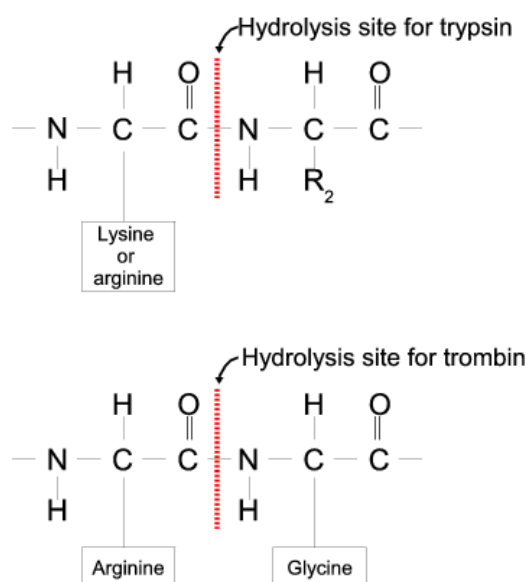


Figure 20.30: Hydrolysis of the peptide bond between two amino acids. Trypsin cleaves unspecifically at lysine or arginine residues whereas trombin cleaves at arginines if asparate or glutamate is absent.

Bioinformatics approaches are used to identify potential peptidase cleavage sites. Fragments can be found by scanning the amino acid sequence for patterns which match the corresponding cleavage site for the protease. When identifying cleaved fragments it is relatively important to know the calculated molecular weight and the isoelectric point.

Other useful resources

The Peptidase Database: <http://merops.sanger.ac.uk/>

Chapter 21

Primers

Contents

21.1 Primer design - an introduction	530
21.1.1 General concept	530
21.1.2 Scoring primers	532
21.2 Setting parameters for primers and probes	532
21.2.1 Primer Parameters	532
21.3 Graphical display of primer information	534
21.3.1 Compact information mode	534
21.3.2 Detailed information mode	535
21.4 Output from primer design	536
21.5 Standard PCR	537
21.5.1 When a single primer region is defined	538
21.5.2 When both forward and reverse regions are defined	539
21.5.3 Standard PCR output table	540
21.6 Nested PCR	541
21.7 TaqMan	542
21.8 Sequencing primers	544
21.9 Alignment-based primer and probe design	545
21.9.1 Specific options for alignment-based primer and probe design	545
21.9.2 Alignment based design of PCR primers	547
21.9.3 Alignment-based TaqMan probe design	549
21.10 Analyze primer properties	550
21.11 Find binding sites and create fragments	551
21.11.1 Binding parameters	552
21.11.2 Results - binding sites and fragments	553
21.12 Order primers	555

CLC Genomics Workbench offers graphically and algorithmically advanced design of primers and probes for various purposes. This chapter begins with a brief introduction to the general concepts of the primer designing process. Then follows instructions on how to adjust parameters for primers, how to inspect and interpret primer properties graphically and how to interpret, save

and analyze the output of the primer design analysis. After a description of the different reaction types for which primers can be designed, the chapter closes with sections on how to match primers with other sequences and how to create a primer order.

21.1 Primer design - an introduction

Primer design can be accessed in two ways:

Tools | Molecular Biology Tools (🔧) | Primers and Probes (📄) | Design Primers (🔍) | OK

or **right-click sequence in Navigation Area | Show | Primer Designer (🔍)**

In the primer view (see figure 21.1), the basic options for viewing the template sequence are the same as for the standard sequence view (see section 15.2 for an explanation of these options). This means that annotations such as known SNPs or exons can be displayed on the template sequence to guide the choice of primer regions. In addition, traces in sequencing reads can be shown along with the structure to guide the re-sequencing of poorly resolved regions.



Figure 21.1: The initial view of the sequence used for primer design.

21.1.1 General concept

The concept of the primer view is that the user first chooses the desired reaction type for the session in the Primer Parameters preference group, e.g. *Standard PCR*. Reflecting the choice of reaction type, it is now possible to select one or more regions on the sequence and to use the right-click mouse menu to designate these as primer or probe regions (see figure 21.2).

When a region is chosen, graphical information about the properties of all possible primers in this region will appear in lines beneath it. By default, information is shown using a compact mode but the user can change to a more detailed mode in the Primer information preference group.

The number of information lines reflects the chosen length interval for primers and probes. In the compact information mode one line is shown for every possible primer-length and each of these lines contain information regarding all possible primers of the given length. At each potential primer starting position, a circular information point is shown which indicates whether the primer fulfills the requirements set in the primer parameters preference group. A green circle indicates a primer which fulfills all criteria and a red circle indicates a primer which fails to meet one or

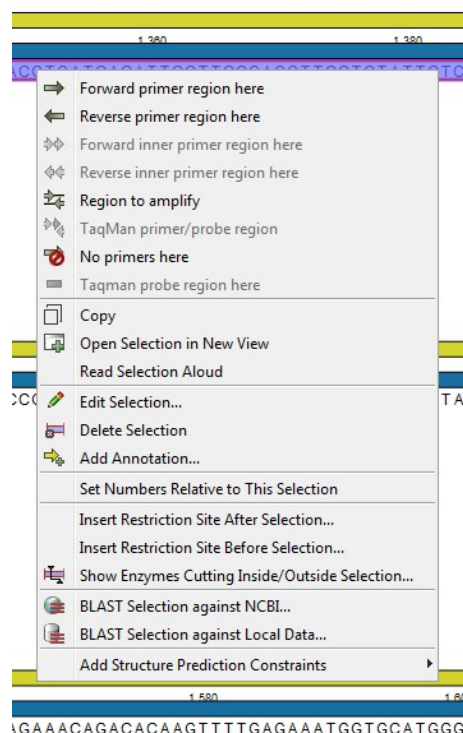


Figure 21.2: Right-click menu allowing you to specify regions for the primer design

more of the set criteria. For more detailed information, place the mouse cursor over the circle representing the primer of interest. A tool-tip will then appear on screen, displaying detailed information about the primer in relation to the set criteria. To locate the primer on the sequence, simply left-click the circle using the mouse.

The various primer parameters can now be varied to explore their effect and the view area will dynamically update to reflect this allowing for a high degree of interactivity in the primer design process.

After having explored the potential primers the user may have found a satisfactory primer and choose to export this directly from the view area using a mouse right-click on the primers information point. This does not allow for any design information to enter concerning the properties of primer/probe pairs or sets e.g. primer pair annealing and T_m difference between primers. If the latter is desired the user can use the **Calculate** button at the bottom of the Primer parameter preference group. This will activate a dialog, the contents of which depends on the chosen mode. Here, the user can set primer-pair specific setting such as allowed or desired T_m difference and view the single-primer parameters which were chosen in the Primer parameters preference group.

Upon pressing finish, an algorithm will generate all possible primer sets and rank these based on their characteristics and the chosen parameters. A list will appear displaying the 100 most high scoring sets and information pertaining to these. The search result can be saved to the navigator. From the result table, suggested primers or primer/probe sets can be explored since clicking an entry in the table will highlight the associated primers and probes on the sequence. It is also possible to save individual primers or sets from the table through the mouse right-click menu. For a given primer pair, the amplified PCR fragment can also be opened or saved using the mouse right-click menu.

21.1.2 Scoring primers

CLC Genomics Workbench employs a proprietary algorithm to rank primer and probe solutions. The algorithm considers both the parameters pertaining to single oligos, such as e.g. the secondary structure score and parameters pertaining to oligo-pairs such as e.g. the oligo pair-annealing score. The ideal score for a solution is 100 and solutions are thus ranked in descending order. Each parameter is assigned an ideal value and a tolerance. Consider for example oligo self-annealing, here the ideal value of the annealing score is 0 and the tolerance corresponds to the maximum value specified in the side panel. The contribution to the final score is determined by how much the parameter deviates from the ideal value and is scaled by the specified tolerance. Hence, a large deviation from the ideal and a small tolerance will give a large deduction in the final score and a small deviation from the ideal and a high tolerance will give a small deduction in the final score.

21.2 Setting parameters for primers and probes

The primer-specific view options and settings are found in the **Primer parameters** preference group in the **Side Panel** to the right of the view (see figure 21.3).

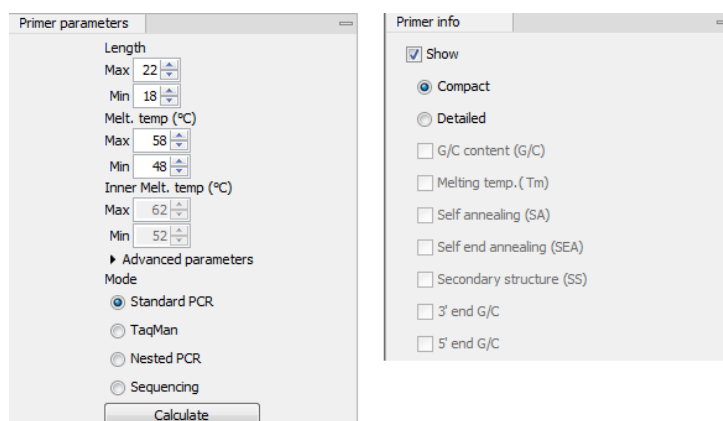


Figure 21.3: The two groups of primer parameters (in the program, the Primer information group is listed below the other group).

21.2.1 Primer Parameters

In this preference group a number of criteria can be set, which the selected primers must meet. All the criteria concern *single primers*, as primer pairs are not generated until the **Calculate** button is pressed. Parameters regarding primer and probe sets are described in detail for each reaction mode (see below).

- **Length.** Determines the length interval within which primers can be designed by setting a maximum and a minimum length. The upper and lower lengths allowed by the program are 50 and 10 nucleotides respectively.
- **Melting temperature.** Determines the temperature interval within which primers must lie. When the *Nested PCR* or *TaqMan* reaction type is chosen, the first pair of melting temperature interval settings relate to the outer primer pair i.e. not the probe. Melting temperatures are calculated by a nearest-neighbor model which considers stacking interactions between

neighboring bases in the primer-template complex. The model uses state-of-the-art thermodynamic parameters [SantaLucia, 1998] and considers the important contribution from the dangling ends that are present when a short primer anneals to a template sequence [Bommarito et al., 2000]. A number of parameters can be adjusted concerning the reaction mixture and which influence melting temperatures (see below). Melting temperatures are corrected for the presence of monovalent cations using the model of [SantaLucia, 1998] and temperatures are further corrected for the presence of magnesium, deoxynucleotide triphosphates (dNTP) and dimethyl sulfoxide (DMSO) using the model of [von Ahsen et al., 2001].

- **Inner melting temperature.** This option is only activated when the *Nested PCR* or *TaqMan* mode is selected. In *Nested PCR* mode, it determines the allowed melting temperature interval for the inner/nested pair of primers, and in *TaqMan* mode it determines the allowed temperature interval for the TaqMan probe.
- **Advanced parameters.** A number of less commonly used options
 - **Buffer properties.** A number of parameters concerning the reaction mixture which influence melting temperatures.
 - * **Primer concentration.** Specifies the concentration of primers and probes in units of nanomoles (nM). Note that in the case of a mix of primers, the concentration here refers to the individual primer and not the combined primers concentration.
 - * **Salt concentration.** Specifies the concentration of monovalent cations ($[Na^+]$, $[K^+]$ and equivalents) in units of millimoles (mM)
 - * **Magnesium concentration.** Specifies the concentration of magnesium cations ($[Mg^{++}]$) in units of millimoles (mM)
 - * **dNTP concentration.** Specifies the combined concentration of all deoxynucleotide triphosphates in units of millimoles (mM)
 - * **DMSO concentration.** Specifies the concentration of dimethyl sulfoxide in units of volume percent ($vol.\%$)
 - **GC content.** Determines the interval of GC content (% C and G nucleotides in the primer) within which primers must lie by setting a maximum and a minimum GC content.
 - **Self annealing.** Determines the maximum self annealing value of all primers and probes. This determines the amount of base-pairing allowed between two copies of the same molecule. The self annealing score is measured in number of hydrogen bonds between two copies of primer molecules, with A-T base pairs contributing 2 hydrogen bonds and G-C base pairs contributing 3 hydrogen bonds.
 - **Self end annealing.** Determines the maximum self end annealing value of all primers and probes. This determines the number of consecutive base pairs allowed between the 3' end of one primer and another copy of that primer. This score is calculated in number of hydrogen bonds (the example below has a score of 4 - derived from 2 A-T base pairs each with 2 hydrogen bonds).

```

AATTCCCTACAATCCCCAAA
      ||
      AAACCCCTAACATCCCTTAA

```

- **Secondary structure.** Determines the maximum score of the optimal secondary DNA structure found for a primer or probe. Secondary structures are scored by the number of hydrogen bonds in the structure, and 2 extra hydrogen bonds are added for each stacking base-pair in the structure.
- **3' end G/C restrictions.** When this checkbox is selected it is possible to specify restrictions concerning the number of G and C molecules in the 3' end of primers and probes. A low G/C content of the primer/probe 3' end increases the specificity of the reaction. A high G/C content facilitates a tight binding of the oligo to the template but also increases the possibility of mispriming. Unfolding the preference groups yields the following options:
 - **End length.** The number of consecutive terminal nucleotides for which to consider the C/G content
 - **Max no. of G/C.** The maximum number of G and C nucleotides allowed within the specified length interval
 - **Min no. of G/C.** The minimum number of G and C nucleotides required within the specified length interval
- **5' end G/C restrictions.** When this checkbox is selected it is possible to specify restrictions concerning the number of G and C molecules in the 5' end of primers and probes. A high G/C content facilitates a tight binding of the oligo to the template but also increases the possibility of mis-priming. Unfolding the preference groups yields the same options as described above for the 3' end.
- **Mode.** Specifies the reaction type for which primers are designed:
 - **Standard PCR.** Used when the objective is to design primers, or primer pairs, for PCR amplification of a single DNA fragment.
 - **Nested PCR.** Used when the objective is to design two primer pairs for nested PCR amplification of a single DNA fragment.
 - **Sequencing.** Used when the objective is to design primers for DNA sequencing.
 - **TaqMan.** Used when the objective is to design a primer pair and a probe for TaqMan quantitative PCR.

Each mode is described further below.

- **Calculate.** Pushing this button will activate the algorithm for designing primers

21.3 Graphical display of primer information

The primer information settings are found in the **Primer information** preference group in the **Side Panel** to the right of the view (see figure 21.3).

There are two different ways to display the information relating to a single primer, the detailed and the compact view. Both are shown below the primer regions selected on the sequence.

21.3.1 Compact information mode

This mode offers a condensed overview of all the primers that are available in the selected region. When a region is chosen primer information will appear in lines beneath it (see figure 21.4).

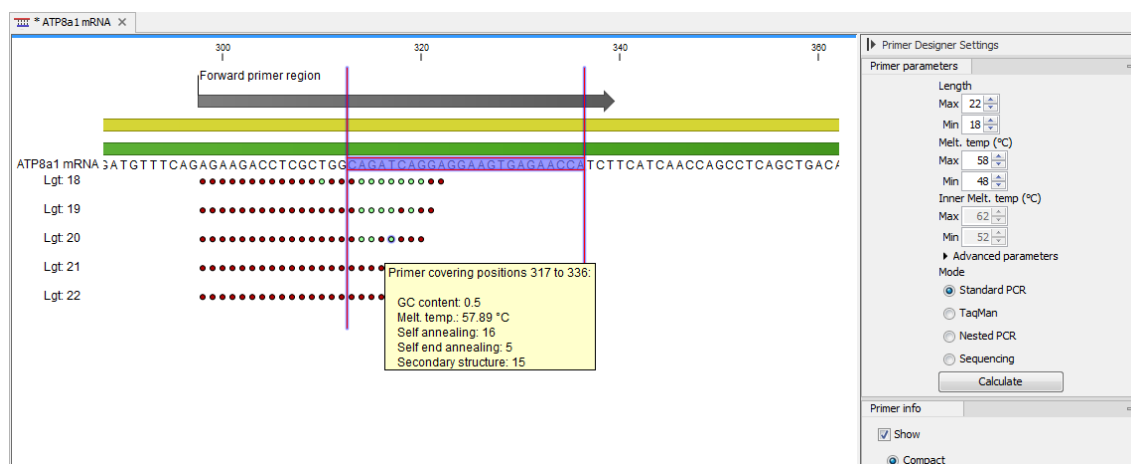


Figure 21.4: Compact information mode.

The number of information lines reflects the chosen length interval for primers and probes. One line is shown for every possible primer-length, if the length interval is widened more lines will appear. At each potential primer starting position a circle is shown which indicates whether the primer fulfills the requirements set in the primer parameters preference group. A green primer indicates a primer which fulfills all criteria and a red primer indicates a primer which fails to meet one or more of the set criteria. For more detailed information, place the mouse cursor over the circle representing the primer of interest. A tool-tip will then appear on screen displaying detailed information about the primer in relation to the set criteria. To locate the primer on the sequence, simply left-click the circle using the mouse.

The various primer parameters can now be varied to explore their effect and the view area will dynamically update to reflect this. If e.g. the allowed melting temperature interval is widened more green circles will appear indicating that more primers now fulfill the set requirements and if e.g. a requirement for 3' G/C content is selected, red circles will appear at the starting points of the primers which fail to meet this requirement.

21.3.2 Detailed information mode

In this mode a very detailed account is given of the properties of all the available primers. When a region is chosen primer information will appear in groups of lines beneath it (see figure 21.5).

The number of information-line-groups reflects the chosen length interval for primers and probes. One group is shown for every possible primer length. Within each group, a line is shown for every primer property that is selected from the checkboxes in the primer information preference group. Primer properties are shown at each potential primer starting position and are of two types:

Properties with numerical values are represented by bar plots. A green bar represents the starting point of a primer that meets the set requirement and a red bar represents the starting point of a primer that fails to meet the set requirement:

- G/C content
- Melting temperature
- Self annealing score

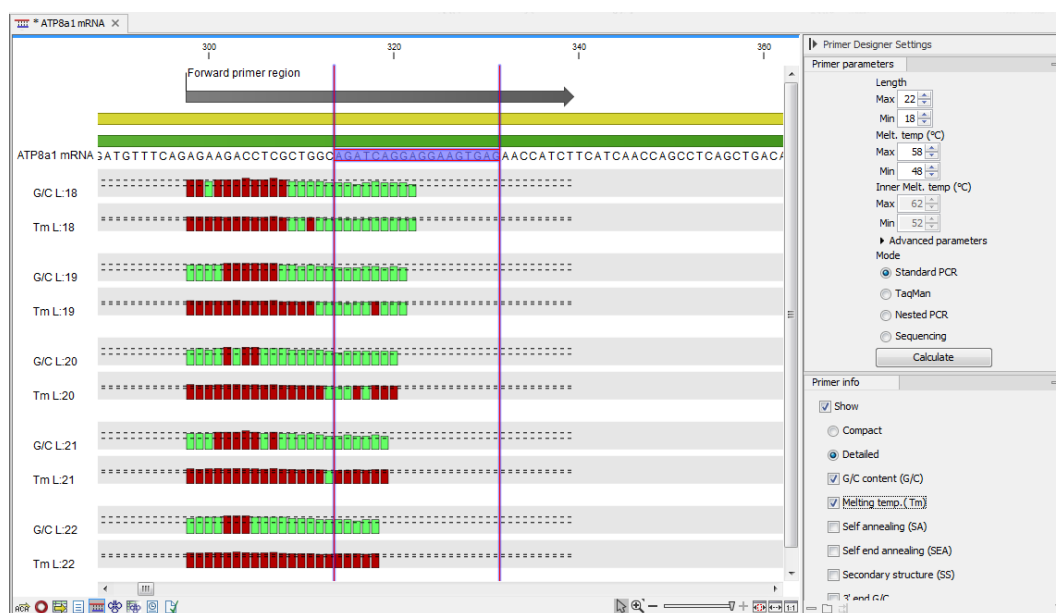


Figure 21.5: Detailed information mode.

- Self end annealing score
- Secondary structure score

Properties with Yes - No values. If a primer meets the set requirement a green circle will be shown at its starting position and if it fails to meet the requirement a red dot is shown at its starting position:

- C/G at 3' end
- C/G at 5' end

Common to both sorts of properties is that mouse clicking an information point (filled circle or bar) will cause the region covered by the associated primer to be selected on the sequence.

21.4 Output from primer design

The output generated by the primer design algorithm is a table of proposed primers or primer pairs with the accompanying information (see figure 21.6).

In the preference panel of the table, it is possible to customize which columns are shown in the table. See the sections below on the different reaction types for a description of the available information.

The columns in the output table can be sorted by the present information. For example the user can choose to sort the available primers by their score (default) or by their self annealing score, simply by right-clicking the column header.

The output table interacts with the accompanying primer editor such that when a proposed combination of primers and probes is selected in the table the primers and probes in this solution are highlighted on the sequence.

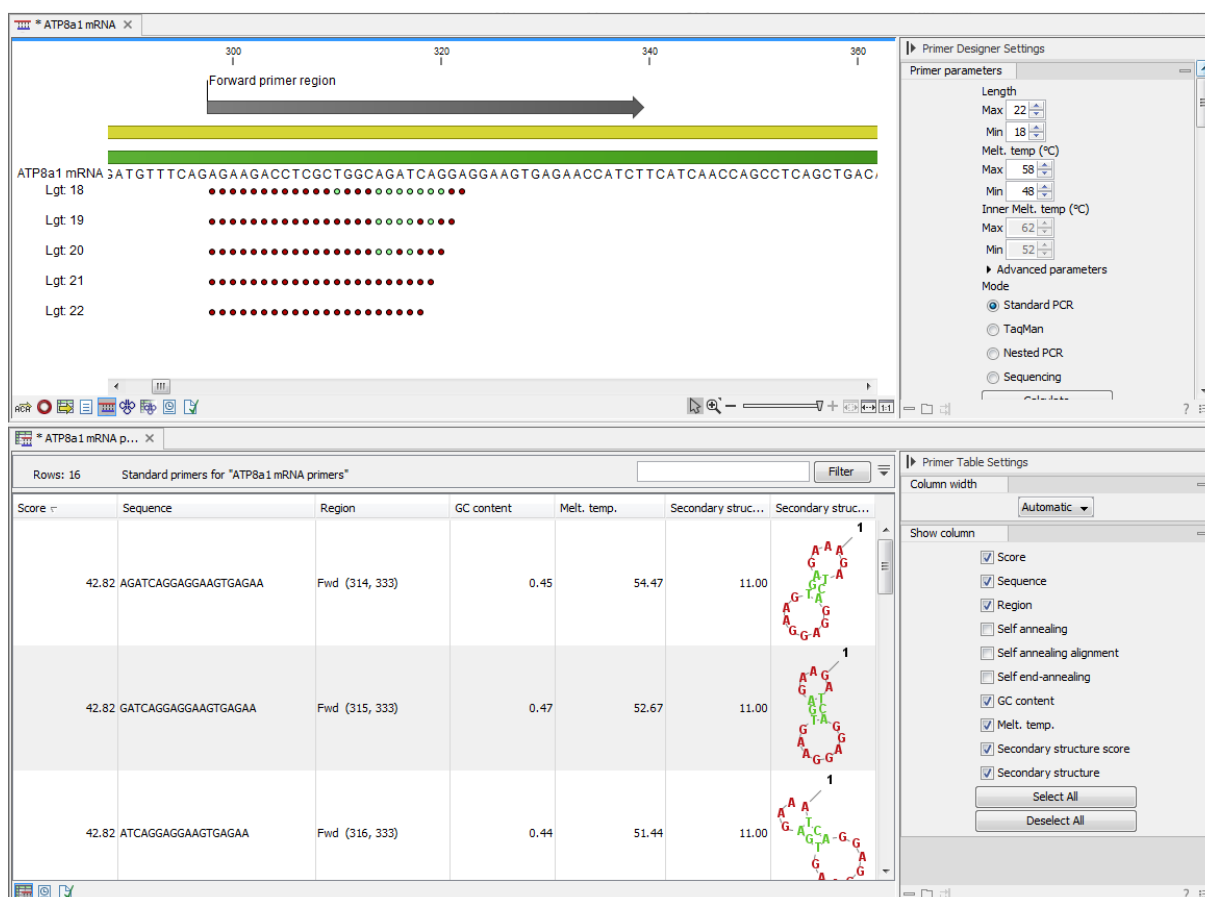


Figure 21.6: Proposed primers.

Saving primers Primer solutions in a table row can be saved by selecting the row and using the right-click mouse menu. This opens a dialog that allows the user to save the primers to the desired location. Primers and probes are saved as DNA sequences in the program. This means that all available DNA analyzes can be performed on the saved primers. Furthermore, the primers can be edited using the standard sequence view to introduce e.g. mutations and restriction sites.

Saving PCR fragments The PCR fragment generated from the primer pair in a given table row can also be saved by selecting the row and using the right-click mouse menu. This opens a dialog that allows the user to save the fragment to the desired location. The fragment is saved as a DNA sequence and the position of the primers is added as annotation on the sequence. The fragment can then be used for further analysis and included in e.g. an in-silico cloning experiment using the cloning editor.

Adding primer binding annotation You can add an annotation to the template sequence specifying the binding site of the primer: Right-click the primer in the table and select **Mark primer annotation on sequence**.

21.5 Standard PCR

This mode is used to design primers for a PCR amplification of a single DNA fragment.

In this mode the user must define either a *Forward primer region*, a *Reverse primer region*, or both. These are defined by making a selection on the sequence and right-clicking the selection.

It is also possible to define a *Region to amplify* in which case a forward- and a reverse primer region are automatically placed so as to ensure that the designated region will be included in the PCR fragment. If areas are known where primers must not bind (e.g. repeat rich areas), one or more *No primers here* regions can be defined.

If two regions are defined, it is required that at least a part of the *Forward primer region* is located upstream of the *Reverse primer region*.

After exploring the available primers (see section 21.3) and setting the desired parameter values in the Primer Parameters preference group, the **Calculate** button will activate the primer design algorithm.

21.5.1 When a single primer region is defined

If only a single region is defined, only *single primers* will be suggested by the algorithm.

After pressing the **Calculate** button a dialog will appear (see figure 21.7).

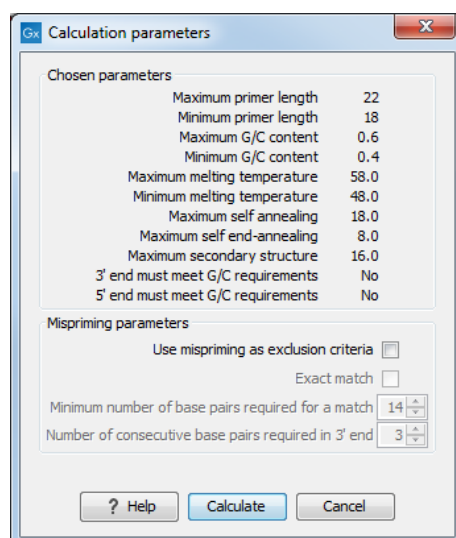


Figure 21.7: Calculation dialog for PCR primers when only a single primer region has been defined.

The top part of this dialog shows the parameter settings chosen in the Primer parameters preference group which will be used by the design algorithm.

Mispriming: The lower part contains a menu where the user can choose to include mispriming as an exclusion criteria in the design process. If this option is selected the algorithm will search for competing binding sites of the primer within the rest of the sequence, to see if the primer would match to multiple locations. If a competing site is found (according to the parameters set), the primer will be excluded.

The adjustable parameters for the search are:

- **Exact match.** Choose only to consider exact matches of the primer, i.e. all positions must base pair with the template for mispriming to occur.
- **Minimum number of base pairs required for a match.** How many nucleotides of the primer that must base pair to the sequence in order to cause mispriming.

- **Number of consecutive base pairs required in 3' end.** How many consecutive 3' end base pairs in the primer that **MUST** be present for mispriming to occur. This option is included since 3' terminal base pairs are known to be essential for priming to occur.

Note! Including a search for potential mispriming sites will prolong the search time substantially if long sequences are used as template and if the minimum number of base pairs required for a match is low. If the region to be amplified is part of a very long molecule and mispriming is a concern, consider extracting part of the sequence prior to designing primers.

21.5.2 When both forward and reverse regions are defined

If both a forward and a reverse region are defined, *primer pairs* will be suggested by the algorithm.

After pressing the **Calculate** button a dialog will appear (see figure 21.8).

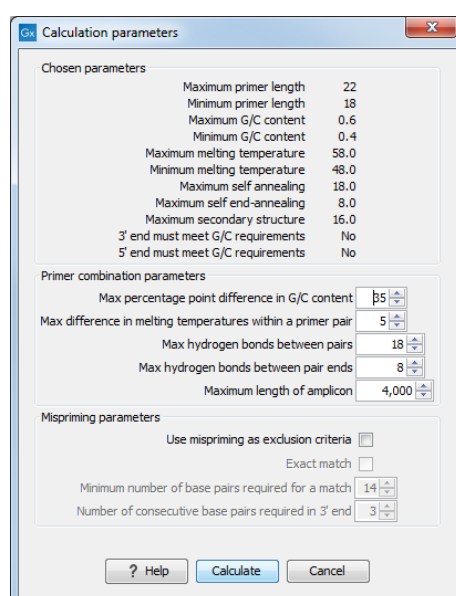


Figure 21.8: Calculation dialog for PCR primers when two primer regions have been defined.

Again, the top part of this dialog shows the parameter settings chosen in the Primer parameters preference group which will be used by the design algorithm. The lower part again contains a menu where the user can choose to include mispriming of both primers as a criteria in the design process (see section 21.5.1). The central part of the dialog contains parameters pertaining to primer pairs. Here three parameters can be set:

- Maximum percentage point difference in G/C content - if this is set at e.g. 5 points a pair of primers with 45% and 49% G/C nucleotides, respectively, will be allowed, whereas a pair of primers with 45% and 51% G/C nucleotides, respectively will not be included.
- Maximal difference in melting temperature of primers in a pair - the number of degrees Celsius that primers in a pair are all allowed to differ.
- Max hydrogen bonds between pairs - the maximum number of hydrogen bonds allowed between the forward and the reverse primer in a primer pair.

- Max hydrogen bonds between pair ends - the maximum number of hydrogen bonds allowed in the consecutive ends of the forward and the reverse primer in a primer pair.
- Maximum length of amplicon - determines the maximum length of the PCR fragment.

21.5.3 Standard PCR output table

If only a single region is selected the following columns of information are available:

- Sequence - the primer's sequence.
- Score - measures how much the properties of the primer (or primer pair) deviates from the optimal solution in terms of the chosen parameters and tolerances. The higher the score, the better the solution. The scale is from 0 to 100.
- Region - the interval of the template sequence covered by the primer
- Self annealing - the maximum self annealing score of the primer in units of hydrogen bonds
- Self annealing alignment - a visualization of the highest maximum scoring self annealing alignment
- Self end annealing - the maximum score of consecutive end base-pairings allowed between the ends of two copies of the same molecule in units of hydrogen bonds
- GC content - the fraction of G and C nucleotides in the primer
- Melting temperature of the primer-template complex
- Secondary structure score - the score of the optimal secondary DNA structure found for the primer. Secondary structures are scored by adding the number of hydrogen bonds in the structure, and 2 extra hydrogen bonds are added for each stacking base-pair in the structure
- Secondary structure - a visualization of the optimal DNA structure found for the primer

If both a forward and a reverse region are selected a table of primer pairs is shown, where the above columns (excluding the score) are represented twice, once for the forward primer (designated by the letter F) and once for the reverse primer (designated by the letter R).

Before these, and following the score of the primer pair, are the following columns pertaining to primer pair-information available:

- Pair annealing - the number of hydrogen bonds found in the optimal alignment of the forward and the reverse primer in a primer pair
- Pair annealing alignment - a visualization of the optimal alignment of the forward and the reverse primer in a primer pair.
- Pair end annealing - the maximum score of consecutive end base-pairings found between the ends of the two primers in the primer pair, in units of hydrogen bonds
- Fragment length - the length (number of nucleotides) of the PCR fragment generated by the primer pair

21.6 Nested PCR

Nested PCR is a modification of Standard PCR, aimed at reducing product contamination due to the amplification of unintended primer binding sites (mispriming). If the intended fragment can not be amplified without interference from competing binding sites, the idea is to seek out a larger outer fragment which can be unambiguously amplified and which contains the smaller intended fragment. Having amplified the outer fragment to large numbers, the PCR amplification of the inner fragment can proceed and will yield amplification of this with minimal contamination.

Primer design for nested PCR thus involves designing two primer pairs, one for the outer fragment and one for the inner fragment.

In *Nested PCR* mode the user must thus define four regions a *Forward primer region* (the outer forward primer), a *Reverse primer region* (the outer reverse primer), a *Forward inner primer region*, and a *Reverse inner primer region*. These are defined by making a selection on the sequence and right-clicking the selection. If areas are known where primers must not bind (e.g. repeat rich areas), one or more *No primers here* regions can be defined.

It is required that the *Forward primer region*, is located upstream of the *Forward inner primer region*, that the *Forward inner primer region*, is located upstream of the *Reverse inner primer region*, and that the *Reverse inner primer region*, is located upstream of the *Reverse primer region*.

In *Nested PCR* mode the *Inner melting temperature* menu in the Primer parameters panel is activated, allowing the user to set a separate melting temperature interval for the inner and outer primer pairs.

After exploring the available primers (see section 21.3) and setting the desired parameter values in the Primer parameters preference group, the **Calculate** button will activate the primer design algorithm.

After pressing the **Calculate** button a dialog will appear (see figure 21.9).

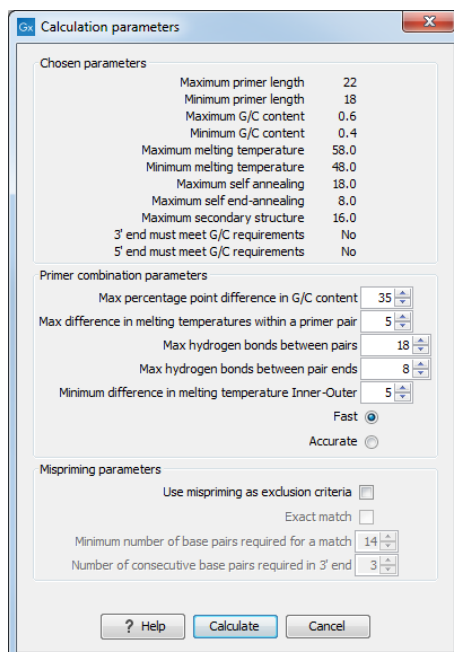


Figure 21.9: Calculation dialog for nested primers.

The top and bottom parts of this dialog are identical to the *Standard PCR* dialog for designing primer pairs described above.

The central part of the dialog contains parameters pertaining to primer pairs and the comparison between the outer and the inner pair. Here five options can be set:

- Maximum percentage point difference in G/C content (described above under *Standard PCR*) - this criteria is applied to both primer pairs independently.
- Maximal difference in melting temperature of primers in a pair - the number of degrees Celsius that primers in a pair are all allowed to differ. This criteria is applied to both primer pairs independently.
- Maximum pair annealing score - the maximum number of hydrogen bonds allowed between the forward and the reverse primer in a primer pair. This criteria is applied to all possible combinations of primers.
- Minimum difference in the melting temperature of primers in the inner and outer primer pair - all comparisons between the melting temperature of primers from the two pairs must be at least this different, otherwise the primer set is excluded. This option is applied to ensure that the inner and outer PCR reactions can be initiated at different annealing temperatures. Please note that to ensure flexibility there is no directionality indicated when setting parameters for melting temperature differences between inner and outer primer pair, i.e. it is not specified whether the inner pair should have a lower or higher T_m . Instead this is determined by the allowed temperature intervals for inner and outer primers that are set in the primer parameters preference group in the side panel. If a higher T_m of inner primers is desired, choose a T_m interval for inner primers which has higher values than the interval for outer primers.
- Two radio buttons allowing the user to choose between a fast and an accurate algorithm for primer prediction.

Nested PCR output table In nested PCR there are four primers in a solution, forward outer primer (FO), forward inner primer (FI), reverse inner primer (RI) and a reverse outer primer (RO).

The output table can show primer-pair combination parameters for all four combinations of primers and single primer parameters for all four primers in a solution (see section on *Standard PCR* for an explanation of the available primer-pair and single primer information).

The fragment length in this mode refers to the length of the PCR fragment generated by the inner primer pair, and this is also the PCR fragment which can be exported.

21.7 TaqMan

CLC Genomics Workbench allows the user to design primers and probes for TaqMan PCR applications.

TaqMan probes are oligonucleotides that contain a fluorescent reporter dye at the 5' end and a quenching dye at the 3' end. Fluorescent molecules become excited when they are irradiated and usually emit light. However, in a TaqMan probe the energy from the fluorescent dye is transferred to the quencher dye by fluorescence resonance energy transfer as long as the quencher and the

dye are located in close proximity i.e. when the probe is intact. TaqMan probes are designed to anneal within a PCR product amplified by a standard PCR primer pair. If a TaqMan probe is bound to a product template, the replication of this will cause the Taq polymerase to encounter the probe. Upon doing so, the 5' exonuclease activity of the polymerase will cleave the probe. This cleavage separates the quencher and the dye, and as a result the reporter dye starts to emit fluorescence.

The TaqMan technology is used in Real-Time quantitative PCR. Since the accumulation of fluorescence mirrors the accumulation of PCR products it can be monitored in real-time and used to quantify the amount of template initially present in the buffer.

The technology is also used to detect genetic variation such as SNP's. By designing a TaqMan probe which will specifically bind to one of two or more genetic variants it is possible to detect genetic variants by the presence or absence of fluorescence in the reaction.

A specific requirement of TaqMan probes is that a G nucleotide can not be present at the 5' end since this will quench the fluorescence of the reporter dye. It is recommended that the melting temperature of the TaqMan probe is about 10 degrees celsius higher than that of the primer pair.

Primer design for TaqMan technology involves designing a primer pair and a TaqMan probe.

In *TaqMan* the user must thus define three regions: a *Forward primer region*, a *Reverse primer region*, and a *TaqMan probe region*. The easiest way to do this is to designate a *TaqMan primer/probe region* spanning the sequence region where TaqMan amplification is desired. This will automatically add all three regions to the sequence. If more control is desired about the placing of primers and probes the *Forward primer region*, *Reverse primer region* and *TaqMan probe region* can all be defined manually. If areas are known where primers or probes must not bind (e.g. repeat rich areas), one or more *No primers here* regions can be defined. The regions are defined by making a selection on the sequence and right-clicking the selection.

It is required that at least a part of the *Forward primer region* is located upstream of the *TaqMan Probe region*, and that the *TaqMan Probe region*, is located upstream of a part of the *Reverse primer region*.

In *TaqMan* mode the *Inner melting temperature* menu in the primer parameters panel is activated allowing the user to set a separate melting temperature interval for the TaqMan probe.

After exploring the available primers (see section 21.3) and setting the desired parameter values in the Primer Parameters preference group, the **Calculate** button will activate the primer design algorithm.

After pressing the **Calculate** button a dialog will appear (see figure 21.10) which is similar to the *Nested PCR* dialog described above (see section 21.6).

In this dialog the options to set a minimum and a desired melting temperature difference between outer and inner refers to primer pair and probe respectively.

Furthermore, the central part of the dialog contains an additional parameter

- Maximum length of amplicon - determines the maximum length of the PCR fragment generated in the TaqMan analysis.

TaqMan output table In TaqMan mode there are two primers and a probe in a given solution, forward primer (F), reverse primer (R) and a TaqMan probe (TP).

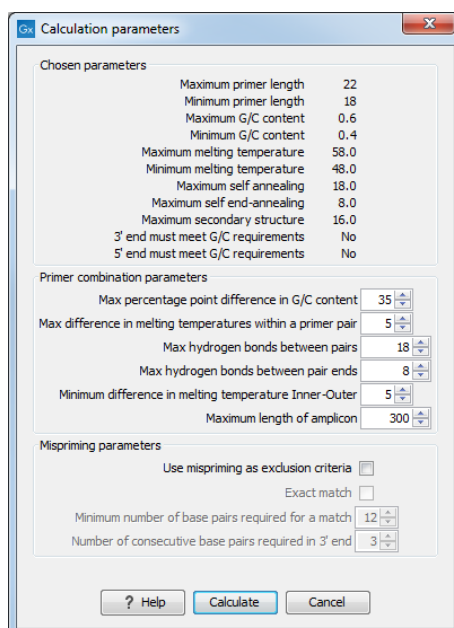


Figure 21.10: Calculation dialog for taqman primers.

The output table can show primer/probe-pair combination parameters for all three combinations of primers and single primer parameters for both primers and the TaqMan probe (see section on Standard PCR for an explanation of the available primer-pair and single primer information).

The fragment length in this mode refers to the length of the PCR fragment generated by the primer pair, and this is also the PCR fragment which can be exported.

21.8 Sequencing primers

This mode is used to design primers for DNA sequencing.

In this mode the user can define a number of *Forward primer regions* and *Reverse primer regions* where a sequencing primer can start. These are defined by making a selection on the sequence and right-clicking the selection. If areas are known where primers must not bind (e.g. repeat rich areas), one or more *No primers here* regions can be defined.

No requirements are instated on the relative position of the regions defined.

After exploring the available primers (see section 21.3) and setting the desired parameter values in the Primer Parameters preference group, the **Calculate** button will activate the primer design algorithm.

After pressing the **Calculate** button a dialog will appear (see figure 21.11).

Since design of sequencing primers does not require the consideration of interactions between primer pairs, this dialog is identical to the dialog shown in *Standard PCR* mode when only a single primer region is chosen (see section 21.5 for a description).

Sequencing primers output table In this mode primers are predicted independently for each region, but the optimal solutions are all presented in one table. The solutions are numbered consecutively according to their position on the sequence such that the forward primer region closest to the 5' end of the molecule is designated F1, the next one F2 etc.

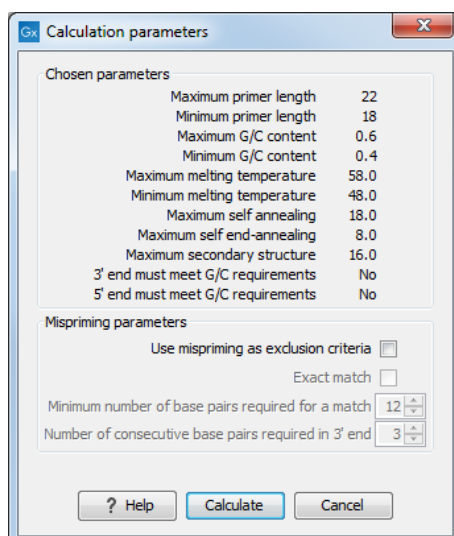


Figure 21.11: Calculation dialog for sequencing primers.

For each solution, the single primer information described under Standard PCR is available in the table.

21.9 Alignment-based primer and probe design

CLC Genomics Workbench allows the user to design PCR primers and TaqMan probes based on an alignment of multiple sequences.

The primer designer for alignments can be accessed with:

Tools | Molecular Biology Tools (🔍) | Primers and Probes (📁) | Design Primers (🧬)

Or if the alignment is already open, click Primer Designer (🧬) in the lower left part of the view.

In the alignment primer view (see figure 21.12), the basic options for viewing the template alignment are the same as for the standard view of alignments (see section 24 for an explanation of these options). This means that annotations such as known SNPs or exons can be displayed on the template sequence to guide the choice of primer regions.

21.9.1 Specific options for alignment-based primer and probe design

Compared to the primer view of a single sequence, the most notable difference is that the alignment primer view has no available graphical information. Furthermore, the selection boxes found to the left of the names in the alignment play an important role in specifying the oligo design process.

The **Primer Parameters** group in the **Side Panel** has the same options as the ones defined for primers design based on single sequences, but differs by the following submenus(see figure 21.12):

- In the **Mode** submenu, specify either:



Figure 21.12: The initial view of an alignment used for primer design.

- **Standard PCR.** Used when the objective is to design primers, or primer pairs, for PCR amplification of a single DNA fragment.
- **TaqMan.** Used when the objective is to design a primer pair and a probe set for TaqMan quantitative PCR.
- In the **Primer solution** submenu, specify requirements for the match of a PCR primer against the template sequences. These options are described further below. It contains the following options:
 - **Perfect match**
 - **Allow degeneracy**
 - **Allow mismatches**

The workflow when designing alignment based primers and probes is as follows (see figure 21.13):

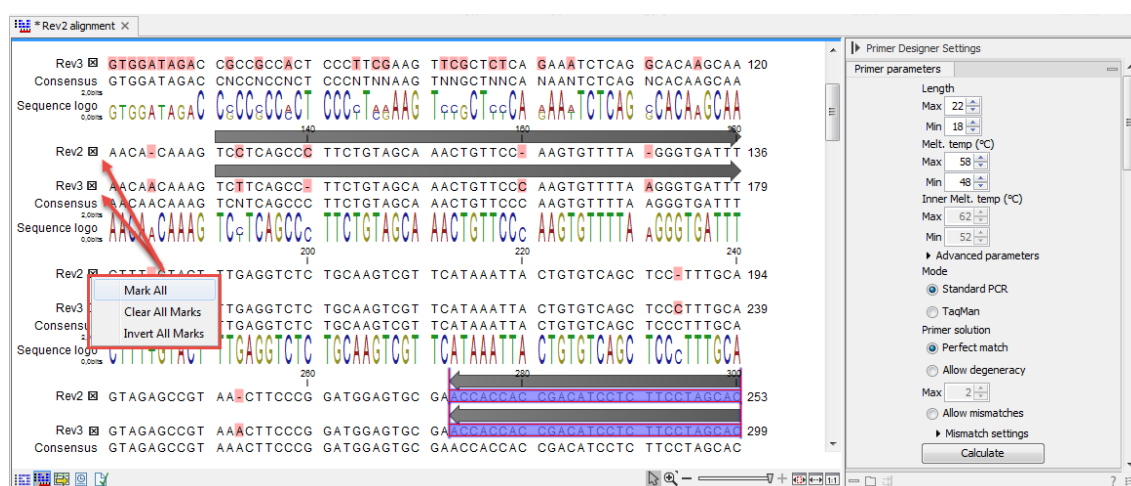


Figure 21.13: The initial view of an alignment used for primer design.

- Use selection boxes to specify groups of included and excluded sequences. To select all the sequences in the alignment, right-click one of the selection boxes and choose **Mark All**.

- Mark either a single forward primer region, a single reverse primer region or both on the sequence (and perhaps also a TaqMan region). Selections must cover all sequences in the included group. You can also specify that there should be no primers in a region (No Primers Here) or that a whole region should be amplified (Region to Amplify).
- Adjust parameters regarding single primers in the preference panel.
- Click the **Calculate** button.

21.9.2 Alignment based design of PCR primers

In this mode, a single or a pair of PCR primers are designed. *CLC Genomics Workbench* allows the user to design primers which will specifically amplify a group of *included* sequences but **not** amplify the remainder of the sequences, the *excluded* sequences. The selection boxes are used to indicate the status of a sequence, if the box is checked the sequence belongs to the included sequences, if not, it belongs to the excluded sequences. To design primers that are general for all primers in an alignment, simply add them all to the set of included sequences by checking all selection boxes. Specificity of priming is determined by criteria set by the user in the dialog box which is shown when the **Calculate** button is pressed (see below).

Different options can be chosen concerning the match of the primer to the template sequences in the included group:

- **Perfect match.** Specifies that the designed primers must have a perfect match to all relevant sequences in the alignment. When selected, primers will thus only be located in regions that are completely conserved within the sequences belonging to the included group.
- **Allow degeneracy.** Designs primers that may include ambiguity characters where heterogeneities occur in the included template sequences. The allowed fold of degeneracy is user defined and corresponds to the number of possible primer combinations formed by a degenerate primer. Thus, if a primer covers two 4-fold degenerate site and one 2-fold degenerate site the total fold of degeneracy is $4 * 4 * 2 = 32$ and the primer will, when supplied from the manufacturer, consist of a mixture of 32 different oligonucleotides. When scoring the available primers, degenerate primers are given a score which decreases with the fold of degeneracy.
- **Allow mismatches.** Designs primers which are allowed a specified number of mismatches to the included template sequences. The melting temperature algorithm employed includes the latest thermodynamic parameters for calculating T_m when single-base mismatches occur.

When in Standard PCR mode, clicking the **Calculate** button will prompt the dialog shown in figure 21.14.

The top part of this dialog shows the single-primer parameter settings chosen in the Primer parameters preference group which will be used by the design algorithm.

The central part of the dialog contains parameters pertaining to primer specificity (this is omitted if all sequences belong to the included group). Here, three parameters can be set:

- Minimum number of mismatches - the minimum number of mismatches that a primer must have against all sequences in the excluded group to ensure that it does not prime these.
- Minimum number of mismatches in 3' end - the minimum number of mismatches that a primer must have in its 3' end against all sequences in the excluded group to ensure that it does not prime these.
- Length of 3' end - the number of consecutive nucleotides to consider for mismatches in the 3' end of the primer.

The lower part of the dialog contains parameters pertaining to primer pairs (this is omitted when only designing a single primer). Here, three parameters can be set:

- Maximum percentage point difference in G/C content - if this is set at e.g. 5 points a pair of primers with 45% and 49% G/C nucleotides, respectively, will be allowed, whereas a pair of primers with 45% and 51% G/C nucleotides, respectively will not be included.
- Maximal difference in melting temperature of primers in a pair - the number of degrees Celsius that primers in a pair are all allowed to differ.
- Max hydrogen bonds between pairs - the maximum number of hydrogen bonds allowed between the forward and the reverse primer in a primer pair.
- Maximum length of amplicon - determines the maximum length of the PCR fragment.

The output of the design process is a table of single primers or primer pairs as described for primer design based on single sequences. These primers are specific to the included sequences in the alignment according to the criteria defined for specificity. The only novelty in the table, is that melting temperatures are displayed with both a maximum, a minimum and an average value to reflect that degenerate primers or primers with mismatches may have heterogeneous behavior on the different templates in the group of included sequences.

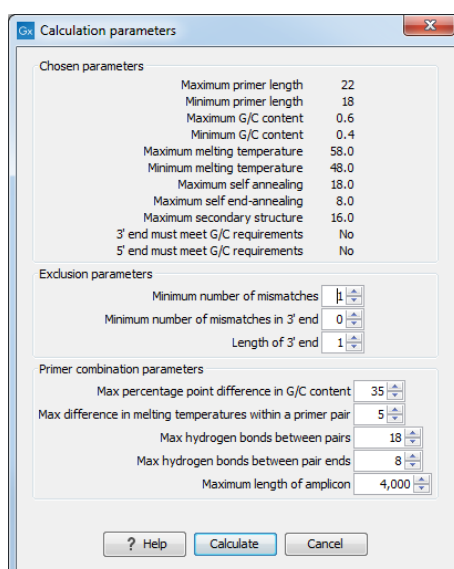


Figure 21.14: Calculation dialog shown when designing alignment based PCR primers.

21.9.3 Alignment-based TaqMan probe design

CLC Genomics Workbench allows the user to design solutions for TaqMan quantitative PCR which consist of four oligos: a general primer pair which will amplify all sequences in the alignment, a specific TaqMan probe which will match the group of *included* sequences but **not** match the *excluded* sequences and a specific TaqMan probe which will match the group of *excluded* sequences but **not** match the *included* sequences. As above, the selection boxes are used to indicate the status of a sequence, if the box is checked the sequence belongs to the included sequences, if not, it belongs to the excluded sequences. We use the terms included and excluded here to be consistent with the section above although a probe solution is presented for both groups. In TaqMan mode, primers are not allowed degeneracy or mismatches to any template sequence in the alignment, variation is only allowed/required in the TaqMan probes.

Pushing the **Calculate** button will cause the dialog shown in figure 21.15 to appear.

The top part of this dialog is identical to the *Standard PCR* dialog for designing primer pairs described above.

The central part of the dialog contains parameters to define the specificity of TaqMan probes. Two parameters can be set:

- Minimum number of mismatches - the minimum total number of mismatches that must exist between a specific TaqMan probe and all sequences which belong to the group not recognized by the probe.
- Minimum number of mismatches in central part - the minimum number of mismatches in the central part of the oligo that must exist between a specific TaqMan probe and all sequences which belong to the group not recognized by the probe.

The lower part of the dialog contains parameters pertaining to primer pairs and the comparison between the outer oligos (primers) and the inner oligos (TaqMan probes). Here, five options can be set:

- Maximum percentage point difference in G/C content (described above under Standard PCR).
- Maximal difference in melting temperature of primers in a pair - the number of degrees Celsius that primers in the primer pair are all allowed to differ.
- Maximum pair annealing score - the maximum number of hydrogen bonds allowed between the forward and the reverse primer in an oligo pair. This criteria is applied to all possible combinations of primers and probes.
- Minimum difference in the melting temperature of primer (outer) and TaqMan probe (inner) oligos - all comparisons between the melting temperature of primers and probes must be at least this different, otherwise the solution set is excluded.
- Desired temperature difference in melting temperature between outer (primers) and inner (TaqMan) oligos - the scoring function discounts solution sets which deviate greatly from this value. Regarding this, and the minimum difference option mentioned above, please note that to ensure flexibility there is no directionality indicated when setting parameters for melting temperature differences between probes and primers, i.e. it is not specified

whether the probes should have a lower or higher T_m . Instead this is determined by the allowed temperature intervals for inner and outer oligos that are set in the primer parameters preference group in the side panel. If a higher T_m of probes is required, choose a T_m interval for probes which has higher values than the interval for outer primers.

The output of the design process is a table of solution sets. Each solution set contains the following: a set of primers which are general to all sequences in the alignment, a TaqMan probe which is specific to the set of included sequences (sequences where selection boxes are checked) and a TaqMan probe which is specific to the set of excluded sequences (marked by *). Otherwise, the table is similar to that described above for TaqMan probe prediction on single sequences.

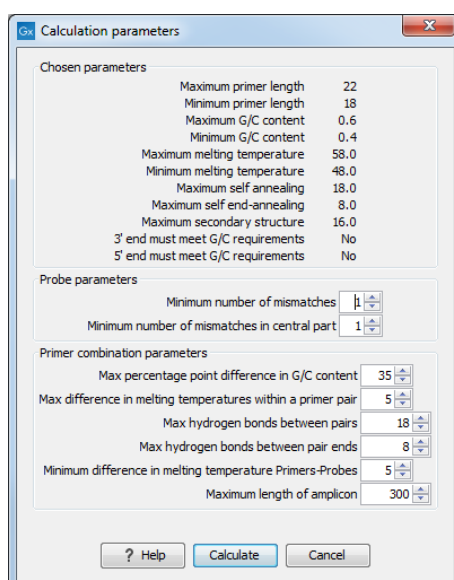


Figure 21.15: Calculation dialog shown when designing alignment based TaqMan probes.

21.10 Analyze primer properties

CLC Genomics Workbench can calculate and display the properties of predefined primers and probes:

Tools | Molecular Biology Tools (🔧) | Primers and Probes (📄) | Analyze Primer Properties (🔍)

If a sequence was selected before launching the tool, this sequence will be listed in the **Selected Elements** window of the dialog. Use the arrows to add or remove a sequence from the selected elements. (Primers are represented as DNA sequences in the Navigation Area).

Clicking **Next** generates the dialog seen in figure 21.16:

In the *Concentrations* panel a number of parameters can be specified concerning the reaction mixture and which influence melting temperatures

- **Primer concentration.** Specifies the concentration of primers and probes in units of nanomoles (nM)

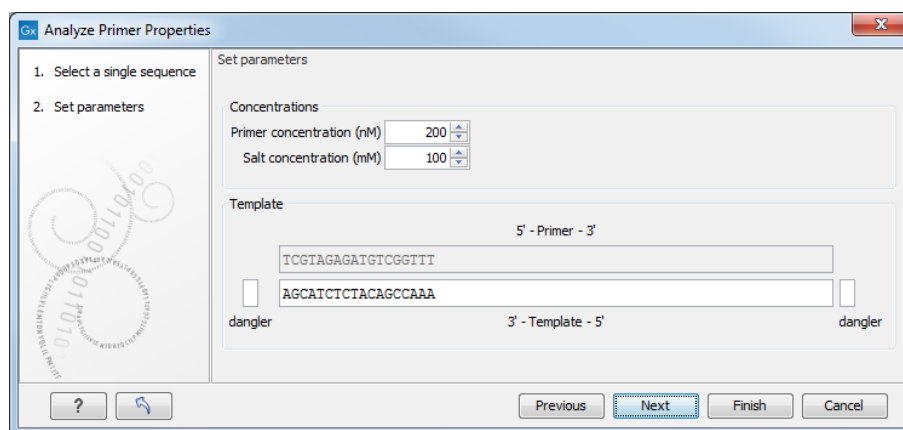


Figure 21.16: The parameters for analyzing primer properties.

- **Salt concentration.** Specifies the concentration of monovalent cations ($[Na^+]$, $[K^+]$ and equivalents) in units of millimoles (mM)

In the *Template panel* the sequences of the chosen primer and the template sequence are shown. The template sequence is as default set to the reverse complement of the primer sequence i.e. as perfectly base-pairing. However, it is possible to edit the template to introduce mismatches which may affect the melting temperature. At each side of the template sequence a text field is shown. Here, the dangling ends of the template sequence can be specified. These may have an important affect on the melting temperature [Bommarito et al., 2000]

Click **Finish** to start the tool. The result is shown in figure 21.17:

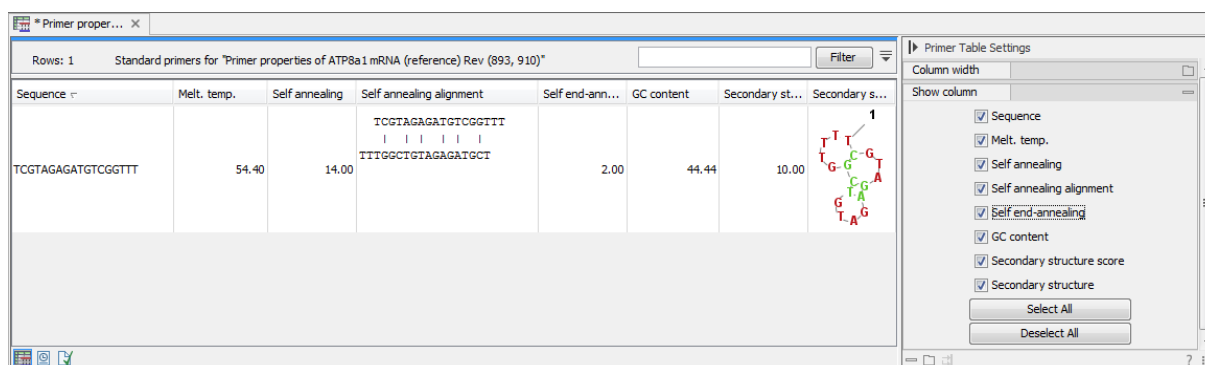


Figure 21.17: Properties of a primer.

In the **Side Panel** you can specify the information to display about the primer. The information parameters of the primer properties table are explained in section 21.5.3.

21.11 Find binding sites and create fragments

In *CLC Genomics Workbench* you have the possibility of matching known primers against one or more DNA sequences or a list of DNA sequences. This can be applied to test whether a primer used in a previous experiment is applicable to amplify a homologous region in another species, or to test for potential mispriming. This functionality can also be used to extract the resulting PCR product when two primers are matched. This is particularly useful if your primers have extensions in the 5' end. Note that this tool is not meant to analyze rapidly high-throughput data. The

maximum amount of sequences the tool will handle in a reasonable amount of time depends on your computer processing capabilities.

To search for primer binding sites:

Tools | Molecular Biology Tools (🔍) | Primers and Probes (📄) | Find Binding Sites and Create Fragments (🔗)

If a sequence was already selected in the Navigation Area, this sequence is now listed in the **Selected Elements** window of the dialog. Use the arrows to add or remove sequences or sequence lists from the selected elements.

Click **Next** when all the sequence have been added.

Note! You should not add the primer sequences at this step.

21.11.1 Binding parameters

This opens the dialog displayed in figure 21.18:

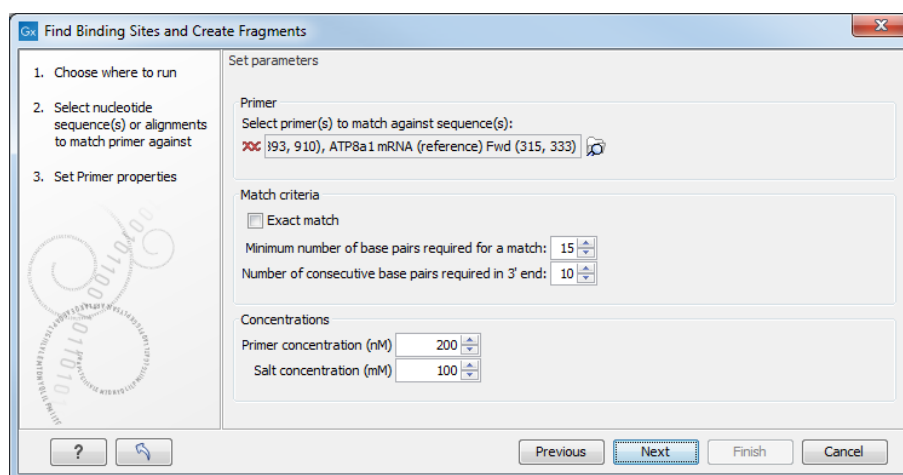


Figure 21.18: Search parameters for finding primer binding sites.

At the top, select one or more primers by clicking the browse (🔍) button. In *CLC Genomics Workbench*, primers are just DNA sequences like any other, but there is a filter on the length of the sequence. Only sequences up to 400 bp can be added.

The **Match criteria** for matching a primer to a sequence are:

- **Exact match.** Choose only to consider exact matches of the primer, i.e. all positions must base pair with the template.
- **Minimum number of base pairs required for a match.** How many nucleotides of the primer that must base pair to the sequence in order to cause priming/mispriming.
- **Number of consecutive base pairs required in 3' end.** How many consecutive 3' end base pairs in the primer that **MUST** be present for priming/mispriming to occur. This option is included since 3' terminal base pairs are known to be essential for priming to occur.

Note that the number of mismatches is reported in the output, so you will be able to filter on this afterwards (see below).

Below the match settings, you can adjust **Concentrations** concerning the reaction mixture. This is used when reporting melting temperatures for the primers.

- **Primer concentration.** Specifies the concentration of primers and probes in units of nanomoles (nM)
- **Salt concentration.** Specifies the concentration of monovalent cations ($[Na^+]$, $[K^+]$ and equivalents) in units of millimoles (mM)

21.11.2 Results - binding sites and fragments

Specify the output options as shown in figure 21.19:

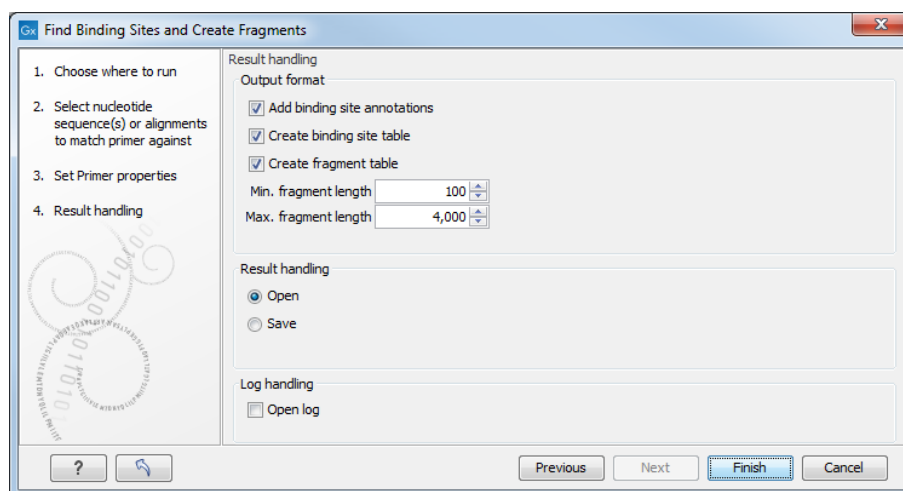


Figure 21.19: Output options include reporting of binding sites and fragments.

The output options are:

- **Add binding site annotations.** This will add annotations to the input sequences (see details below).
- **Create binding site table.** Creates a table of all binding sites. Described in details below.
- **Create fragment table.** Showing a table of all fragments that could result from using the primers. Note that you can set the minimum and maximum sizes of the fragments to be shown. The table is described in detail below.

Click **Finish** to start the tool.

An example of a **binding site annotation** is shown in figure 21.20.

The annotation has the following information:

- **Sequence of the primer.** Positions with mismatches will be in lower-case (see the fourth position in figure 21.20 where the primer has an a and the template sequence has a T).
- **Number of mismatches.**

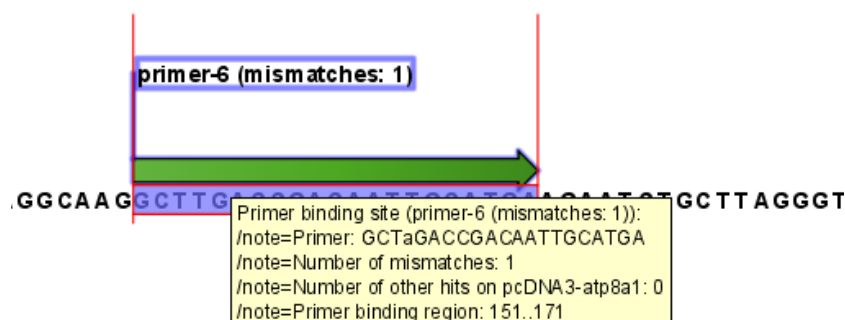


Figure 21.20: Annotation showing a primer match.

- **Number of other hits on the same sequence.** This number can be useful to check specificity of the primer.
- **Binding region.** This region ends with the 3' exact match and is simply the primer length upstream. This means that if you have 5' extensions to the primer, part of the binding region covers sequence that will actually not be annealed to the primer.

An example of the **primer binding site table** is shown in figure 21.21.

Primer name	Primer Sequence	Orientation	Region	Mismatches	Number of other hit...
Primer 1 - HindIII	aaGcttGGTGGGAGGTCTATATAA	fwd	787..810	5	0
Primer 5	ACGGTGGGAGGTCTATATAA	fwd	791..810	0	0
Primer 1	GGTGGGAGGTCTATATAA	fwd	793..810	0	0
Primer 7	GGAAGTGAGATAGAGGA	rev	complement(1373..1390)	0	0
Primer 6	AGCAGGGCAATAAGAGAA	rev	complement(1412..1430)	0	0

Figure 21.21: A table showing all binding sites.

The information here is the same as in the primer annotation and furthermore you can see additional information about melting temperature etc. by selecting the options in the **Side Panel**. See a more detailed description of this information in section 21.5.3. You can use this table to browse the binding sites. If you make a split view of the table and the sequence (see section 2.1.4), you can browse through the binding positions by clicking in the table. This will cause the sequence view to jump to the position of the binding site.

An example of a **fragment table** is shown in figure 21.22.

The table first lists the names of the forward and reverse primers, then the length of the fragment and the region. The last column tells if there are other possible fragments fulfilling the length criteria on this sequence. This information can be used to check for competing products in the PCR. In the **Side Panel** you can show information about melting temperature for the primers as well as the difference between melting temperatures.

You can use this table to browse the fragment regions. If you make a split view of the table and

Fwd. name	Rev. name	Fragment length	Region	Other fragments
Primer 1 - HindIII	Primer 7	604	787..1390	0
Primer 5	Primer 7	600	791..1390	0
Primer 1	Primer 7	598	793..1390	0
Primer 1 - HindIII	Primer 6	644	787..1430	0
Primer 5	Primer 6	640	791..1430	0
Primer 1	Primer 6	638	793..1430	0

Figure 21.22: A table showing all possible fragments of the specified size.

the sequence (see section 2.1.4), you can browse through the fragment regions by clicking in the table. This will cause the sequence view to jump to the start position of the fragment.

There are some additional options in the fragment table. First, you can annotate the fragment on the original sequence. This is done by right-clicking (Ctrl-click on Mac) the fragment and choose **Annotate Fragment** as shown in figure 21.23.

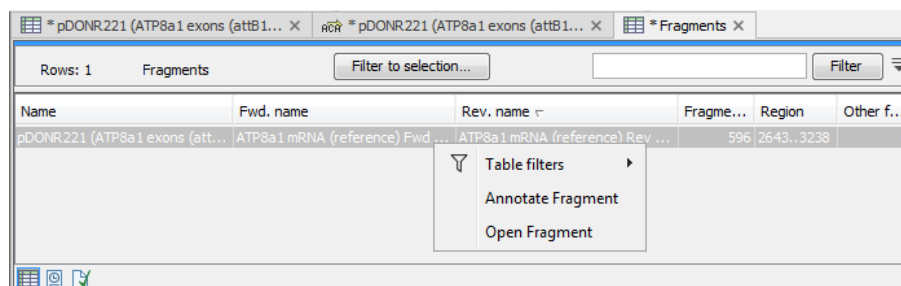


Figure 21.23: Right-clicking a fragment allows you to annotate the region on the input sequence or open the fragment as a new sequence.

This will put a *PCR fragment* annotations on the input sequence covering the region specified in the table. As you can see from figure 21.23, you can also choose to **Open Fragment**. This will create a new sequence representing the PCR product that would be the result of using these two primers. Note that if you have extensions on the primers, they will be used to construct the new sequence.

If you are doing restriction cloning using primers with restriction site extensions, you can use this functionality to retrieve the PCR fragment for us in the cloning editor (see section 23.3).

21.12 Order primers

To facilitate the ordering of primers and probes, *CLC Genomics Workbench* offers an easy way of displaying and saving a textual representation of one or more primers:

Tools | Molecular Biology Tools (🔍) | Primers and Probes (📄) | Order Primers (📋)

This opens a dialog where you can choose primers to generate a textual representation of the

primers (see figure 21.24).

The first line states the number of primers being ordered and after this follows the names and nucleotide sequences of the primers in 5'-3' orientation. From the editor, the primer information can be copied and pasted to web forms or e-mails. This file can also be saved and exported as a text file.

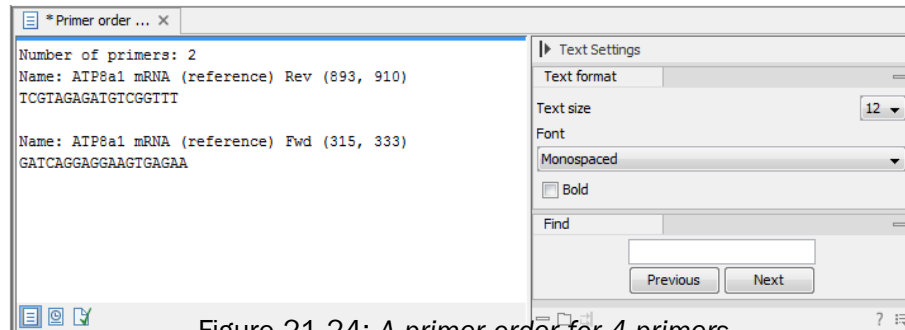


Figure 21.24: A primer order for 4 primers.

Chapter 22

Sequencing data analyses

Contents

22.1 Importing and viewing trace data	558
22.1.1 Trace settings in the Side Panel	558
22.2 Trim sequences	559
22.2.1 Trimming using the Trim Sequences tool	559
22.2.2 Manual trimming	562
22.3 Assemble sequences	562
22.4 Assemble sequences to reference	564
22.5 Sort sequences by name	567
22.6 Add sequences to an existing contig	570
22.7 View and edit contigs and read mappings	571
22.7.1 View settings in the Side Panel	571
22.7.2 Editing a contig or read mapping	576
22.7.3 Sorting reads	576
22.7.4 Read conflicts	577
22.7.5 Using the mapping	577
22.7.6 Extracting reads from mappings	578
22.7.7 Variance table	579
22.8 Reassemble contig	580
22.9 Secondary peak calling	581
22.10 Extract Consensus Sequence	583

This chapter explains the features in *CLC Genomics Workbench* for handling data analysis of low-throughput conventional Sanger sequencing data. For analysis of high-throughput sequencing data, please refer to part [IV](#).

This chapter first explains how to trim sequence reads. Next follows a description of how to assemble reads into contigs both with and without a reference sequence. In the final section, the options for viewing and editing contigs are explained.

22.1 Importing and viewing trace data

A number of different binary trace data formats can be imported into the program, including *Standard Chromatogram Format* (.SCF), *ABI sequencer data files* (.ABI and .AB1), *PHRED output files* (.PHD) and *PHRAP output files* (.ACE) (see section 7.1).

After import, the sequence reads and their trace data are saved as DNA sequences. This means that all analyses that apply to DNA sequences can be performed on the sequence reads.

You can see additional information about the quality of the traces by holding the mouse cursor on the imported sequence. This will display a tool tip as shown in figure 22.1.

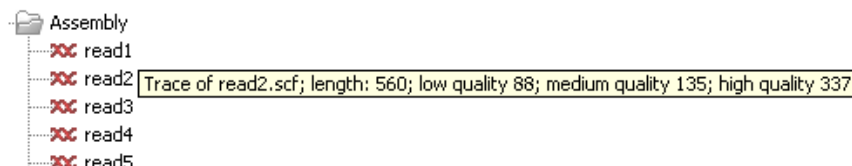


Figure 22.1: A tooltip displaying information about the quality of the chromatogram.

The qualities are based on the phred scoring system, with scores below 19 counted as low quality, scores between 20 and 39 counted as medium quality, and those 40 and above counted as high quality.

If the trace file does not contain information about quality, only the sequence length will be shown.

To view the trace data, open the sequence read in a standard sequence view (ACT).

The traces can be scaled by dragging the trace vertically as shown in figure 22.2. The Workbench automatically adjust the height of the traces to be readable, but if the trace height varies a lot, this manual scaling is very useful.

The height of the area available for showing traces can be adjusted in the **Side Panel** as described in section 22.1.1.

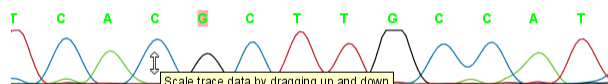


Figure 22.2: Grab the traces to scale.

22.1.1 Trace settings in the Side Panel

In the Nucleotide info preference group the display of trace data can be selected and unselected. When selected, the trace data information is shown as a plot beneath the sequence. The appearance of the plot can be adjusted using the following options (see figure 22.3):

- **Nucleotide trace.** For each of the four nucleotides the trace data can be selected and unselected.
- **Scale traces.** A slider which allows the user to scale the height of the trace area. Scaling the traces individually is described in section 22.1.

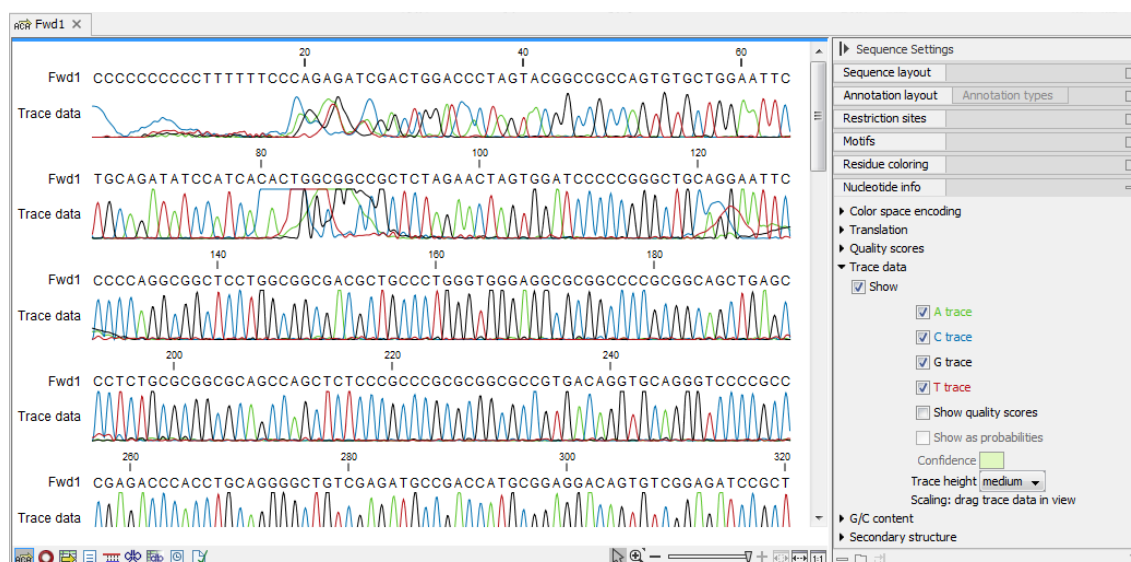


Figure 22.3: A sequence with trace data. The preferences for viewing the trace are shown in the Side Panel.

When working with stand-alone mappings containing reads with trace data, you can view the traces by turning on the trace setting options as described here **and** choosing **Not compact** in the Read layout setting for the mapping.

Please see section 30.3.3.

22.2 Trim sequences

Trimming as described in this section involves marking of low quality and/or vector sequence with a Trim annotation as shown in figure 22.4). Such annotated regions are then ignored when using downstream analysis tools located in the same area in the Tools menu, for example Assembly (see section 22.3). The trimming described here annotates, but does not remove data, allowing you to explore the output of different trimming schemes easily.

Trimming as a separate task can be done manually or using a tool designed specifically for this task.

To remove existing trimming information from a sequence, simply remove its trim annotation (see section 15.3.2). When exporting to fasta format, there is an option to remove sequence ends covered by Trim annotations.

Note! To directly remove trimmed regions rather than adding a Trim annotation, use the **Trim Reads** tool (see section 28.2).

22.2.1 Trimming using the Trim Sequences tool

Sequence reads can be trimmed based on a number of different criteria. Using a trimming tool for this is particularly useful if:

- You have many sequences to trim.
- You wish to trim vector contamination from sequencing reads.

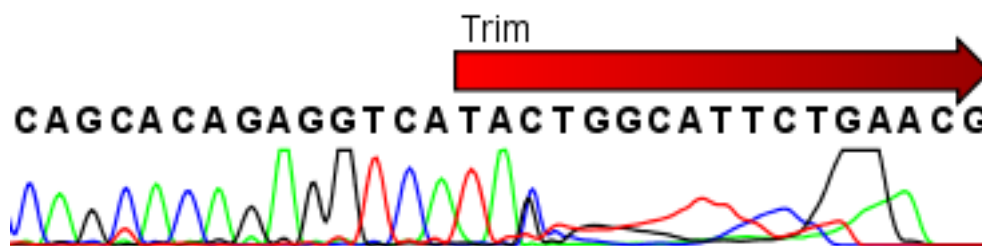


Figure 22.4: *Trimming creates annotations on the regions that will be ignored in the assembly process.*

- You wish to ensure that consistency when trimming. That is, you wish to ensure the same criteria are used for all the sequences in a set.

To start up the Trim Sequences tool in the Workbench, go to the menu option:

Tools | Molecular Biology Tools (🔧) | Sanger Sequencing Analysis (📊) | Trim Sequences (✂️)

This opens a dialog where you can choose the sequences to trim, by using the arrows to move them between the Navigation Area and the 'Selected Elements' box.

You can then specify the trim parameters as displayed in figure 22.5.

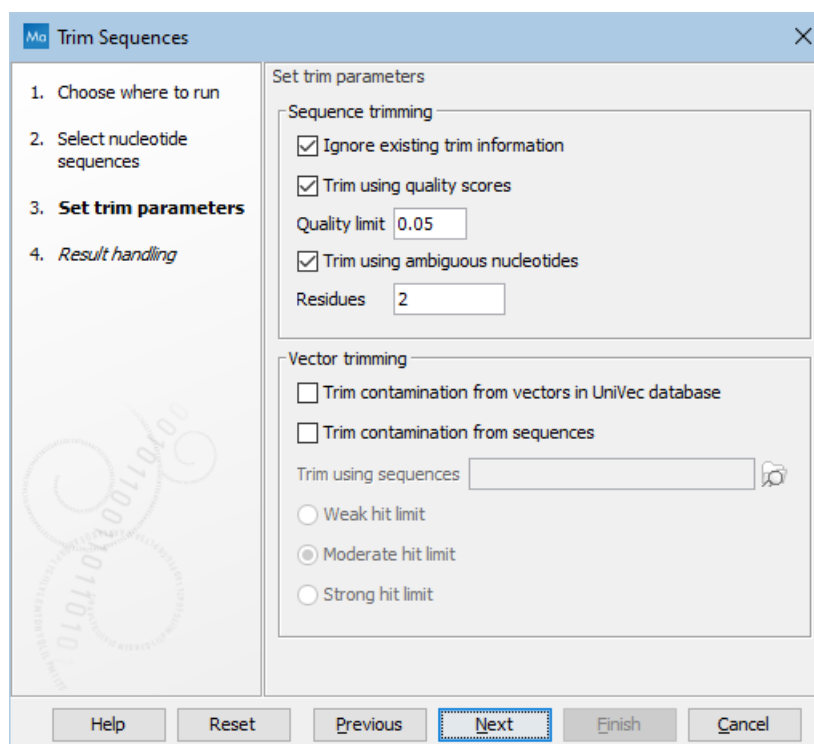


Figure 22.5: *Setting parameters for trimming.*

The following parameters can be adjusted in the dialog:

- **Ignore existing trim information.** If you have previously trimmed the sequences, you can

check this to remove existing trimming annotation prior to analysis.

- **Trim using quality scores.** If the sequence files contain quality scores from a base caller algorithm this information can be used for trimming sequence ends. The program uses the modified-Mott trimming algorithm for this purpose (Richard Mott, personal communication):

Quality scores in the Workbench are on a Phred scale, and formats using other scales will be converted during import. The Phred quality scores (Q), defined as: $Q = -10\log_{10}(P)$, where P is the base-calling error probability, can then be used to calculate the error probabilities, which in turn can be used to set the limit for, which bases should be trimmed.

Hence, the first step in the trim process is to convert the quality score (Q) to an error probability: $p_{\text{error}} = 10^{-\frac{Q}{10}}$. (This now means that low values are high quality bases.)

Next, for every base a new value is calculated: $\text{Limit} - p_{\text{error}}$. This value will be negative for low quality bases, where the error probability is high.

For every base, the Workbench calculates the running sum of this value. If the sum drops below zero, it is set to zero. The part of the sequence not trimmed will be the region ending at the highest value of the running sum and starting at the last zero value before this highest score. Everything before and after this region will be trimmed. A read will be completely removed if the score never makes it above zero.

At <https://resources.qiagenbioinformatics.com/testdata/trim.zip> you find an example sequence and an Excel sheet showing the calculations done for this particular sequence to illustrate the procedure described above.

- **Trim ambiguous nucleotides.** This option trims the sequence ends based on the presence of ambiguous nucleotides (typically N). Note that the automated sequencer generating the data must be set to output ambiguous nucleotides in order for this option to apply. The algorithm takes as input the *maximal number of ambiguous nucleotides allowed in the sequence after trimming*. If this maximum is set to e.g. 3, the algorithm finds the maximum length region containing 3 or fewer ambiguities and then trims away the ends not included in this region. The "Trim ambiguous nucleotides" option trims all types of ambiguous nucleotides (see Appendix H).

- **Trim contamination from vectors in UniVec database.** If selected, the program will match the sequence reads against all vectors in the UniVec database and mark sequence ends with significant matches with a 'Trim' annotation.

The UniVec database build 10.1 is included when you install the *CLC Genomics Workbench*. A list of all the vectors in the database can be found at <https://www.ncbi.nlm.nih.gov/VecScreen/replist.html>.

- **Trim contamination from sequences.** This option lets you use your own vector sequences that you have imported into the *CLC Genomics Workbench*. If selected, **Trim using sequences** will be enabled and you can choose one or more sequences.
- **Hit limit for vector trimming.** When at least one vector trimming parameter is selected, the strictness for vector contamination trimming can be specified. Since vector contamination usually occurs at the beginning or end of a sequence, different criteria are applied for terminal and internal matches. A match is considered terminal if it is located within the first 25 bases at either sequence end. Three match categories are defined according to the expected frequency of an alignment with the same score occurring between random

sequences. The *CLC Genomics Workbench* uses the same settings as VecScreen (<https://www.ncbi.nlm.nih.gov/tools/vecscreen/>):

- **Weak hit limit** Expect 1 random match in 40 queries of length 350 kb.
 - * Terminal match with Score 16 to 18.
 - * Internal match with Score 23 to 24.
- **Moderate hit limit** Expect 1 random match in 1,000 queries of length 350 kb.
 - * Terminal match with Score 19 to 23.
 - * Internal match with Score 25 to 29.
- **Strong hit limit** Expect 1 random match in 1,000,000 queries of length 350 kb.
 - * Terminal match with Score at least 24.
 - * Internal match with Score at least 30.

In the last step of the wizard, you can choose to create a report, summarizing how each sequence has been trimmed. Click **Finish** to start the tool. This will start the trimming process. Views of each trimmed sequence will be shown, and you can inspect the result by looking at the "Trim" annotations (they are colored red as default). Note that the trim annotations are used to signal that this part of the sequence is to be ignored during further analyses, hence the trimmed sequences are not deleted. If there are no trim annotations, the sequence has not been trimmed.

22.2.2 Manual trimming

Sequence reads can be trimmed manually while inspecting their trace and quality data.

Trimming sequences manually involves adding an annotation of type Trim, with the special condition that this annotation can only be applied to the ends of a sequence:



double-click the sequence to trim in the Navigation Area | select the region you want to trim | right-click the selection | Trim sequence left/right to determine the direction of the trimming

This will add a trimming annotation to the end of the sequence in the selected direction. No sequence is being deleted here. Rather, the regions covered by trim annotations are noted by downstream analyses (in the same section of the Tools menu as the Trim Sequences tool) as regions to be ignored.

22.3 Assemble sequences

This section describes how to assemble a number of sequence reads into a contig without the use of a reference sequence (a known sequence that can be used for comparison with the other sequences, see section 22.4).

Note! You can assemble a maximum of 10,000 sequences at a time.

To assemble more sequences, please use the **De Novo Assembly** () tool under **De Novo Sequencing** () under the **Tools** menu instead.

To perform the assembly:

Tools | Molecular Biology Tools () | **Sanger Sequencing Analysis** () | **Assemble Sequences** ()

This will open a dialog where you can select sequences to assemble. If you already selected sequences in the Navigation Area, these will be shown in 'Selected Elements'. You can alter your choice of sequences to assemble, or add others, by using the arrows to move sequences between the Navigation Area and the 'Selected Elements' box. You can also add sequence lists.

When the sequences are selected, click **Next**. This will show the dialog in figure 22.6

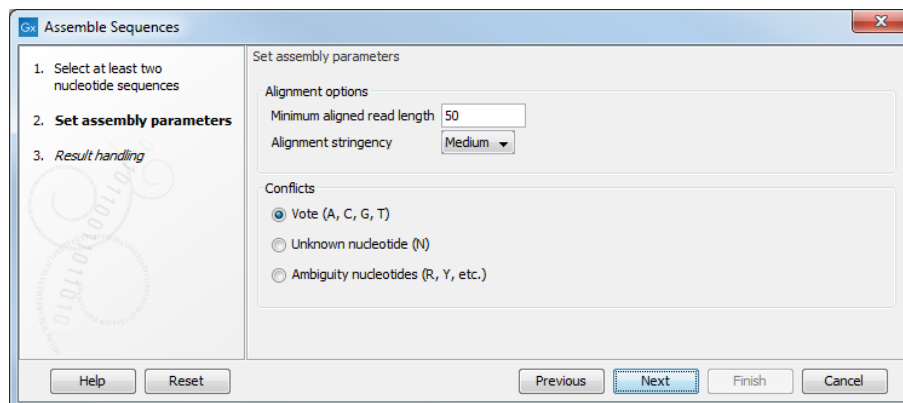



Figure 22.6: Setting assembly parameters.

This dialog gives you the following options for assembly:

- **Minimum aligned read length.** The minimum number of nucleotides in a read which must be successfully aligned to the contig. If this criteria is not met by a read, the read is excluded from the assembly.
- **Alignment stringency.** Specifies the stringency (Low, Medium or High) of the scoring function used by the alignment step in the contig assembly algorithm. A higher stringency level will tend to produce contigs with fewer ambiguities but will also tend to omit more sequencing reads and to generate more and shorter contigs.
- **Conflicts.** If there is a conflict, i.e. a position where there is disagreement about the residue (A, C, T or G), you can specify how the contig sequence should reflect the conflict:
 - **Vote (A, C, G, T).** The conflict will be solved by counting instances of each nucleotide and then letting the majority decide the nucleotide in the contig. In case of equality, ACGT are given priority over one another in the stated order.
 - **Unknown nucleotide (N).** The contig will be assigned an 'N' character in all positions with conflicts (conflicts are registered already when two nucleotides differ).
 - **Ambiguity nucleotides (R, Y, etc.).** The contig will display an ambiguity nucleotide reflecting the different nucleotides found in the reads (nucleotide ambiguity is registered already when two nucleotides differ). For an overview of ambiguity codes, see Appendix H.

Note, that conflicts will always be highlighted no matter which of the options you choose. Furthermore, each conflict will be marked as annotation on the contig sequence and will be present if the contig sequence is extracted for further analysis. As a result, the details of any experimental heterogeneity can be maintained and used when the result of single-sequence analyzes is interpreted. Read more about conflicts in section 22.7.4.

- **Create full contigs, including trace data.** This will create a contig where all the aligned reads are displayed below the contig sequence. (You can always extract the contig sequence without the reads later on.) For more information on how to use the contigs that are created, see section 22.7.
- **Show tabular view of contigs.** A contig can be shown both in a graphical as well as a tabular view. If you select this option, a tabular view of the contig will also be opened (Even if you do not select this option, you can show the tabular view of the contig later on by clicking **Table**  at the bottom of the view.) For more information about the tabular view of contigs, see section 22.7.7.
- **Create only consensus sequences.** This will not display a contig but will only output the assembled contig sequences as single nucleotide sequences. If you choose this option it is not possible to validate the assembly process and edit the contig based on the traces.

When the assembly process has ended, a number of views will be shown, each containing a contig of two or more sequences that have been matched. If the number of contigs seem too high or low, try again with another **Alignment stringency** setting. Depending on your choices of output options above, the views will include trace files or only contig sequences. However, the calculation of the contig is carried out the same way, no matter how the contig is displayed.

See section 22.7 on how to use the resulting contigs.

22.4 Assemble sequences to reference

This section describes how to assemble a number of sequence reads into a contig using a reference sequence, a process called read mapping. A reference sequence can be particularly helpful when the objective is to characterize SNP variation in the data.

Note! You can assemble a maximum of 10,000 sequences at a time.

To assemble a larger number of sequences, please use **Map Reads to Reference** , which is in the **Resequencing Analysis** folder under the **Tools** menu.


To run the **Assemble Sequences to Reference** tool, go to:

Tools | Molecular Biology Tools  **| Sanger Sequencing Analysis**  **| Assemble Sequences to Reference** 

This opens a dialog where you can alter your choice of sequences to assemble. If you have already selected sequences in the Navigation Area, these will be shown in Selected Elements, however you can remove these or add others, by using the arrows to move sequences between the Navigation Area and Selected Elements boxes. You can also add sequence lists.

When the sequences are selected, click **Next**, and you will see the dialog shown in figure 22.7

This dialog gives you the following options for assembling:

- **Reference sequence.** Click the **Browse and select element** icon  in order to select one or more sequences to use as reference(s).
- **Include reference sequence(s) in contig(s).** This will create a contig for each reference with the corresponding reference sequence at the top and the aligned sequences below. This

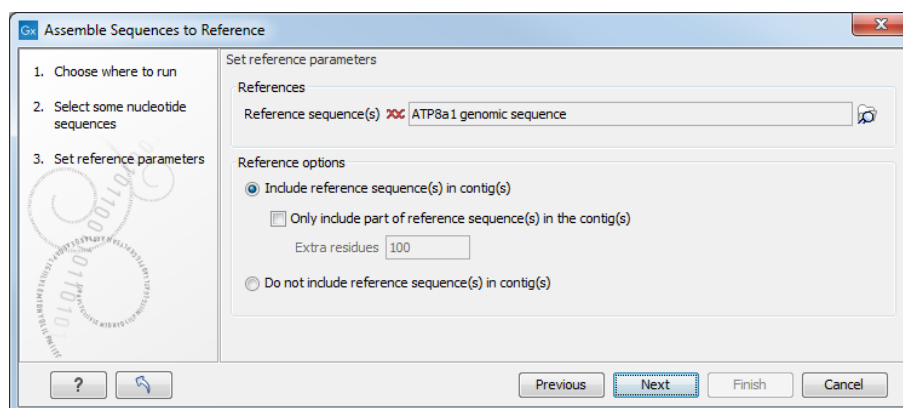


Figure 22.7: Parameters for how the reference should be handled when assembling sequences to a reference sequence.

option is useful when comparing sequence reads to a closely related reference sequence e.g. when sequencing for SNP characterization.

- **Only include part of reference sequence(s) in the contig(s).** If the aligned sequences only cover a small part of a reference sequence, it may not be desirable to include the whole reference sequence in a contig. When this option is selected, you can specify the number of residues from reference sequences that should be included on each side of regions spanned by aligned sequences using the **Extra residues** field.
- **Do not include reference sequence(s) in contig(s).** This will produce contigs without any reference sequence where the input sequences have been assembled using reference sequences as a scaffold. The input sequences are first aligned to the reference sequence(s). Next, the consensus sequence for regions spanned by aligned sequences are extracted and output as contigs. This option is useful when performing assembling sequences where the reference sequences that are not closely related to the input sequencing.

When the reference sequence has been selected, click **Next**, to see the dialog shown in figure 22.8

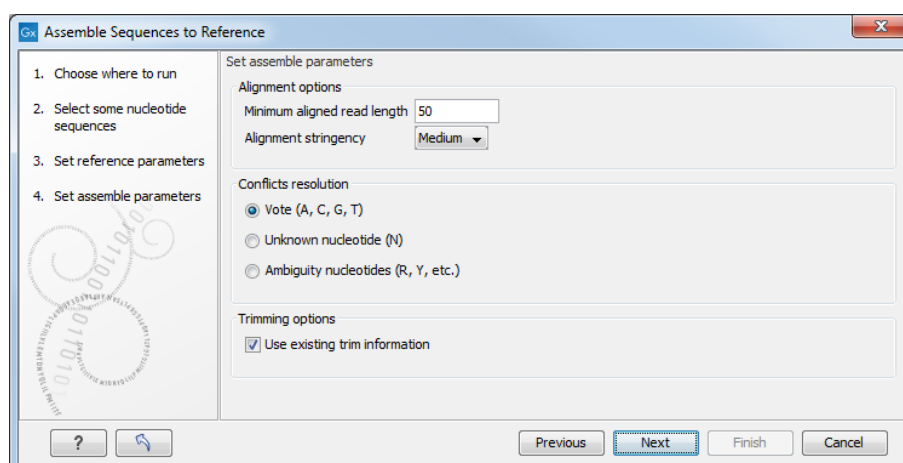


Figure 22.8: Options for how the input sequences should be aligned and how nucleotide conflicts should be handled.

In this dialog, you can specify the following options:

- **Minimum aligned read length.** The minimum number of nucleotides in a read which must match a reference sequence. If an input sequence does not meet this criteria, the sequence is excluded from the assembly.
- **Alignment stringency.** Specifies the stringency (Low, Medium or High) of the scoring function used for aligning the input sequences to the reference sequence(s). A higher stringency level often produce contigs with lower levels of ambiguity but also reduces the ability to align distant homologs or sequences with a high error rate to reference sequences. The result of a higher stringency level is often that the number of contigs increases and the average length of contigs decreases while the quality of each contig increases.

The stringency settings Low, Medium and High are based on the following score values (mt=match, ti=transition, tv=transversion, un=unknown):

Score values			
	Low	Medium	High
Match (mt)	2	2	2
Transversion (tv)	-6	-10	-20
Transition (ti)	-2	-6	-16
Unknown (un)	-2	-6	-16
Gap	-8	-16	-36

Score Matrix					
	A	C	G	T	N
A	mt	tv	ti	tv	un
C	tv	mt	tv	ti	un
G	ti	tv	mt	tv	un
T	tv	ti	tv	mt	un
N	un	un	un	un	un

- **Conflicts resolution.** If there is a conflict, i.e. a position where aligned sequences disagreement about the residue (A, C, T or G), you can specify how the contig sequence should reflect this conflict:
 - **Unknown nucleotide (N).** The contig will be assigned an 'N' character in all positions with conflicts (conflicts are registered already when two nucleotides differ).
 - **Ambiguity nucleotides (R, Y, etc.).** The contig will display an ambiguity nucleotide reflecting the different nucleotides found in the aligned sequences (nucleotide ambiguity is registered when two nucleotides differ). For an overview of ambiguity codes, see Appendix H.
 - **Vote (A, C, G, T).** The conflict will be solved by counting instances of each nucleotide and then letting the majority decide the nucleotide in the contig. In case of equality, ACGT are given priority over one another in the stated order.

Note, that conflicts will be highlighted for all options. Furthermore, conflicts will be marked with an annotation on each contig sequence which are preserved if the contig sequence is extracted for further analysis. As a result, the details of any experimental heterogeneity can be maintained and used when the result of single-sequence analyzes is interpreted.

- **Trimming options.** When aligning sequences to a reference sequence, trimming is generally not necessary, but if you wish to use trimming you can check this box. It requires that the sequence reads have been trimmed beforehand (see section 22.2 for more information about trimming).

Click **Finish** to start the tool. This will start the assembly process. See section 22.7 on how to use the resulting contigs.

22.5 Sort sequences by name

With this functionality you will be able to group sequencing reads based on their file name. A typical example would be that you have a list of files named like this:

```
...  
A02__Asp_F_016_2007-01-10  
A02__Asp_R_016_2007-01-10  
A02__Gln_F_016_2007-01-11  
A02__Gln_R_016_2007-01-11  
A03__Asp_F_031_2007-01-10  
A03__Asp_R_031_2007-01-10  
A03__Gln_F_031_2007-01-11  
A03__Gln_R_031_2007-01-11  
...
```

In this example, the names have five distinct parts (we take the first name as an example):

- **A02** which is the position on the 96-well plate
- **Asp** which is the name of the gene being sequenced
- **F** which describes the orientation of the read (forward/reverse)
- **016** which is an ID identifying the sample
- **2007-01-10** which is the date of the sequencing run

To start mapping these data, you probably want to have them divided into groups instead of having all reads in one folder. If, for example, you wish to map each sample separately, or if you wish to map each gene separately, you cannot simply run the mapping on all the sequences in one step.

That is where **Sort Sequences by Name** comes into play. It will allow you to specify which part of the name should be used to divide the sequences into groups. We will use the example described above to show how it works:

Tools | Molecular Biology Tools  | **Sanger Sequencing Analysis**  | **Sort Sequences by Name** 

This opens a dialog where you can add the sequences you wish to sort, by using the arrows to move them between the Navigation Area and 'Selected Elements'. You can also add sequence

lists or the contents of an entire folder by right-clicking the folder and choose: **Add folder contents**.

When you click **Next**, you will be able to specify the details of how the grouping should be performed. First, you have to choose how each part of the name should be identified. There are three options:

- **Simple**. This will simply use a designated character to split up the name. You can choose a character from the list:
 - Underscore _
 - Dash -
 - Hash (number sign / pound sign) #
 - Pipe |
 - Tilde ~
 - Dot .
- **Positions**. You can define a part of the name by entering the start and end positions, e.g. from character number 6 to 14. For this to work, the names have to be of equal lengths.
- **Java regular expression**. This is an option for advanced users where you can use a special syntax to have total control over the splitting. See more below.

In the example above, it would be sufficient to use a simple split with the underscore _ character, since this is how the different parts of the name are divided.

When you have chosen a way to divide the name, the parts of the name will be listed in the table at the bottom of the dialog. There is a checkbox next to each part of the name. This checkbox is used to specify which of the name parts should be used for grouping. In the example above, if we want to group the reads according to date and analysis position, these two parts should be checked as shown in figure 22.9.

At the middle of the dialog there is a preview panel listing:

- **Sequence name**. This is the name of the first sequence that has been chosen. It is shown here in the dialog in order to give you a sample of what the names in the list look like.
- **Resulting group**. The name of the group that this sequence would belong to if you proceed with the current settings.
- **Number of sequences**. The number of sequences chosen in the first step.
- **Number of groups**. The number of groups that would be produced when you proceed with the current settings.

This preview cannot be changed. It is shown to guide you when finding the appropriate settings.

Click **Finish** to start the tool. A new sequence list will be generated for each group. It will be named according to the group, e.g. 2004-08-24_A02 will be the name of one of the groups in the example shown in figure 22.9.

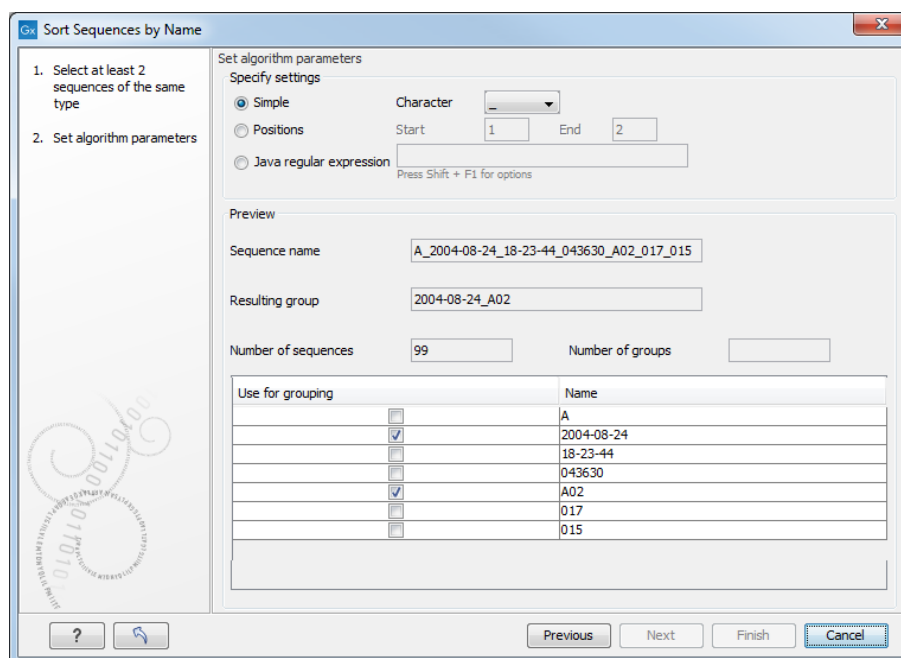


Figure 22.9: Splitting up the name at every underscore (`_`) and using the date and analysis position for grouping.

Advanced splitting using regular expressions

You can see a more detail explanation of the regular expressions syntax in section [18.9.3](#).

In this section you will see a practical example showing how to create a regular expression. Consider a list of files as shown below:

```
...
adk-29_adk1n-F
adk-29_adk2n-R
adk-3_adk1n-F
adk-3_adk2n-R
adk-66_adk1n-F
adk-66_adk2n-R
atp-29_atpA1n-F
atp-29_atpA2n-R
atp-3_atpA1n-F
atp-3_atpA2n-R
atp-66_atpA1n-F
atp-66_atpA2n-R
...
```

In this example, we wish to group the sequences into three groups based on the number after the `-` and before the `_` (i.e. 29, 3 and 66). The simple splitting as shown in figure [22.9](#) requires the same character before and after the text used for grouping, and since we now have both a `-` and a `_`, we need to use the regular expressions instead (note that dividing by position would not work because we have both single and double digit numbers (3, 29 and 66)).

The regular expression for doing this would be `(.*)-(.*)_(.*)` as shown in figure [22.10](#).

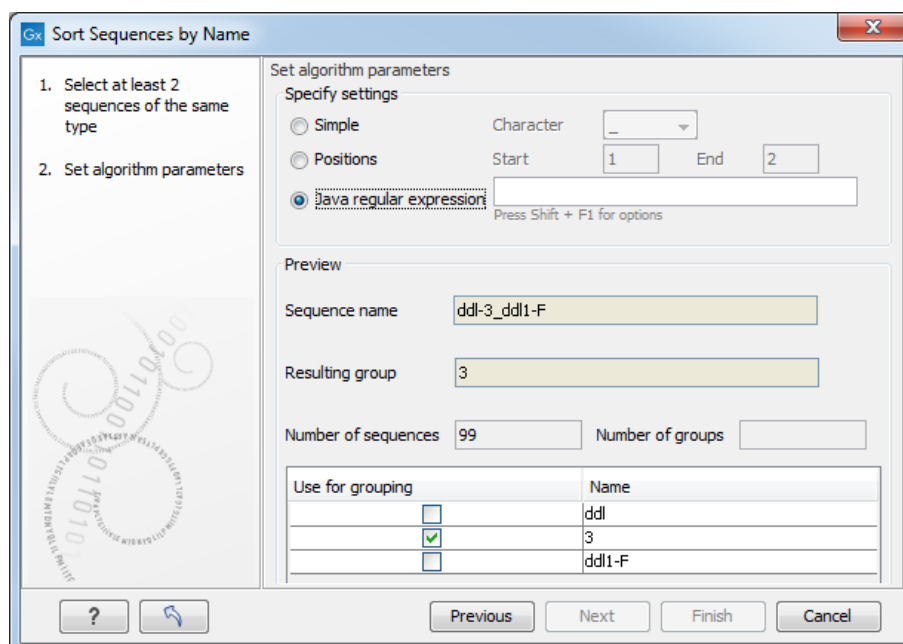


Figure 22.10: Dividing the sequence into three groups based on the number in the middle of the name.




The round brackets () denote the part of the name that will be listed in the groups table at the bottom of the dialog. In this example we actually did not need the first and last set of brackets, so the expression could also have been `. *- (. *) _ . *` in which case only one group would be listed in the table at the bottom of the dialog.

22.6 Add sequences to an existing contig

This section describes how to assemble sequences to an existing contig. This feature can be used for example to provide a steady work-flow when a number of exons from the same gene are sequenced one at a time and assembled to a reference sequence.

Note that the new sequences will be added to the existing contig, which will not be extended. If the new sequences extend beyond the existing contig, they will be cut off.

To run the **Add Sequences to Contig** tool, go to:

Tools | Molecular Biology Tools () | Sanger Sequencing Analysis () | Add Sequences to Contig ()

or

or **Right-click in the empty white area of the contig | Add Sequences to Contig ()**

This opens a dialog where you can select one contig and a number of sequences to assemble. If you have already selected sequences in the Navigation Area, these will be shown in the 'Selected Elements' box. However, you can remove these, or add others, by using the arrows to move sequences between the Navigation Area and Selected Elements boxes. You can also add sequence lists.

Often, the results of the assembly will be better if the sequences are trimmed first (see section 22.2.1).

When the elements are selected, click **Next**, and you will see the dialog shown in figure 22.11

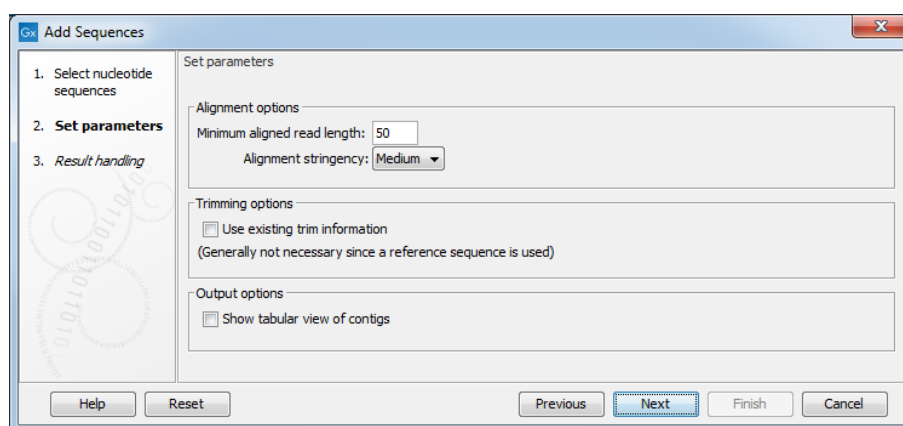


Figure 22.11: Setting assembly parameters when assembling to an existing contig.

The options in this dialog are similar to the options that are available when assembling to a reference sequence (see section 22.4).

Click **Finish** to start the tool. This will start the assembly process. See section 22.7 on how to use the resulting contig.

Note that the new sequences will be added to the existing contig which will not be extended. If the new sequences extend beyond the existing contig, they will be cut off.

22.7 View and edit contigs and read mappings

The results of Assemble Sequences or Assemble Sequences to Reference are contigs or read mappings, respectively (figure 22.12). A mapping is generated for each reference supplied as input to Assemble Sequences to Reference. Individual read mappings can be opened by double-clicking on a row in the table that lists the mappings.

Customizing how a contig or mapping looks is covered in section 22.7.1.

The residue colors and traces at the ends of reads are faded in regions that do not contribute to the final contig or mapping results. This may be due to trimming before or during the assembly or to misalignment with other reads (assembly) or the reference sequence (mapping).

Simply drag the edge of the faded section to adjust the trimmed area to include more of the read in the contig or mapping (figure 22.13).

Note: Handles for dragging are only available when individual residues can be seen. For this, zoom fully in and chose a **Compactness** level of "Not compact", "Low" or "Packed".

To reverse complement an entire contig or mapping, right-click in the empty white area of the contig or mapping and choose to **Reverse Complement Sequence**.

22.7.1 View settings in the Side Panel

Here we cover a few of the key settings available in the side panel for customizing how a contig or read mapping looks. We step through them according to the tab names in the side panel.

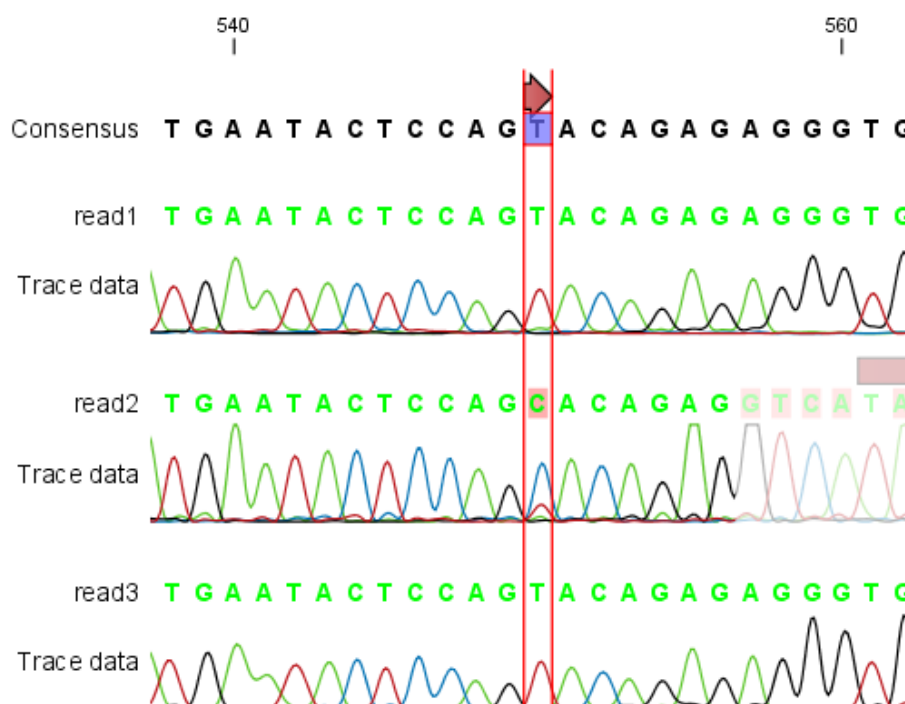


Figure 22.12: The view of a contig. Controls at the bottom allow you to zoom in and out, and settings to the right control how the mapping is displayed.



Figure 22.13: Drag the edge of the faded area to customize how much of a read should be considered in the mapping.

Read layout.

- **Compactness.** Set the level of detail to be displayed. The level of compactness affects other view settings as well as the overall view. For example: if **Compact** is selected, quality scores and annotations on the reads will not be visible, even if these options are turned on under the "Nucleotide info" palette. Compactness can also be changed by pressing and holding the Alt key while scrolling with the mouse wheel or touchpad.
 - **Not compact.** This allows the mapping to be viewed in full detail, including quality scores and trace data for the reads, where present. To view such information, additional viewing options under the **Nucleotide info** view settings must also be selected. For further details on these, see section 22.1.1 and section 15.2.1.
 - **Low.** Hides trace data, quality scores and puts the reads' annotations on the sequence. The editing functions available when right-clicking on a nucleotide with

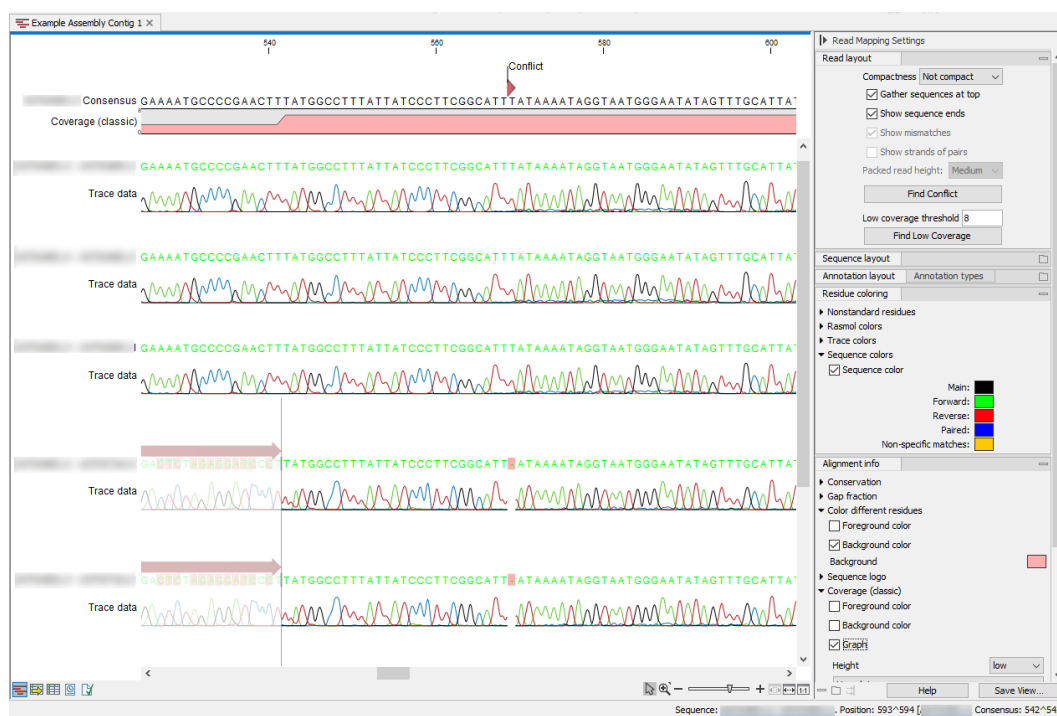


Figure 22.14: Settings in the side panel allow customization of the view of read mappings and contigs from assemblies.

compactness set to Low is shown in figure 22.15.

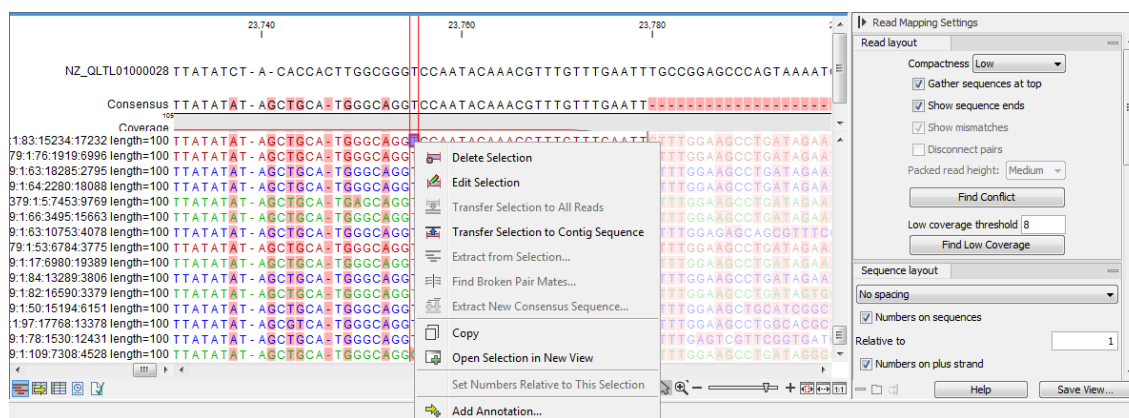


Figure 22.15: An example showing the low compactness setting.

- **Medium.** The labels of the reads and their annotations are hidden, and reads are shown as lines. The residues of the reads cannot be seen, even when zoomed in 100%.
- **Compact.** Like Medium but with less space between the reads.
- **Packed.** This uses all the horizontal space available for displaying the reads (figure 22.16). This differs from the other settings, which stack all reads vertically. When zoomed in to 100%, the individual residues are visible. When zoomed out, reads are represented as lines. Packed mode is useful when viewing large amounts of data, but some functionality is not available. For example, the read mapping cannot be edited, portions cannot be selected, and color coding changes

are not possible.

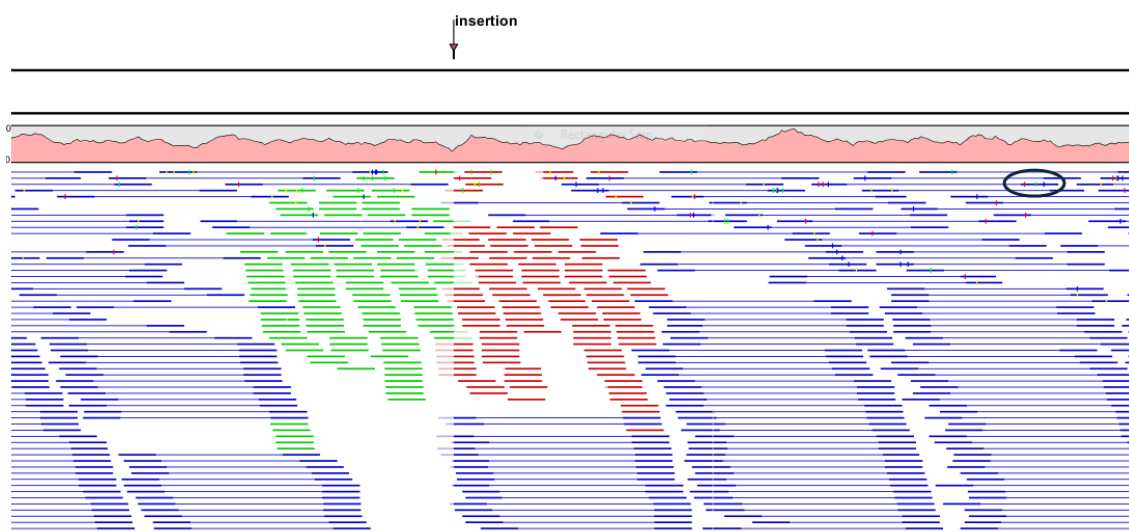


Figure 22.16: An example of the Packed compactness setting. Highlighted in black is an example of 3 narrow vertical lines representing mismatching residues.

- **Gather sequences at top.** When selected, the sequence reads contributing to the mapping at that position are placed right below the reference. This setting has no effect when the compactness level is Packed.
- **Show sequence ends.** When selected, trimmed regions are shown (faded traces and residues). Trimmed regions do not contribute to the mapping or contig.
- **Show mismatches.** When selected and when the compactness is set to Packed, based that do not match the reference at that position are highlighted by coloring them according to the Rasmol color scheme. Reads with mismatches are floated to the top of the view.
- **Show strands of paired reads.** When the compactness is set to Packed, display each member of a read pair in full and color them according to direction. This is particularly useful for reviewing overlap regions in overlapping read pairs.
- **Packed read height.** When the compactness is set to "Packed", select a height for the visible reads.

When there are more reads than the height specified, an overflow graph is displayed that uses the same colors as the sequences. Mismatches in reads are shown as narrow vertical lines, using colors representing the mismatching residue. Horizontal line colors correspond to those used for highlighting mismatches in the sequences (red = A, blue = C, yellow = G, and green = T). For example, a red line with half the height of the blue part of the overflow graph represents a mismatching "A" in half of the paired reads at that particular position.

- **Find Conflict.** Clicking this button selects the next position where there is an conflict. Mismatching residues are colored using the default color settings. You can also click on the Space bar of your keyboard to find the next conflict.
- **Low coverage threshold.** All regions with coverage up to and including this value are considered low coverage. Clicking the 'Find low coverage' button selects the next region in the read mapping with low coverage.

Sequence layout. There is one parameter in this section in addition to those described in section 15.2.1

- **Matching residues as dots.** When selected, matching residues are presented as dots instead of as letters.

Annotation Layout and Annotation Types See section 15.3.1.

Residue coloring. There is one parameter in this section in addition to those described in section 15.2.1.

- **Sequence colors.** This setting controls the coloring of sequences when working in most compactness modes. The exception is Packed mode, where colors are controlled with settings under the "Match coloring" tab, described below.
 - **Main.** The color of the consensus and reference sequence. Black by default.
 - **Forward.** The color of forward reads. Green by default.
 - **Reverse.** The color of reverse reads. Red by default.
 - **Paired.** The color of read pairs. Blue by default. Reads from **broken pairs** are colored according to their orientation (forward or reverse) or as a non-specific match, but with a darker hue than the color of ordinary reads.
 - **Non-specific matches.** When a read would have matched equally well another place in the mapping, it is considered a non-specific match and is colored yellow by default. Coloring to indicate a non-specific match overrules other coloring. For mappings with several reference sequences, a read is considered a non-specific match if it matches more than once *across all the contigs/references*.

Colors can be adjusted by clicking on an individual color and selecting from the palette presented.

Alignment info. There are several parameters in this section in addition to the ones described in section 24.2.

- **Coverage:** Shows how many reads are contributing information to a given position in the read mapping. The level of coverage is relative to the overall number of reads.
- **Paired distance:** Plots the distance between the members of paired reads.
- **Single paired reads:** Plots the percentage of reads marked as single paired reads (when only one of the reads in a pair matches).
- **Non-specific matches:** Plots the percentage of reads that also match other places.
- **Non-perfect matches:** Plots the percentage of reads that do not match perfectly.
- **Spliced matches:** Plots the percentage of reads that are spliced.

Options for these parameters are:

- **Foreground color.** Colors the residues using a gradient, where the left side color is used for low coverage and the right side is used for maximum coverage.
- **Background color.** Colors the background of the residues using a gradient, where the left side color is used for low coverage and the right side is used for maximum coverage.

- **Graph.** Read coverage is displayed as a graph. The data points for the graph can be exported (see section 8.3).
 - **Height.** Specifies the height of the graph.
 - **Type.** The graph can be displayed as Line plot, Bar plot or as a Color bar.
 - **Color box.** For Line and Bar plots, the color of the plot can be set by clicking the color box. If a Color bar is chosen, the color box is replaced by a gradient color box as described under Foreground color.

Nucleotide info See section 15.2.1

Match coloring Coloring of the mapped reads when the Packed compactness option is selected. Colors can be adjusted by clicking on an individual color and selecting from the palette presented. Coloring of bases when other compactness settings are selected is controlled under the "Residue coloring" tab.

Find See section 15.2.1

Text format See section 15.2.1

22.7.2 Editing a contig or read mapping

When editing contigs and read mappings, you are typically interested in confirming or changing single bases.

To do this:

Select the base to edit | Type the desired base

If you want to replace a residue with a gap, use the **Delete** key.

If you wish to edit a selection of more than one residue:

right-click the selection | Edit Selection ()

There are three shortcut keys for easily finding the positions where there are conflicts:

- Space bar: Finds the *next* conflict.
- "." (punctuation mark key): Finds the *next* conflict.
- "," (comma key): Finds the *previous* conflict.

In the contig or mapping view, you can use **Zoom in** () to zoom to a greater level of detail than in other views (see figure 22.12).

Note: For contigs or mappings with more than 1,000 reads, you can only do single-residue replacements. When the compactness is **Packed**, you cannot edit any of the reads.

All changes are recorded in the history of the element (see section 2.5).

22.7.3 Sorting reads

To change the order of the sequence reads, simply drag the label of the sequence up and down. Note that this is not possible if you have chosen **Gather sequences at top** or set the compactness to **Packed** in the **Side Panel**.



You can also sort the reads by right-clicking a sequence label and choose from the following options:

- **Sort Reads by Alignment Start Position.** This will list the first read in the alignment at the top etc.
- **Sort Reads by Name.** Sort the reads alphabetically.
- **Sort Reads by Length.** The shortest reads will be listed at the top.

22.7.4 Read conflicts

After assembly or mapping, conflicts between the reads are annotated on the consensus sequence. The definition of a conflict is *a position where at least one of the reads has a different residue compared to the reference*.

A conflict can be in two states:

- **Conflict.** Both the annotation and the corresponding row in the Table  are colored **red**.
- **Resolved.** Both the annotation and the corresponding row in the Table  are colored **green**.


The conflict can be resolved by correcting the deviating residues in the reads as described above.

A fast way of making all the reads reflect the consensus sequence is to select the position in the consensus, right-click the selection, and choose **Transfer Selection to All Reads**.


The opposite is also possible: make a selection on one of the reads, right click, and **Transfer Selection to Contig Sequence**.

22.7.5 Using the mapping

Due to the integrated nature of *CLC Genomics Workbench* it is easy to use the consensus sequences as input for additional analyses. If you wish to extract the consensus sequence for further use, use the **Extract Consensus Sequence** tool (see section 22.10).

You can also right-click the consensus sequence and select **Open Sequence**. This will not create a new sequence but simply let you see the sequence in a sequence view. This means that the sequence still "belong" to the mapping and will be saved together with the mapping. It also means that if you add annotations to the sequence, they will be shown in the mapping view as well. This can be very convenient for Primer design ( for example.

If you wish to BLAST the consensus sequence, simply select the whole contig for your BLAST search. It will automatically extract the consensus sequence and perform the BLAST search.

In order to preserve the history of the changes you have made to the contig, the contig itself should be saved from the contig view, using either the save button () or by dragging it to the **Navigation Area**.

22.7.6 Extracting reads from mappings

This section describes functionality for extracting reads from stand-alone read mappings. Extracting reads from reads tracks and sequence tracks is described in section 27.3.5.

Reads can be extracted from stand-alone read mappings in multiple ways:

- **Extract from Selection.** Available from the right-click menu of the reference sequence or consensus sequence (figure 22.17). A new stand-alone read mapping consisting of just the reads that are completely covered by the selected region will be created. Options are available to specify the nature of the extracted reads (e.g. match specificity, paired status, etc.). These options are the same as those provided in the **Extract Reads** tool.
- **Extract Reads.** Available from the Tools menu. It extracts all reads or a subset of reads, specified based on location relative to a set of regions and/or based on specified characteristics of the reads. Reads can be extracted to a reads track or sequence list. See section 37.2.
- **Extract Sequences.** Available from the right-click menu of the coverage graph or a read (figure 22.18), or from the Tools menu. It extracts all reads to a sequence list or individual sequences. See section 18.2.

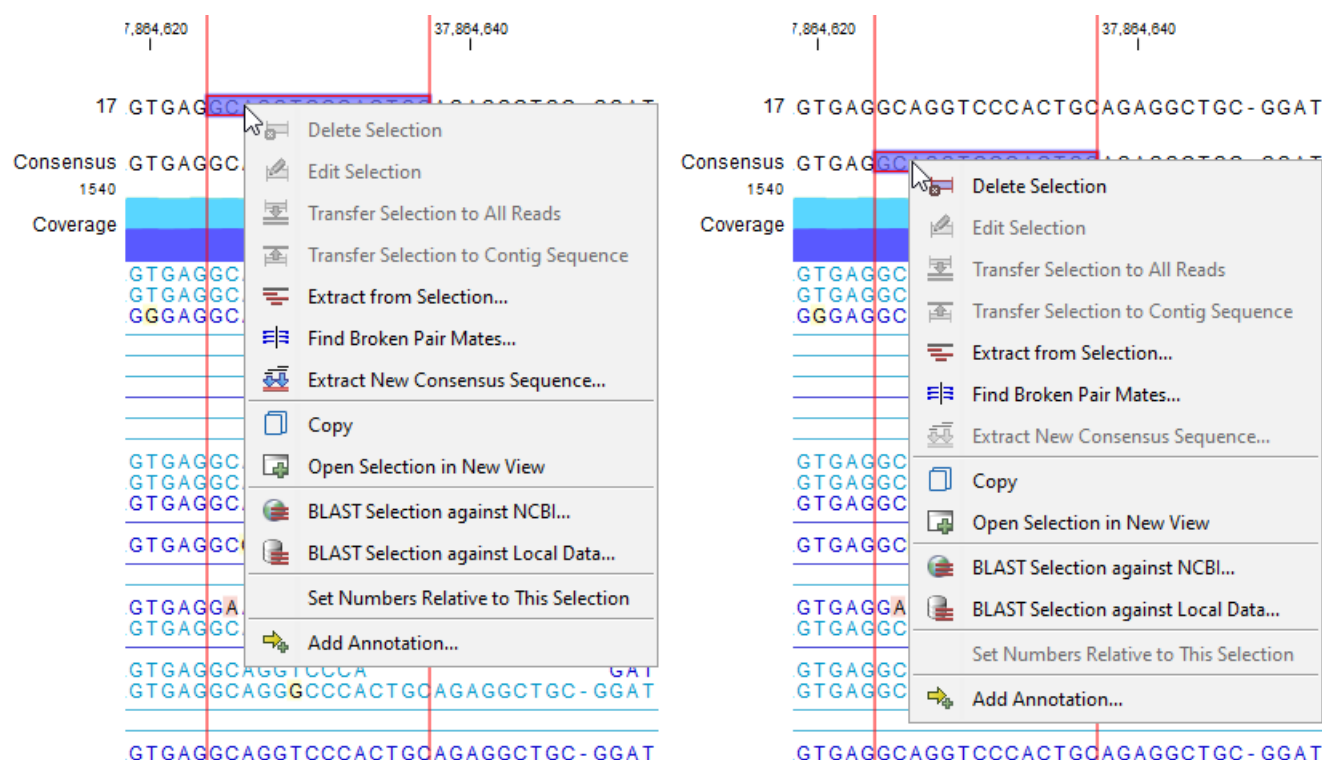


Figure 22.17: Right-click on the selected region on the reference sequence (left) or consensus sequence (right) in a stand-alone read mapping for revealing the available options.

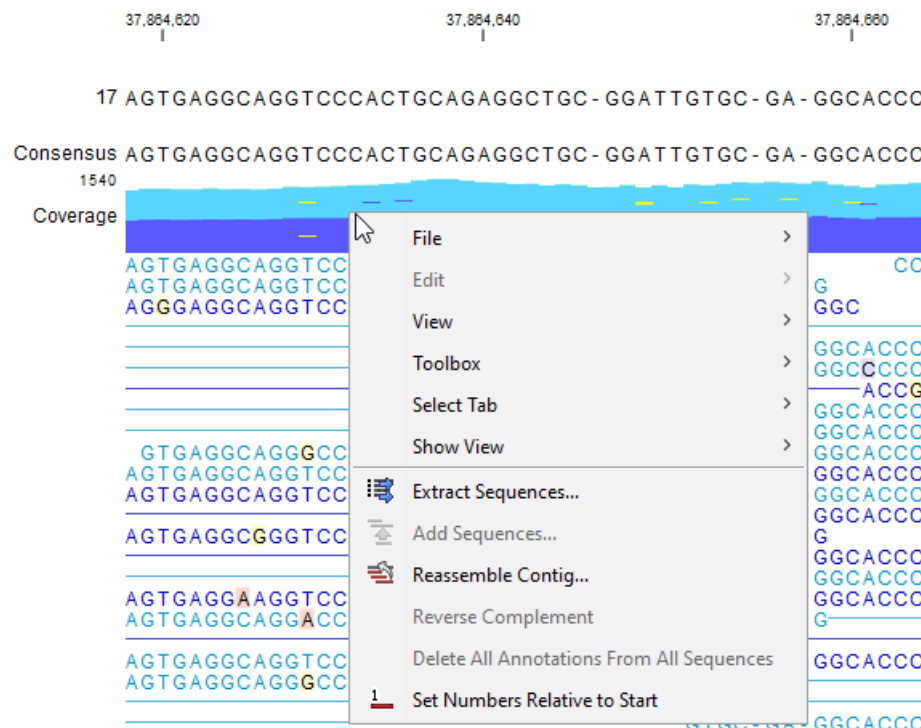


Figure 22.18: Right-click on the coverage graph or reads for revealing the available options.

22.7.7 Variance table

In addition to the standard graphical display of a contig or mapping as described above, you can also see a tabular overview of the conflicts between the reads by clicking the **Table** (📊) icon at the bottom of the view.

This will display a new view of the conflicts as shown in figure 22.19.

The table has the following columns:

- **Reference position.** The position of the conflict measured from the starting point of the reference sequence.
- **Consensus position.** The position of the conflict measured from the starting point of the consensus sequence.
- **Consensus residue.** The consensus's residue at this position. The residue can be edited in the graphical view, as described above.
- **Other residues.** Lists the residues of the reads. Inside the brackets, you can see the number of reads having this residue at this position. In the example in figure 22.19, you can see that at position 637 there is a 'C' in the top read in the graphical view. The other two reads have a 'T'. Therefore, the table displays the following text: 'C (1), T (2)'.
- **IUPAC.** The ambiguity code for this position. The ambiguity code reflects the residues in the reads - not in the consensus sequence. (The IUPAC codes can be found in section H.)
- **Status.** The status can either be conflict or resolved:



Figure 22.19: The graphical view is displayed at the top, and underneath the conflicts are shown in a table. At the conflict at position 313, the user has entered a comment in the table (to see it, make sure the Notes column is wide enough to display all text lines). This comment is now also added to the tooltip of the conflict annotation in the graphical view above.

- **Conflict.** Initially, all the rows in the table have this status. This means that there is one or more differences between the sequences at this position.
- **Resolved.** If you edit the sequences, e.g. if there was an error in one of the sequences, and they now all have the same residue at this position, the status is set to *Resolved*.
- **Note.** Can be used for your own comments on this conflict. Right-click in this cell of the table to add or edit the comments. The comments in the table are associated with the conflict annotation in the graphical view. Therefore, the comments you enter in the table will also be attached to the annotation on the consensus sequence (the comments can be displayed by placing the mouse cursor on the annotation for one second - see figure 22.19). The comments are saved when you **Save** (⌘S).

By clicking a row in the table, the corresponding position is highlighted in the graphical view. Clicking the rows of the table is another way of navigating the contig or the mapping, as are using the **Find Conflict** button or using the **Space bar**. You can use the up and down arrow keys to navigate the rows of the table.

22.8 Reassemble contig

If you have edited a contig, changed trimmed regions, or added or removed reads, you may wish to reassemble the contig. This can be done in two ways:

Tools | Molecular Biology Tools (M) | **Sanger Sequencing Analysis (S)** | **Reassemble Contig (R)**

Select the contig from Navigation Area, move to 'Selected Elements' and click Next. You can also right-click in the empty white area of the contig and choose to **Reassemble contig** (🖱️).

This opens a dialog as shown in figure 22.20

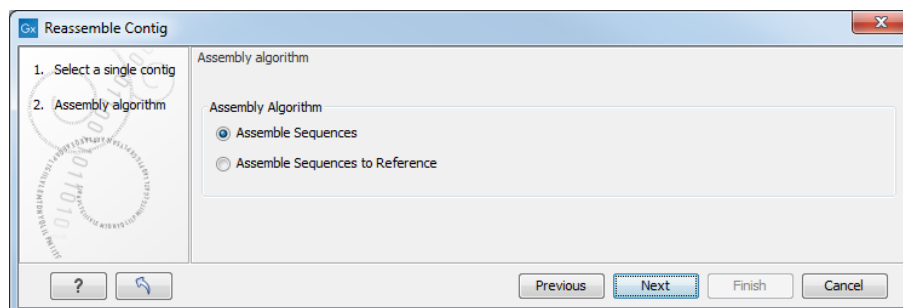


Figure 22.20: Re-assembling a contig.

In this dialog, you can choose:

- **De novo assembly.** This will perform a normal assembly in the same way as if you had selected the reads as individual sequences. When you click **Next**, you will follow the same steps as described in section 22.3. The consensus sequence of the contig will be ignored.
- **Reference assembly.** This will use the consensus sequence of the contig as reference. When you click **Next**, you will follow the same steps as described in section 22.4.

When you click **Finish**, a new contig is created, so you do not lose the information in the old contig.

22.9 Secondary peak calling

CLC Genomics Workbench is able to detect secondary peaks - a peak within a peak - to help discover heterozygous mutations. Looking at the height of the peak below the top peak, the CLC Genomics Workbench considers all positions in a sequence, and if a peak is higher than the threshold set by the user, it will be "called".

The peak detection investigates any secondary high peaks in the same interval as the already called peaks. The peaks must have a peak shape in order to be considered (i.e. a fading signal from the previous peak will be ignored). **Note!** The secondary peak caller does not call and annotate secondary peaks that have already been called by the Sanger sequencing machine and denoted with an ambiguity code.

Regions that are trimmed (i.e. covered by Trim annotations) are ignored in the analysis (section 22.2).

When a secondary peak is called, the residue is changed to an ambiguity character to reflect that two bases are possible at this position, and optionally an annotation is added at this position.

To call secondary peaks, go to:

Tools | Molecular Biology Tools (🔧) | **Sanger Sequencing Analysis** (📊) | **Call Secondary Peaks** (📈)

This opens a dialog where you can add the sequences to be analyzed. If you had already selected sequence in the Navigation Area, these will be shown in the 'Selected Elements' box. However you can remove these, or add others, by using the arrows to move sequences between the Navigation Area and Selected Elements boxes.

When the sequences are selected, click **Next**.

This opens the dialog displayed in figure 22.21.

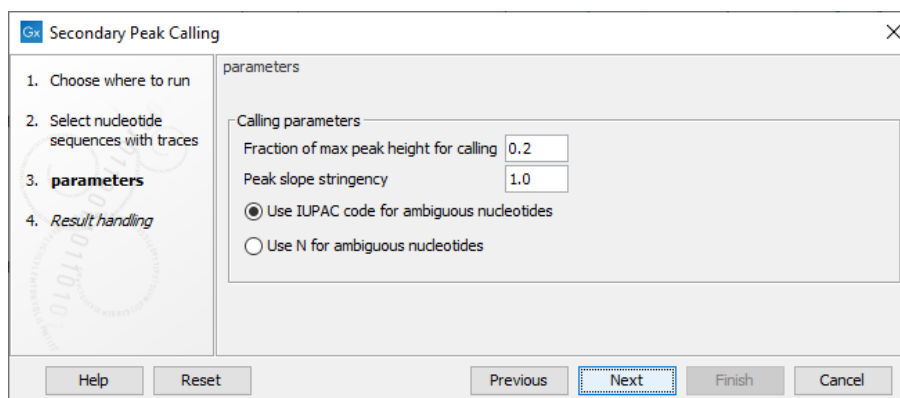


Figure 22.21: Setting parameters secondary peak calling.

The following parameters can be adjusted in the dialog:

- **Fraction of max peak height for calling.** Adjust this value to specify how high the secondary peak must be to be called.
- **Peak slope stringency.** Control how pronounced each nucleotide peak must be. Decreasing this will detect more peaks. Increasing it will detect fewer.
- **Use IUPAC code / N for ambiguous nucleotides.** When a secondary peak is called, the residue at this position can either be replaced by an N or by a ambiguity character based on the IUPAC codes (see section H).

Clicking **Next** allows you to add annotations. In addition to changing the actual sequence, annotations can be added for each base that has been called. The annotations hold information about the fraction of the max peak height.

Click **Finish** to start the tool. This will start the secondary peak calling. A detailed history entry will be added to the history specifying all the changes made to the sequence.

Secondary peaks are marked in the output sequence as can be seen in figure 22.22. When the mouse is hovered over a secondary peak, **Before** and **Peak ratio** values are shown. The **Before** value refers to the original residue that was present in the sequence, while the **Peak ratio** shows the ratio between the original peak and the secondary peak signal strength values (the base associated with the secondary peak is shown in parentheses next to the peak ratio). In the case of figure 22.22, it can be seen that the original residue is G while the residue C yields a secondary peak. This then results in the ambiguity character S shown in the sequence.

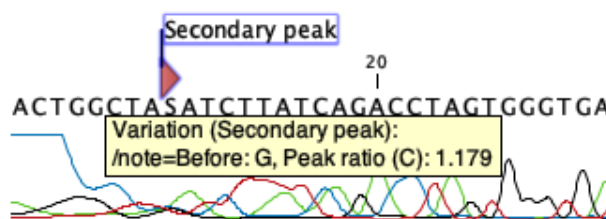


Figure 22.22: Secondary peaks are marked in the output sequence.

22.10 Extract Consensus Sequence

Extract Consensus Sequence takes as input stand-alone read mappings, read tracks, or nucleotide BLAST results, and outputs a consensus sequence.

Consensus sequences can also be extracted directly from elements opened in the View Area by selecting **Extract New Consensus Sequence** (🔧) from the right-click menu on:

- For stand-alone read mappings: the reference sequence or a selection within it, or the consensus sequence.
- For nucleotide BLAST results: a selection within the query sequence.

To run the tool, go to:

Tools | Resequencing Analysis (🔧) | **Extract Consensus Sequence** (🔧)

The following options for extracting the consensus sequence can be configured (figure 22.23):

- **Threshold** Coverage above this value is considered high, while coverage at or below it is considered low. Overlapping paired-end reads count as two when calculating the coverage. A higher threshold yields a more reliable consensus but reduces completeness.
- **Low coverage handling** Options for handling regions with low coverage.
 - **Remove regions with low coverage** No consensus is generated in regions of low coverage.
 - * **Join after removal** High coverage regions are joined into a single consensus sequence.
 - * **Split into separate sequences** Each high coverage region produces an individual consensus sequence.
 - **Insert 'N' ambiguity symbols** The consensus is set to 'N' at each position with low coverage.
 - **Fill from reference sequence** The consensus is set to the reference symbol at each position with low coverage.
- **Conflict resolution** Options for resolving read disagreements at individual positions within high coverage regions.
 - **Vote** The consensus is set to the most supported symbol (see **Use quality score**), excluding ambiguous symbols.

Figure 22.23: Options for consensus sequence extraction.

In case of a tie, symbols are chosen in the order: A > C > G > T.

To preserve biological heterozygous variation, see **Insert ambiguity codes**.

- **Insert ambiguity codes** The consensus is set to the IUPAC ambiguity code (section H) that best reflects the variation observed in the reads.

The following options determine which symbols contribute to the ambiguity codes:

- * **Noise threshold** Only symbols with support greater than this value (see **Use quality score**) contribute to the ambiguity code.
- * **Minimum nucleotide count** Only symbols present in at least this number of reads contribute to the ambiguity code.

Positions where no symbol qualifies to contribute to the ambiguity code are not included in the consensus.

- **Use quality score** The support of a symbol is determined by:
 - When unchecked: the percentage of reads containing the symbol.
 - When checked: the sum of the symbol's quality scores, expressed as a percentage of the total quality score at that position.

The following options for annotations the consensus sequence can be configured (figure 22.24):

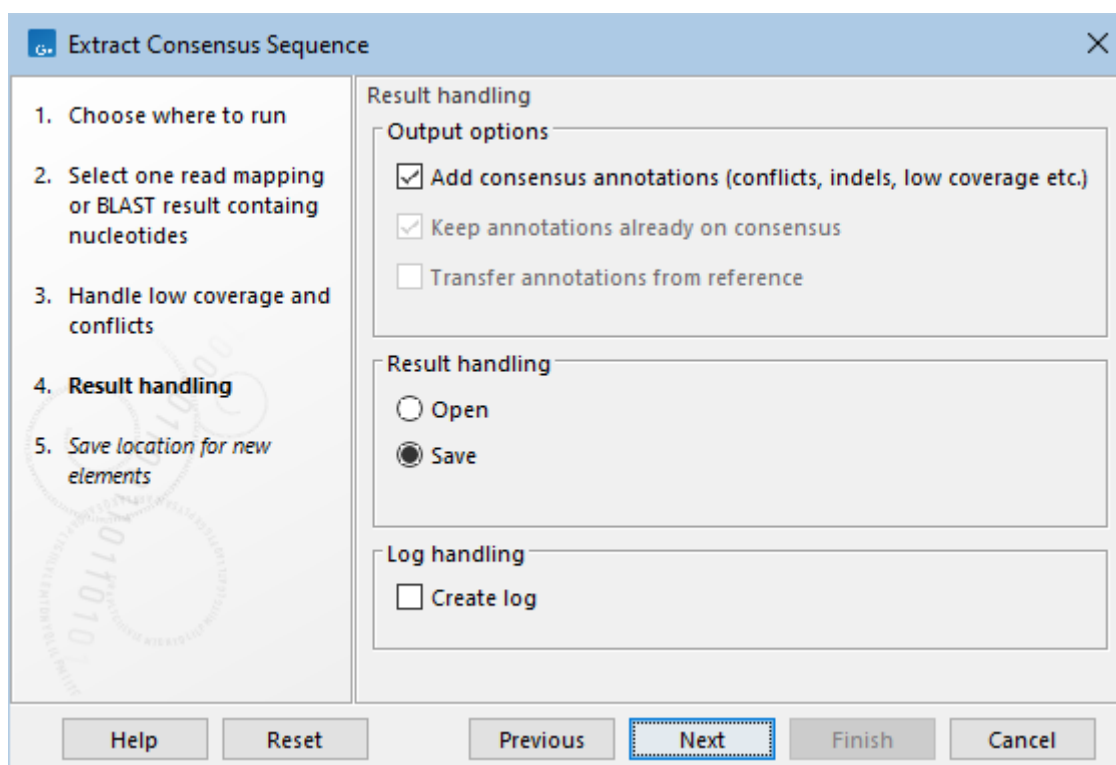


Figure 22.24: Options for adding annotations to the extracted consensus sequence.

- **Add consensus annotations (conflicts, indels, low coverage etc.)** When checked, annotations are added to the consensus sequence to indicate resolved conflicts, deletions relative to the reference, and low coverage regions, provided the **Split into separate sequences** option is not selected.

For inputs containing many reads or long references, many such annotations may be generated.

- **Keep annotations already on consensus** and **Transfer annotations from reference** When checked, annotations present on the consensus or on the reference in the input stand-alone read mapping are copied to the extracted consensus sequence. The copied annotations are placed in regions corresponding to their original location in the input data, although actual coordinates may differ. Annotations may be split if the **Split into separate sequences** option is selected.

These options are not enabled for types of input other than stand-alone read mapping.

Quality scores on the consensus sequence

When quality scores are present in the input, they are propagated to the extracted consensus as follows.

Consider a consensus symbol X , and let Y and Z be quality scores sums at its position, defined as:

- Y : The sum of quality scores from all reads.
- Z : The sum of quality scores from reads supporting X , defined as:
 - Reads containing the symbol X , if the **Vote** option is selected.
 - Reads containing symbols that contribute to the ambiguity code, if the **Insert ambiguity codes** option is selected.

The consensus symbol X is assigned a quality score $Q = Z - (Y - Z)$, bounded to a minimum of 0 and a maximum of 64.

Chapter 23

Cutting and cloning

Contents

23.1 Restriction site analyses	587
23.1.1 Dynamic restriction sites	588
23.1.2 Restriction Site Analysis	593
23.1.3 Insert restriction site	596
23.2 Restriction enzyme lists	596
23.3 Restriction Based Cloning	598
23.3.1 Introduction to the Cloning Editor	599
23.3.2 The restriction cloning workflow	600
23.3.3 Manual cloning	602
23.4 Homology Based Cloning	606
23.4.1 Working with homology based cloning	607
23.4.2 Adjust the homology based cloning design	608
23.4.3 Homology Based Cloning outputs	610
23.4.4 Detailed description of the Homology Based Cloning wizard	611
23.4.5 Working with mutations	615
23.5 Gateway cloning	615
23.5.1 Add attB sites	616
23.5.2 Create entry clones (BP)	619
23.5.3 Create expression clones (LR)	620
23.6 Gel electrophoresis	622
23.6.1 Gel view	623

CLC Genomics Workbench offers graphically advanced *in silico* cloning and design of vectors, together with restriction enzyme analysis and functionalities for managing lists of restriction enzymes.

23.1 Restriction site analyses

There are two ways of finding and showing restriction sites:

- In many cases, the dynamic restriction sites found in the **Side Panel** of sequence views is the fastest and easiest way of showing restriction sites.
- Under the **Tools** menu you will find the **Restriction Sites Analysis** tool, which provides more control over the analysis and more output options, such as a table of restriction sites. It also allows you to perform the same restriction map analysis on several sequences in one step.

23.1.1 Dynamic restriction sites

If you open a sequence or a sequence list, you will find a **Restriction sites** palette available in the Side Panel.

Restriction sites can be shown on the sequence as colored triangles and lines (figure 23.1): check the "Show" option on top of the Restriction sites section, then specify the enzymes that should be displayed.



Figure 23.1: Showing restriction sites of ten restriction enzymes.

The color of the restriction enzyme can be changed by clicking the colored box next to the enzyme's name. The name of the enzyme can also be shown next to the restriction site by selecting **Show** above the list of restriction enzymes.

There is also an option to specify how the **Labels** should be shown:

- **No labels.** This will just display the cut site with no information about the name of the enzyme. Placing the mouse button on the cut site will reveal this information as a tool tip.
- **Flag.** This will place a flag just above the sequence with the enzyme name (see an example in figure 23.2). Note that this option will make it hard to see when several cut sites are

located close to each other. In the circular view, this option is replaced by the Radial option.



Figure 23.2: Restriction site labels shown as flags.

- **Radial.** This option is only available in the circular view. It will place the restriction site labels as close to the cut site as possible (see an example in figure 23.3).

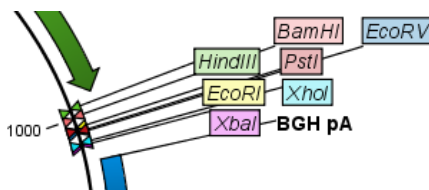


Figure 23.3: Restriction site labels in radial layout.

- **Stacked.** This is similar to the flag option for linear sequence views, but it will stack the labels so that all enzymes are shown. For circular views, it will align all the labels on each side of the circle. This can be useful for clearly seeing the order of the cut sites when they are located closely together (see an example in figure 23.4).

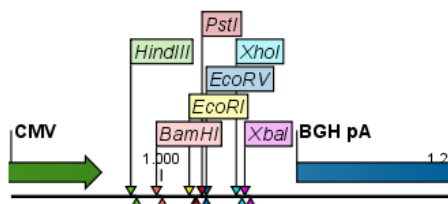


Figure 23.4: Restriction site labels stacked.

Note that in a circular view, the **Stacked** and **Radial** options also affect the layout of annotations. Just above the list of enzymes, three buttons can be used for sorting the list (see figure 23.5).

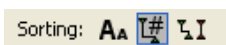


Figure 23.5: Buttons to sort restriction enzymes.

- **Sort enzymes alphabetically (AA).** Clicking this button will sort the list of enzymes alphabetically.
- **Sort enzymes by number of restriction sites (T#).** This will divide the enzymes into four groups:
 - Non-cutters.
 - Single cutters.
 - Double cutters.

- Multiple cutters.

There is a checkbox for each group which can be used to hide / show all the enzymes in a group.

- **Sort enzymes by overhang** (🔍). This will divide the enzymes into three groups:

- Blunt. Enzymes cutting both strands at the same position.
- 3'. Enzymes producing an overhang at the 3' end.
- 5'. Enzymes producing an overhang at the 5' end.

There is a checkbox for each group which can be used to hide / show all the enzymes in a group.

Manage enzymes

The list of restriction enzymes contains per default some of the most popular enzymes, but you can easily modify this list and add more enzymes by clicking the **Manage enzymes button** found at the bottom of the "Restriction sites" palette of the Side Panel.

This will open the dialog shown in figure 23.6.

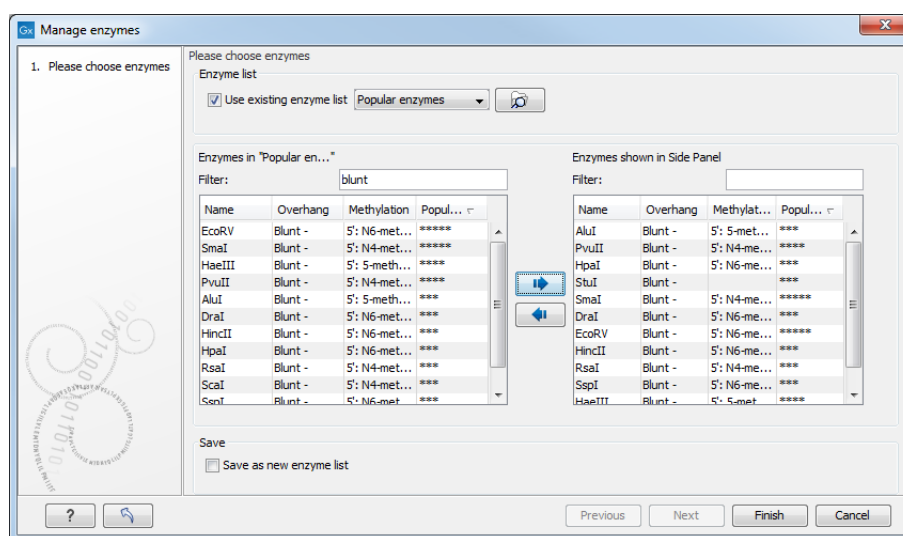


Figure 23.6: Adding or removing enzymes from the Side Panel.

At the top, you can choose to **Use existing enzyme list**. Clicking this option lets you select an enzyme list which is stored in the **Navigation Area**. A list of popular enzymes is available in the Example Data folder you can download from the Help menu.

Below there are two panels:

- To the **left**, you can see all the enzymes that are in the list selected above. If you have not chosen to use a specific enzyme list, this panel shows all the enzymes available.
- To the **right**, you can see the list of the enzymes that will be used.

Select enzymes in the left side panel and add them to the right panel by double-clicking or clicking the **Add** button (➡).

The enzymes can be sorted by clicking the column headings, i.e., Name, Overhang, Methylation or Popularity. This is particularly useful if you wish to use enzymes which produce a 3' overhang for example.

When looking for a specific enzyme, it is easier to use the Filter. You can type HindIII or blunt into the filter, and the list of enzymes will shrink automatically to only include respectively only the HindIII enzyme, or all enzymes producing a blunt cut.

If you need more detailed information and filtering of the enzymes, you can hover your mouse on an enzyme (see figure 23.7). You can also open a view of an enzyme list saved in the Navigation Area.

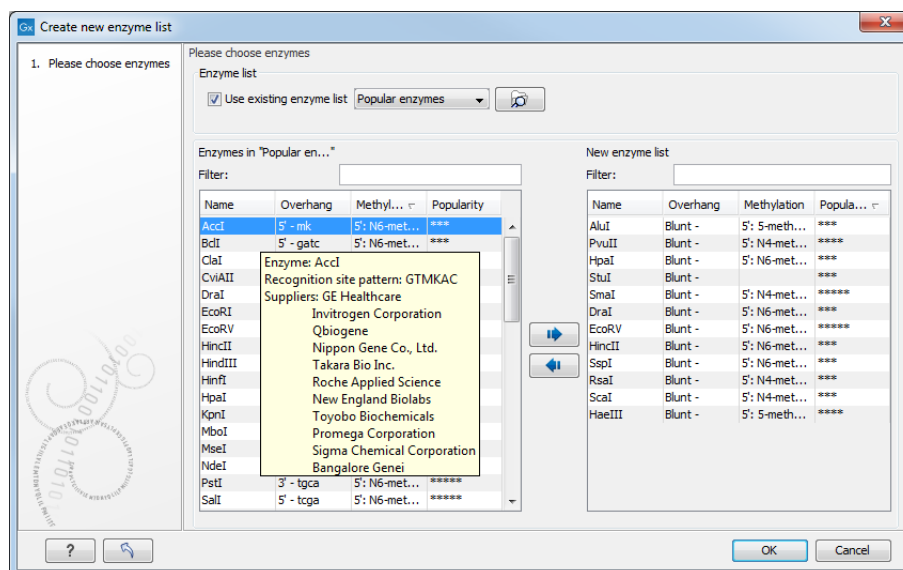


Figure 23.7: Showing additional information about an enzyme like recognition sequence or a list of commercial vendors.

At the bottom of the dialog, you can select to save the updated list of enzymes as a new file. When you click on **Finish**, the enzymes are added to the Side Panel and the cut sites are shown on the sequence. You can save the settings in the Side Panel, including the enzymes just added, as described in section 4.6).

Show enzymes cutting inside/outside selection

In cases where you have a selection on a sequence, and you wish to find enzymes cutting within the selection but not outside, right-click the selection and choose the option **Show Enzymes Cutting Inside/Outside Selection** (🔍).

This will open a wizard where you can specify which enzymes should initially be considered (see section 23.1.1). You can for example select all the enzymes from a custom made list that correspond to all the enzymes that are already available in your lab.

In the following step (figure 23.8), you can define the terms of your search.

At the top of the dialog, you see the selected region, and below are two panels:

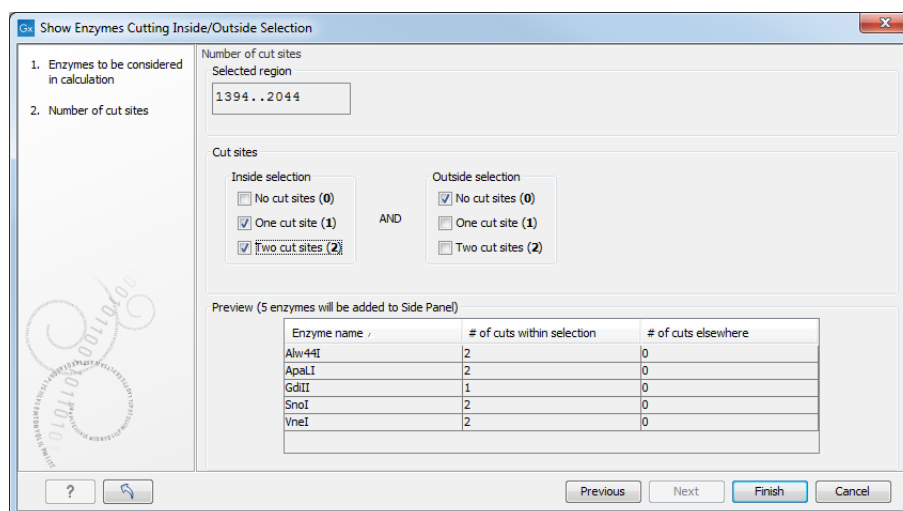


Figure 23.8: Deciding number of cut sites inside and outside the selection.

- **Inside selection.** Specify how many times you wish the enzyme to cut inside the selection.
- **Outside selection.** Specify how many times you wish the enzyme to cut outside the selection (i.e. the rest of the sequence).

These panels offer a lot of flexibility for combining number of cut sites inside and outside the selection, respectively. To give a hint of how many enzymes will be added based on the combination of cut sites, the preview panel at the bottom lists the enzymes which will be added when you click **Finish**. Note that this list is dynamically updated when you change the number of cut sites. The enzymes shown in brackets [] are enzymes which are already present in the Side Panel.

If you have selected more than one region on the sequence (using Ctrl or ⌘), they will be treated as individual regions. This means that the criteria for cut sites apply to each region.

Show enzymes with compatible ends

A third way of adding enzymes to the Side Panel and thereby displaying them on the sequence is based on the overhang produced by cutting with an enzyme. Right-click on a restriction site and choose to **Show Enzymes with Compatible Ends (⌘ I)** to find enzymes producing a compatible overhang (figure 23.9).

At the top you can choose whether the enzymes considered should have an exact match or not. We recommend trying **Exact match** first, and use **All matches** as an alternative if a satisfactory result cannot be achieved. Indeed, since a number of restriction enzymes have ambiguous cut patterns, there will be variations in the resulting overhangs. Choosing **All matches**, you cannot be 100% sure that the overhang will match, and you will need to inspect the sequence further afterwards.

Use the arrows between the two panels to select enzymes which will be displayed on the sequence and added to the Side Panel.

At the bottom of the dialog, the list of enzymes producing compatible overhangs is shown.

When you have added the relevant enzymes, click **Finish**, and the enzymes will be added to the

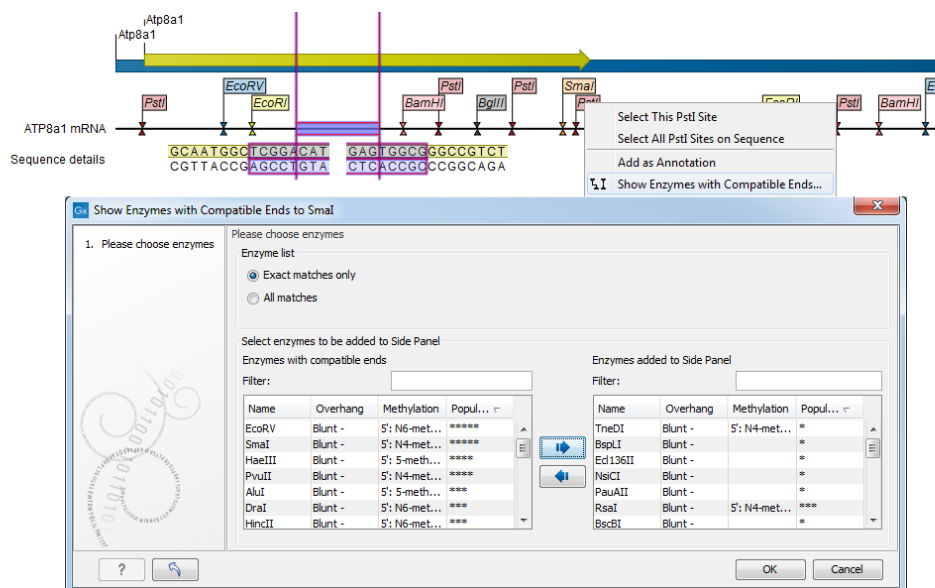


Figure 23.9: Enzymes with compatible ends.

Side Panel and their cut sites displayed on the sequence.

This functionality does not work for enzymes where the cut site is located outside the recognition site.

23.1.2 Restriction Site Analysis

To run the **Restriction Site Analysis** tool, go to:

Tools | Molecular Biology Tools (🔬) | Cloning (🧬) | Restriction Site Analysis (✂️)

You first specify the sequences to analyze, and in the next step, which enzymes to use. See section 23.1.1 for information about managing the restriction enzymes available.

In the next wizard step, you can limit the list of sites reported, depending on how many times they cut a sequence (figure 23.10). The default is to report enzymes that cut the sequence one or two times.

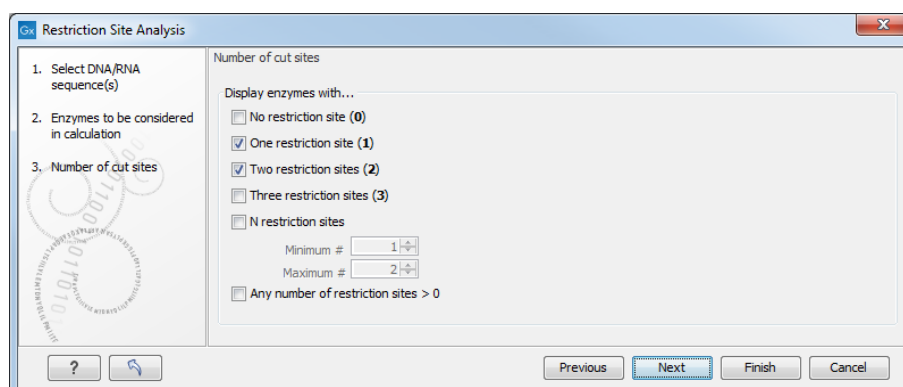


Figure 23.10: Selecting number of cut sites.

The Result handling wizard step (figure 23.11) lets you specify how the result of the restriction

map analysis should be presented.

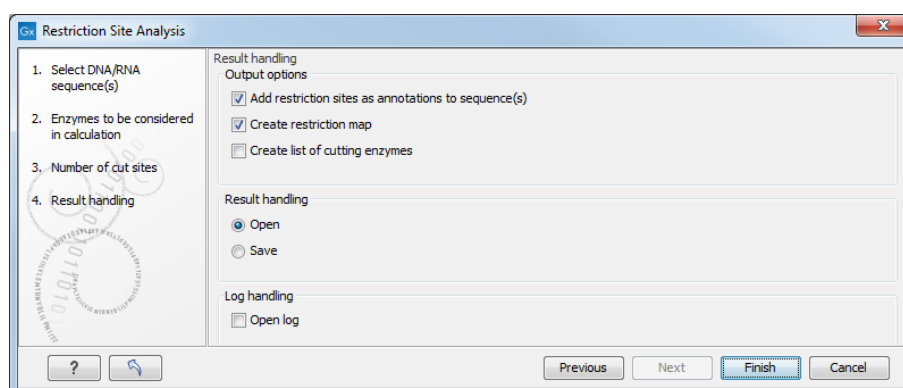


Figure 23.11: Choosing to add restriction sites as annotations or creating a restriction map.

Add restriction sites as annotations to sequence(s) . This option makes it possible to see the restriction sites on the sequence (see figure 23.12) and save the annotations for later use.



Figure 23.12: The result of the restriction analysis shown as annotations.

Create restriction map . When a restriction map is created, it can be shown in three different ways:


- As a **table of restriction sites** as shown in figure 23.13. If more than one sequence were selected, the table will include the restriction sites of all the sequences. This makes it easy to compare the result of the restriction map analysis for two sequences.

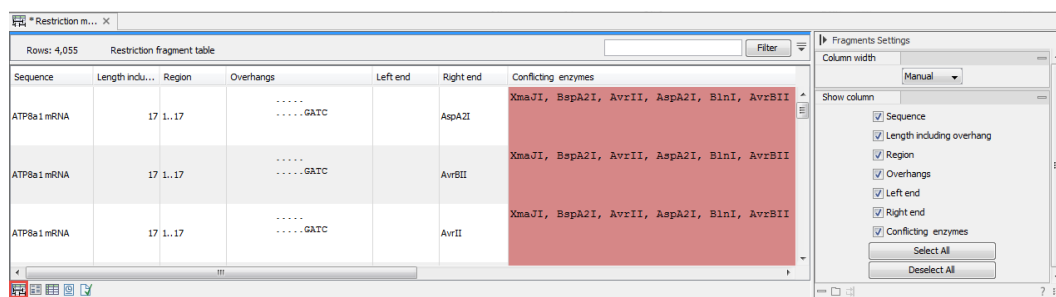
Sequence	Name	Pattern	Length	Overhang	Number of c...	Cut position(s)
ATP8a1 mRNA	AagI	atcgat	6 5'	1	3895	
ATP8a1 mRNA	AarI	cacctgc	7 5'	1	212	
ATP8a1 mRNA	AasI	gacnnnnngtc	12 3'	1	6531	
ATP8a1 mRNA	AatI	agcct	6 Blunt	2	3679, 6393	
ATP8a1 mRNA	AatII	gacgtc	6 3'	1	2593	
ATP8a1 mRNA	AbaI	tgatca	6 5'	2	403, 4981	
ATP8a1 mRNA	AbeI	cctcagc	7 5'	2	352, [1986]	
ATP8a1 mRNA	Acc113I	agtact	6 Blunt	1	386	
ATP8a1 mRNA	Acc65I	ggatcc	6 5'	1	1204	
ATP8a1 mRNA	Acc65I	ggatcc	6 5'	2	2221, 5899	
ATP8a1 mRNA	AccIII	tcgga	6 5'	2	3291, 4074	
ATP8a1 mRNA	AdI	aacgtt	6 5'	2	4495, 6208	
ATP8a1 mRNA	AcrII	ggtnacc	7 5'	1	5633	
ATP8a1 mRNA	AcvI	cacgtg	6 Blunt	1	3530	

Figure 23.13: The result of the restriction analysis shown as a table of restriction sites.

Each row in the table represents a restriction enzyme. The following information is available for each enzyme:

- **Sequence.** The name of the sequence which is relevant if you have performed restriction map analysis on more than one sequence.

- **Name.** The name of the enzyme.
- **Pattern.** The recognition sequence of the enzyme.
- **Length.** the restriction site length.
- **Overhang.** The overhang produced by cutting with the enzyme (3', 5' or Blunt).
- **Number of cut sites.**
- **Cut position(s).** The position of each cut.
 - * **[]** If the enzyme's recognition sequence is on the negative strand, the cut position is put in brackets.
 - * **()** Some enzymes cut the sequence twice for each recognition site, and in this case the two cut positions are surrounded by parentheses.
- As a **table of fragments** which shows the sequence fragments that would be the result of cutting the sequence with the selected enzymes (see figure 23.14). Click the Fragments button () at the bottom of the view.



The screenshot shows a software window titled "Restriction m...". It contains a table with 4,055 rows and several columns. The columns are: Sequence, Length including overhang, Region, Overhangs, Left end, Right end, and Conflicting enzymes. The first three rows of the table are highlighted in red. The first row shows "ATP8a1 mRNA" with a length of 17, region 1..17, and overhangs ".....GATC". The left end is "AspA2I" and the right end is "AspA2I". The conflicting enzymes are "XmaJI, BspA2I, AvrII, AspA2I, BlnI, AvrBII". The second row shows "ATP8a1 mRNA" with a length of 17, region 1..17, and overhangs ".....GATC". The left end is "AvrBII" and the right end is "AvrBII". The conflicting enzymes are "XmaJI, BspA2I, AvrII, AspA2I, BlnI, AvrBII". The third row shows "ATP8a1 mRNA" with a length of 17, region 1..17, and overhangs ".....GATC". The left end is "AvrII" and the right end is "AvrII". The conflicting enzymes are "XmaJI, BspA2I, AvrII, AspA2I, BlnI, AvrBII". To the right of the table is a "Fragments Settings" panel with a "Column width" dropdown set to "Manual". Below this is a "Show column" section with checkboxes for "Sequence", "Length including overhang", "Region", "Overhang", "Left end", "Right end", and "Conflicting enzymes". All these checkboxes are checked. At the bottom of the settings panel are "Select All" and "Deselect All" buttons.

Sequence	Length including overhang	Region	Overhangs	Left end	Right end	Conflicting enzymes
ATP8a1 mRNA	17	1..17GATC	AspA2I	AspA2I	XmaJI, BspA2I, AvrII, AspA2I, BlnI, AvrBII
ATP8a1 mRNA	17	1..17GATC	AvrBII	AvrBII	XmaJI, BspA2I, AvrII, AspA2I, BlnI, AvrBII
ATP8a1 mRNA	17	1..17GATC	AvrII	AvrII	XmaJI, BspA2I, AvrII, AspA2I, BlnI, AvrBII

Figure 23.14: The result of the restriction analysis shown as table of fragments.

Each row in the table represents a fragment. If more than one enzyme cuts in the same region, or if an enzyme's recognition site is cut by another enzyme, there will be a fragment for each of the possible cut combinations. Furthermore, if this is the case, you will see the names of the other enzymes in the **Conflicting Enzymes** column.

The following information is available for each fragment.

- **Sequence.** The name of the sequence which is relevant if you have performed restriction map analysis on more than one sequence.
- **Length including overhang.** The length of the fragment. If there are overhangs of the fragment, these are included in the length (both 3' and 5' overhangs).
- **Region.** The fragment's region on the original sequence.
- **Overhangs.** If there is an overhang, this is displayed with an abbreviated version of the fragment and its overhangs. The two rows of dots (.) represent the two strands of the fragment and the overhang is visualized on each side of the dots with the residue(s) that make up the overhang. If there are only the two rows of dots, it means that there is no overhang.
- **Left end.** The enzyme that cuts the fragment to the left (5' end).
- **Right end.** The enzyme that cuts the fragment to the right (3' end).
- **Conflicting enzymes.** If more than one enzyme cuts at the same position, or if an enzyme's recognition site is cut by another enzyme, a fragment is displayed for each possible combination of cuts. At the same time, this column will display the enzymes

that are in conflict. If there are conflicting enzymes, they will be colored red to alert the user. If the same experiment were performed in the lab, conflicting enzymes could lead to wrong results. For this reason, this functionality is useful to simulate digestions with complex combinations of restriction enzymes.

If views of both the fragment table and the sequence are open, clicking in the fragment table will select the corresponding region on the sequence.

- As a **virtual gel** simulation which shows the fragments as bands on a gel (see figure 23.48). For more information about gel electrophoresis, see section 23.6.

23.1.3 Insert restriction site

If you right-click on a selected region of a sequence, you find this option for inserting the recognition sequence of a restriction enzyme before or after the region you selected. This will display a dialog as shown in figure 23.15.

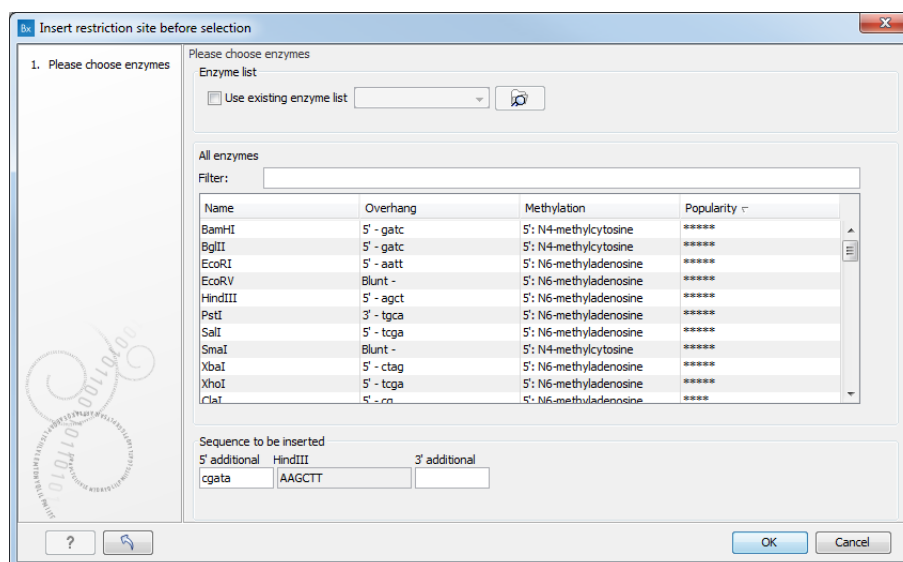


Figure 23.15: Inserting a restriction site and potentially a recognition sequence.

At the top, you can select an existing enzyme list or you can use the full list of enzymes (default). Select an enzyme, and you will see its recognition sequence in the text field below the list (AAGCTT). If you wish to insert additional residues such as tags, this can be typed into the text fields adjacent to the recognition sequence.

Click **OK** will insert the restriction site and the tag(s) before or after the selection. If the enzyme selected was not already present in the list in the **Side Panel**, it will now be added and selected.

23.2 Restriction enzyme lists

CLC Genomics Workbench includes restriction enzymes available in the **REBASE** database, with methylation shown as performed by the cognate methylase rather than by Dam/Dcm. If you want to customize the enzyme database for your installation, see section E. However, when performing restriction site analyses, it is often an advantage to use a customized list of enzymes. In this case, the user can create special lists containing for example all enzymes available in the

laboratory freezer, or all enzymes used to create a given restriction map or all enzymes that are available from the preferred vendor.

In the Example data (import in your Navigation Area using the Help menu), under Nucleotide->Restriction analysis, there are two enzyme lists: one with the 50 most popular enzymes, and another with all enzymes that are included in the *CLC Genomics Workbench*.

Create enzyme list *CLC Genomics Workbench* uses enzymes from the **REBASE** restriction enzyme database at <http://rebase.neb.com>. If you want to customize the enzyme database for your installation, see section E.

To create an enzyme list of a subset of these enzymes:

File | New | Enzyme list 

This opens the dialog shown in figure 23.16

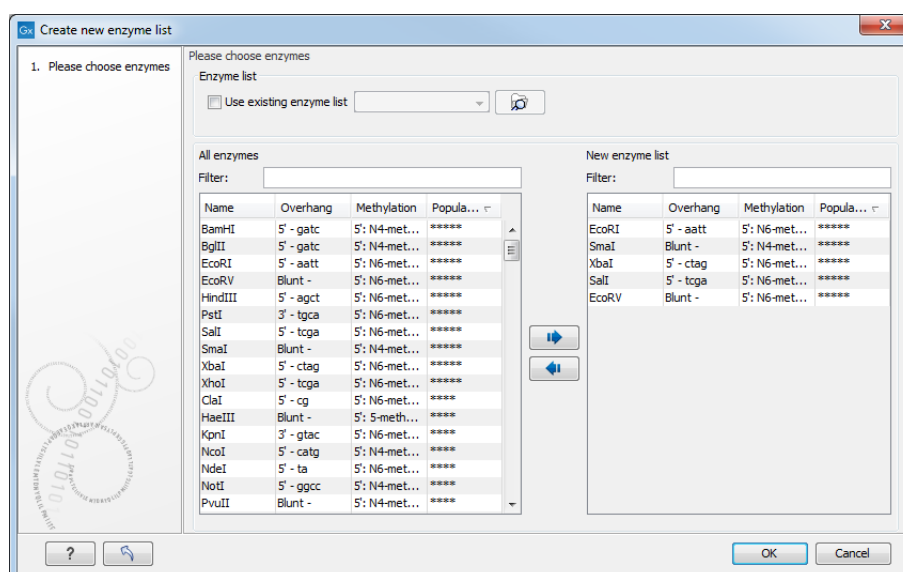


Figure 23.16: Choosing enzymes for the new enzyme list.

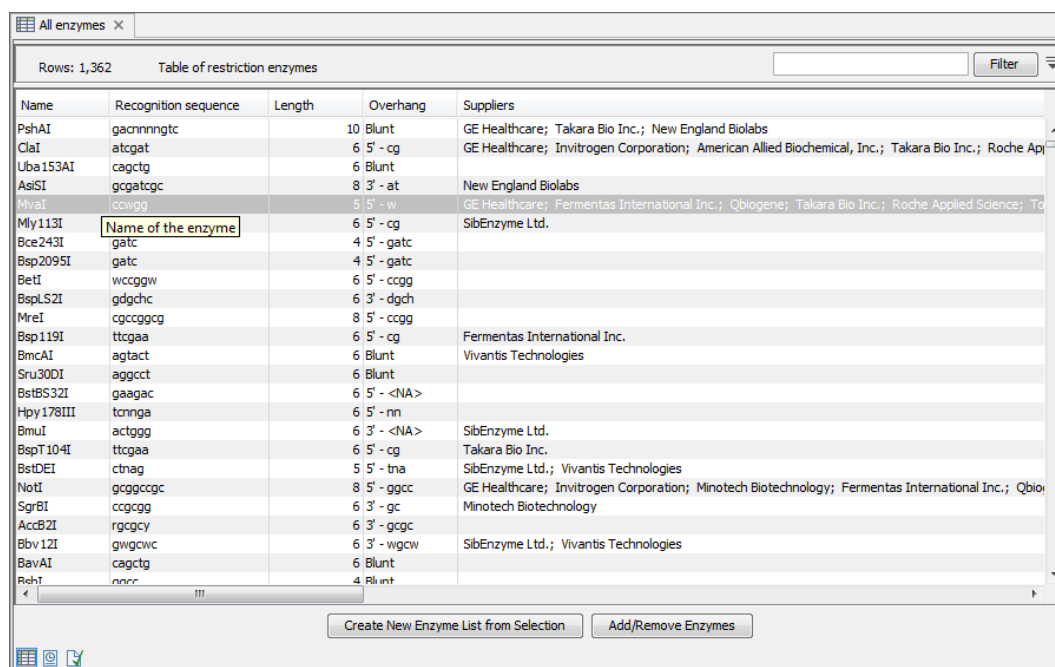
Choose which enzyme you want to include in the new enzyme list (see section 23.1.1), and click **Finish** to open the enzyme list.

View and modify enzyme list An enzyme list is shown in figure 23.17. It can be sorted by clicking the columns, and you can use the filter at the top right corner to search for specific enzymes, recognition sequences etc.

If you wish to remove or add enzymes, click the **Add/Remove Enzymes** button at the bottom of the view. This will present the same dialog as shown in figure 23.16 with the enzyme list shown to the right.

If you wish to extract a subset of an enzyme list, open the list, select the relevant enzymes, right-click on the selection and choose to **Create New Enzyme List from Selection** .

If you combined this method with the filter located at the top of the view, you can extract a very specific set of enzymes. for example, if you wish to create a list of enzymes sold by a particular



Name	Recognition sequence	Length	Overhang	Suppliers
PshAI	gacnngtc	10	Blunt	GE Healthcare; Takara Bio Inc.; New England Biolabs
ClaI	atcgat	6	5' - cg	GE Healthcare; Invitrogen Corporation; American Allied Biochemical, Inc.; Takara Bio Inc.; Roche Applied Science
Uba153AI	cagctg	6	Blunt	
AsiSI	cgatcg	8	3' - at	New England Biolabs
MvaI	ccwgg	5	5' - w	GE Healthcare; Fermentas International Inc.; Qbiogene; Takara Bio Inc.; Roche Applied Science; ThermoFisher Scientific
Mly113I	gacgac	6	5' - cg	SibEnzyme Ltd.
Bce243I	gatc	4	5' - gatc	
Bsp2095I	gatc	4	5' - gatc	
BetI	wcggw	6	5' - ccgg	
BspLS2I	gdgch	6	3' - dgch	
MreI	cccgccg	8	5' - ccgg	
Bsp119I	ttcgaa	6	5' - cg	Fermentas International Inc.
BmcAI	agtact	6	Blunt	Vivantis Technologies
Sru300I	agcct	6	Blunt	
BstBS32I	gaagac	6	5' - <NA>	
Hpy178III	tcnnga	6	5' - nn	
BmuI	actggg	6	3' - <NA>	SibEnzyme Ltd.
BspT104I	ttcgaa	6	5' - cg	Takara Bio Inc.
BstDEI	ctnag	5	5' - tna	SibEnzyme Ltd.; Vivantis Technologies
NciI	gcggccgc	8	5' - gccc	GE Healthcare; Invitrogen Corporation; Minotech Biotechnology; Fermentas International Inc.; Qbiogene
SgrBI	ccggg	6	3' - gc	Minotech Biotechnology
AccB2I	rgcgcy	6	3' - ggc	
Bbv12I	gwgcw	6	3' - wgcw	SibEnzyme Ltd.; Vivantis Technologies
BavAI	cagctg	6	Blunt	
PstI	ccgga	6	3' - gg	

Figure 23.17: An enzyme list.

distributor, type the name of the distributor into the filter and select and create a new enzyme list from the selection.

23.3 Restriction Based Cloning

The *in silico* cloning process in *CLC Genomics Workbench* begins with the Restriction Based Cloning tool:

Molecular Biology Tools (🔧) | Cloning (🧬) | Restriction Based Cloning (🔍)

This will open a dialog where you can select both the sequences containing the fragments you want to clone, as well as the one to be used as vector (figure 23.18).

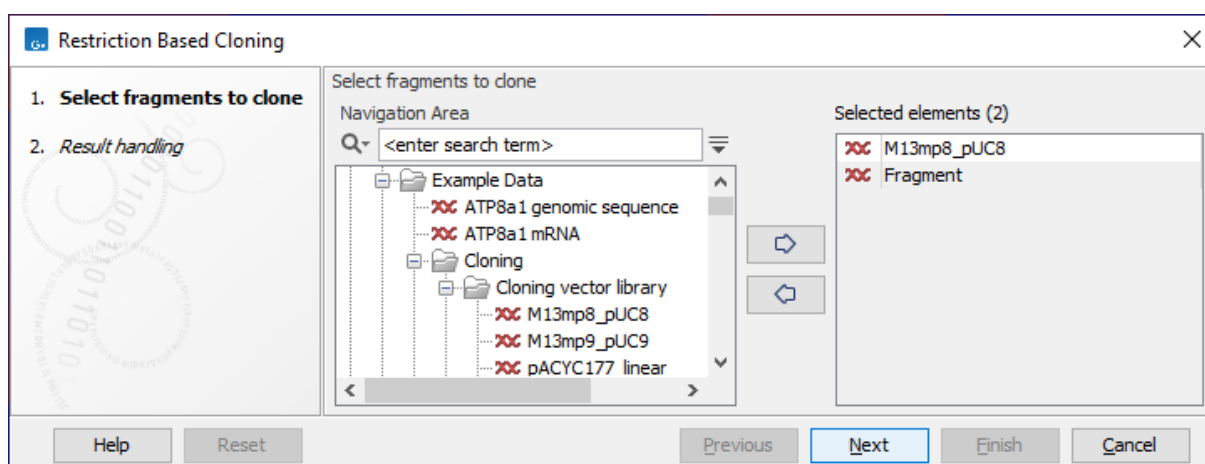


Figure 23.18: Selecting the sequences containing the fragments you want to clone and the vector.

CLC Genomics Workbench will now create a sequence list of the selected fragments and vector

sequences. For cloning work, open the sequence list and switch to the **Cloning Editor** (🔗) at the bottom of the view (figure 23.19).

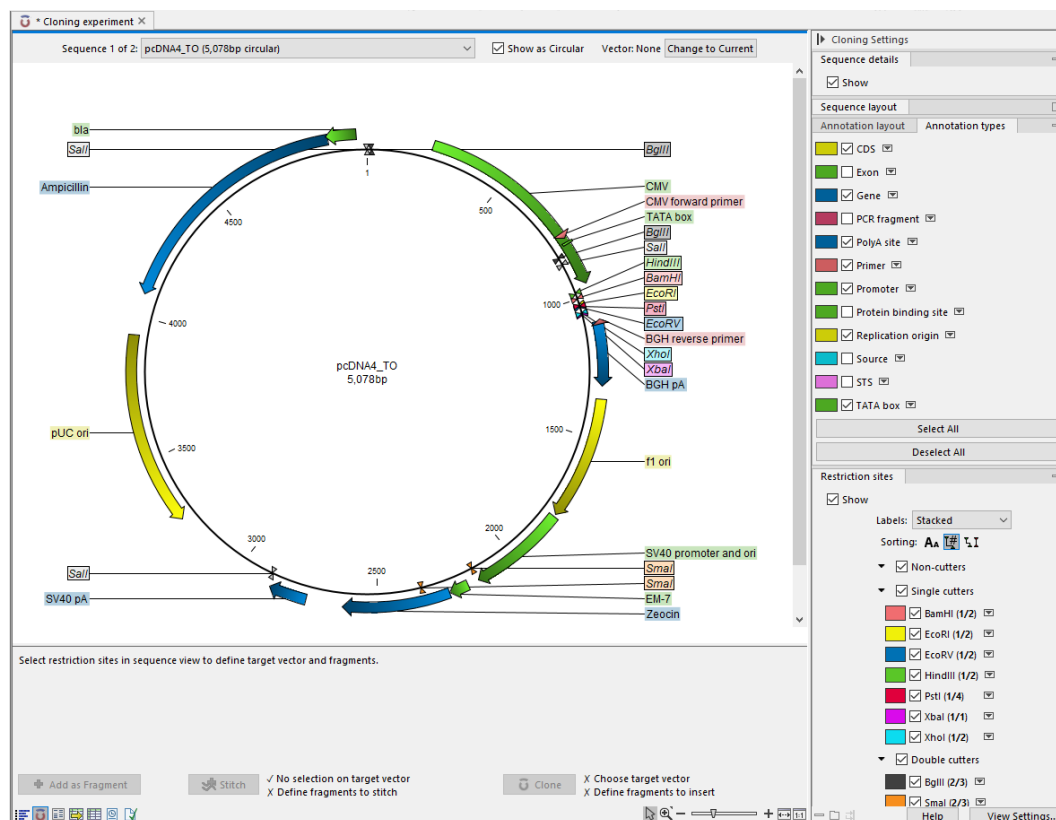


Figure 23.19: Cloning editor view of the sequence list. Choose which sequence to display from the drop down menu.

If you later in the process need additional sequences, right-click anywhere on the empty white area of the view and choose to "Add Sequences".



23.3.1 Introduction to the Cloning Editor

In the Cloning Editor, most of the basic options for viewing, selecting and zooming the sequences are the same as for the standard sequence view (section 15.2). In particular, this means that annotations can be displayed on the sequences to guide the choice of regions to clone.

However, the Cloning Editor has a special layout with three distinct areas (in addition to the **Side Panel** found in other sequence views as well):

- At the top, there is a panel to switch between the sequences selected as input for the cloning. You can also specify whether the sequence should be visualized **as circular** or as a fragment. On the right-hand side, you can select a vector: the button is by default set to **Change to Current**. Click on it to select the currently shown sequence as **vector**.
- In the middle, the selected sequence is shown. This is the central area for defining how the cloning should be performed.
- At the bottom, there is a panel where the selection of fragments and target vector is performed.

The Cloning Editor can be opened in the following ways:

- Click on the **Cloning Editor** icon () in the view area when a sequence list has been opened in the sequence list editor.
- Create a new cloning experiment using the **Restriction Based Cloning** () action from the toolbox. This tool collects a set of existing sequences and creates a new sequence list.

The Cloning Editor can be used in two ways:

- **Cloning mode** Opened when one of the sequences has been selected as 'Vector'. In this mode, you can apply one or more cuts to the vector, thereby creating an opening for insertion of other sequence fragments. From the remaining sequences in the cloning experiment/sequence list, either complete sequences or fragments created by cutting can be inserted into the vector. In the cloning adapter dialog, the order and direction of the inserted fragments can be adjusted prior to adjusting the overhangs to match the cloning conditions.
- **Stitch mode** If no sequence has been selected as 'Vector', a number of fragments (either full sequences or cuttings) can be selected from the cloning experiment. These can then be stitched together into a single new sequence. In the stitching adapter dialog, the order and direction of the fragments can be adjusted prior to adjusting the overhangs to match the stitch conditions.

23.3.2 The restriction cloning workflow


The restriction cloning workflow consists of the following steps:


1. Define one or more fragments

First, select the sequence containing the cloning fragment in the list at the top of the view. Next, make sure the restriction enzyme you wish to use is listed in the **Side Panel** (see section 23.1.1). To specify which part of the sequence should be treated as the fragment, first click one of the cut sites you wish to use. Then press and hold the Ctrl key (⌘ on Mac) while you click the second cut site. You can also right-click the cut sites and use the **Select This ... Site** to select a site. If you just wish to remove the selection of one of the sites, right-click the site on the sequence and choose **De-select This ... Site**.

When this is done, the panel is updated to reflect the selections (see figure 23.20).

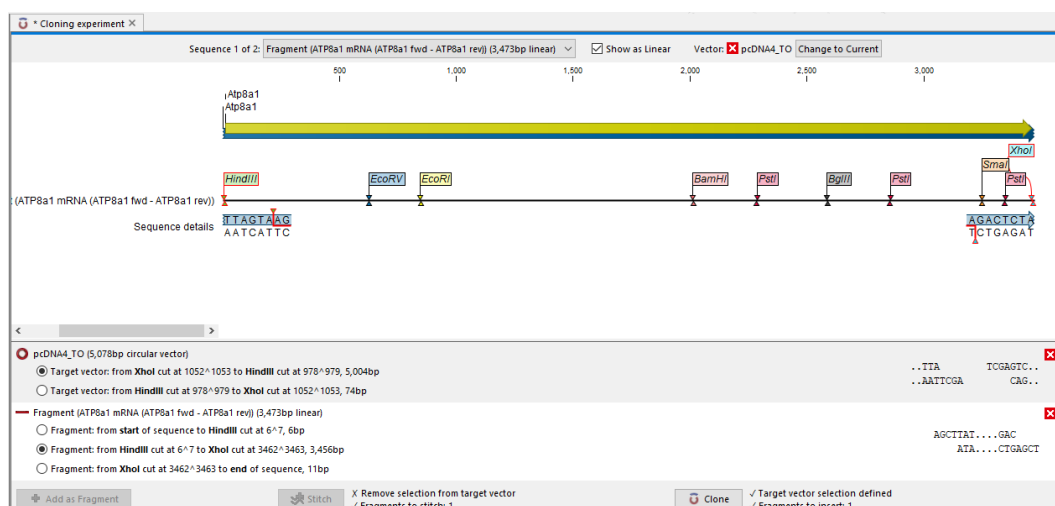
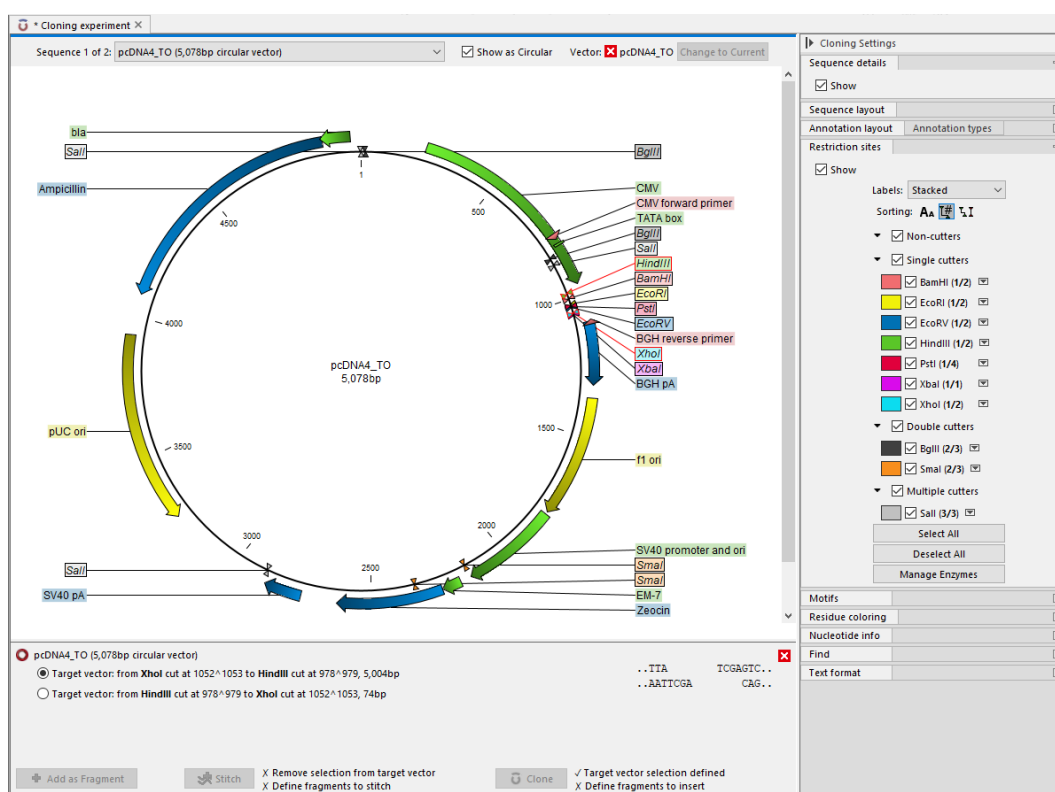
In this example you can see that there are now three fragments that can be used for cloning listed in the panel below the view. The fragment selected per default is the one that is in between the cut sites selected.

If the entire sequence should be selected as fragment, click **Add as Fragment** (.

At any time, the selection of cut sites can be cleared by clicking the **Remove** () icon to the right of the target vector selections.

2. Defining target vector

The next step is to define where the vector should be cut. If the vector sequence should just be opened, click the restriction site you want to use for opening (figure 23.21).

Figure 23.20: *EcoRI* cut sites selected to cut out fragment.Figure 23.21: *EcoRI* site used to open the vector. Note that the "Cloning" button has now been enabled as both criteria ("Target vector selection defined" and "Fragments to insert:...") have been defined.

If you want to cut off part of the vector, click two restriction sites while pressing the Ctrl key (⌘ on Mac). You can also right-click the cut sites and use the **Select This ... Site** to select a site. This will display two options for what the target vector should be (for linear vectors there would have been three option). At any time, the selection of cut sites can be cleared by clicking the **Remove** (✖) icon to the right of the target vector selections.

3. Perform cloning

Once both fragments and vector are selected, click **Clone** (🔄). This will display a dialog to adapt overhangs and change orientation as shown in figure 23.22.

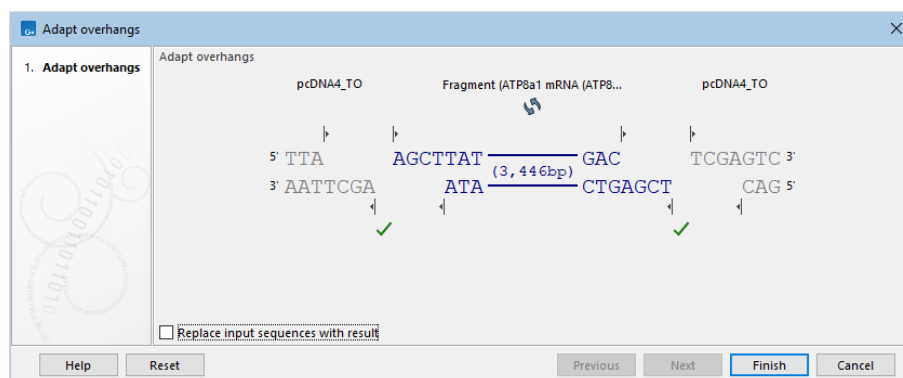


Figure 23.22: Showing the insertion point of the vector.

This dialog visualizes the details of the insertion. The vector sequence is on each side shown in a faded gray color. In the middle the fragment is displayed. If the overhangs of the sequence and the vector do not match (❌), you will not be able to click **Finish**. But you can blunt end or fill in the overhangs using the **drag handles** (⏏) until the overhangs match (✅).

The fragment can be reverse complemented by clicking the **Reverse complement fragment** (↺).

When several fragments are used, the order of the fragments can be changed by clicking the move buttons (➡)/ (⬅).

Per default, the construct will be opened in a new view and can be saved separately. But selecting the option **Replace input sequences with result** will add the construct to the input sequence list and delete the original fragment and vector sequences.

Note that the cloning experiment used to design the construct can be saved as well. If you check the **History** (📄) of the construct, you can see the details about restriction sites and fragments used for the cloning.

23.3.3 Manual cloning

Cloning steps can be done manually. For this, create a sequence list using **Restriction Based Cloning** and use the cloning actions provided via right-click menus. The general workflow is the same as that described in section 23.3.2, but carried out manually.

Manipulate the whole sequence

Right-click the sequence label to the left to see the menu shown in figure 23.23.

- **Duplicate sequence.** Adds a duplicate of the selected sequence to the sequence list accessible from the drop down menu on top of the Cloning view.
- **Insert sequence after this sequence** (➡). The sequence to be inserted can be selected from the sequence list via the drop down menu on top of the Cloning view. The inserted

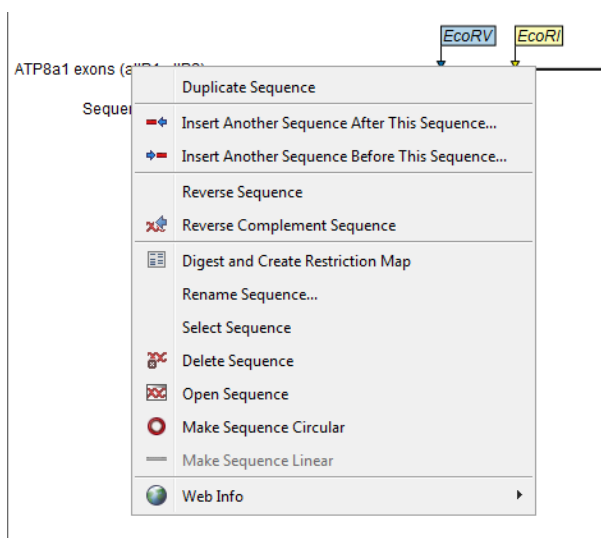


Figure 23.23: Right click on the sequence in the cloning view.

sequence remains on the list of sequences. If the two sequences do not have blunt ends, the ends' overhangs have to match each other.

- **Insert sequence before this sequence** (➡➡). The sequence to be inserted can be selected from the sequence list via the drop down menu on top of the Cloning view. The inserted sequence remains on the list of sequences. If the two sequences do not have blunt ends, the ends' overhangs have to match each other.
- **Reverse sequence**. Reverses the sequence and replaces the original sequence in the list. This is sometimes useful when working with single stranded sequences. Note that this is *not* the same as creating the reverse complement of a sequence.
- **Reverse complement sequence** (↔). Creates the reverse complement of a sequence and replaces the original sequence in the list. This is useful if the vector and the insert sequences are not oriented the same way.
- **Digest and Create Restriction Map** (📋). See section 23.6
- **Rename sequence**. Renames the sequence.
- **Select sequence**. Selects the entire sequence.
- **Delete sequence** (🗑️). Deletes the given sequence from the Cloning Editor.
- **Open sequence** (📄). Opens the selected sequence in a normal sequence view.
- **Make sequence circular** (🔄). Converts a sequence from a linear to a circular form. If the sequence have matching overhangs at the ends, they will be merged together. If the sequence have incompatible overhangs, a dialog is displayed, and the sequence cannot be made circular. The circular form is represented by >> and << at the ends of the sequence.
- **Make sequence linear** (—). Converts a sequence from a circular to a linear form, removing the << and >> at the ends.

Manipulate parts of the sequence

Right-click on a selected region of the sequence to see the menu shown in figure 23.24.

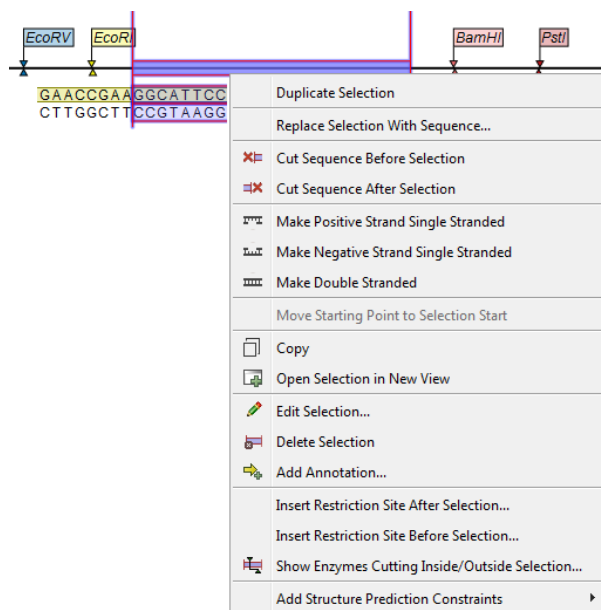






Figure 23.24: Right click on a sequence selection in the cloning view.

- **Duplicate Selection.** If a selection on the sequence is duplicated, the selected region will be added as a new sequence to the Cloning Editor. The new sequence name representing the length of the fragment. When double-clicking on a sequence, the region between the two closest restriction sites is automatically selected.
- **Replace Selection with sequence.** Replaces the selected region with a sequence selected from the drop down menu listing all sequences in the Cloning Editor.
- **Cut Sequence Before Selection** (✂). Cleaves the sequence before the selection and will result in two smaller fragments.
- **Cut Sequence After Selection** (✂). Cleaves the sequence after the selection and will result in two smaller fragments.
- **Make Positive Strand Single Stranded** (☰). Makes the positive strand of the selected region single stranded.
- **Make Negative Strand Single Stranded** (☷). Makes the negative strand of the selected region single stranded.
- **Make Double Stranded** (☰☷). This will make the selected region double stranded.
- **Move Starting Point to Selection Start.** This is only active for circular sequences. It will move the starting point of the sequence to the beginning of the selection.
- **Copy** (📄). Copies the selected region to the clipboard, which will enable it for use in other programs.
- **Open Selection in New View** (📄+). Opens the selected region in the normal sequence view.

- **Edit Selection** (). Opens a dialog box in which it is possible to edit the selected residues.
- **Delete Selection** (). Deletes the selected region of the sequence.
- **Add Annotation** (). Opens the **Add annotation** dialog box.
- **Insert Restriction Sites After/Before Selection.** Shows a dialog where you can choose from a list restriction enzymes (see section 23.1.3).
- **Show Enzymes Cutting Inside/Outside Selection** (). Adds enzymes cutting this selection to the Side Panel.
- **Add Structure Prediction Constraints.** This is relevant for RNA secondary structure prediction:
 - **Force Stem Here** is activated after choosing 2 regions of equal length on the sequence. It will add an annotation labeled "Forced Stem" and will force the algorithm to compute minimum free energy and structure with a stem in the selected region.
 - **Prohibit Stem Here** is activated after choosing 2 regions of equal length on the sequence. It will add an annotation labeled "Prohibited Stem" to the sequence and will force the algorithm to compute minimum free energy and structure without a stem in the selected region.
 - **Prohibit From Forming Base Pairs** will add an annotation labeled "No base pairs" to the sequence, and will force the algorithm to compute minimum free energy and structure without a base pair containing any residues in the selected region.

Insert one sequence into another

Sequences can be inserted into each other in various ways as described in the lists above. When you choose to insert one sequence into another, you will be presented with a dialog where all sequences in the sequence list are present (see figure 23.25).

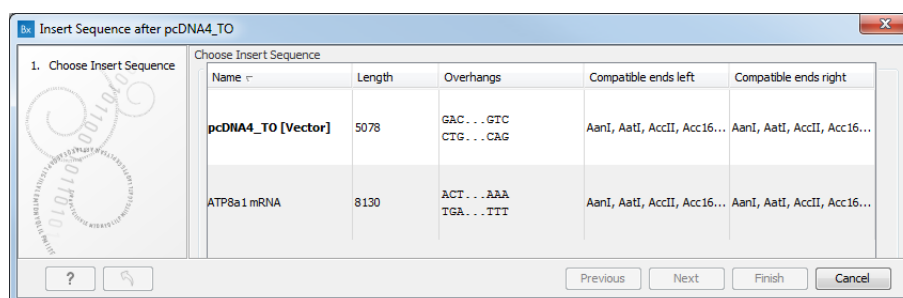


Figure 23.25: Select a sequence for insertion.

The sequence that you have chosen to insert into will be marked with **bold** and the text **[vector]** is appended to the sequence name. Note that this is completely unrelated to the vector concept in the cloning workflow described in section 23.3.2.

Furthermore, the list includes the length of the fragment, an indication of the overhangs, and a list of enzymes that are compatible with this overhang (for the left and right ends, respectively). If not all the enzymes can be shown, place your mouse cursor on the enzymes, and a full list will be shown in the tool tip.

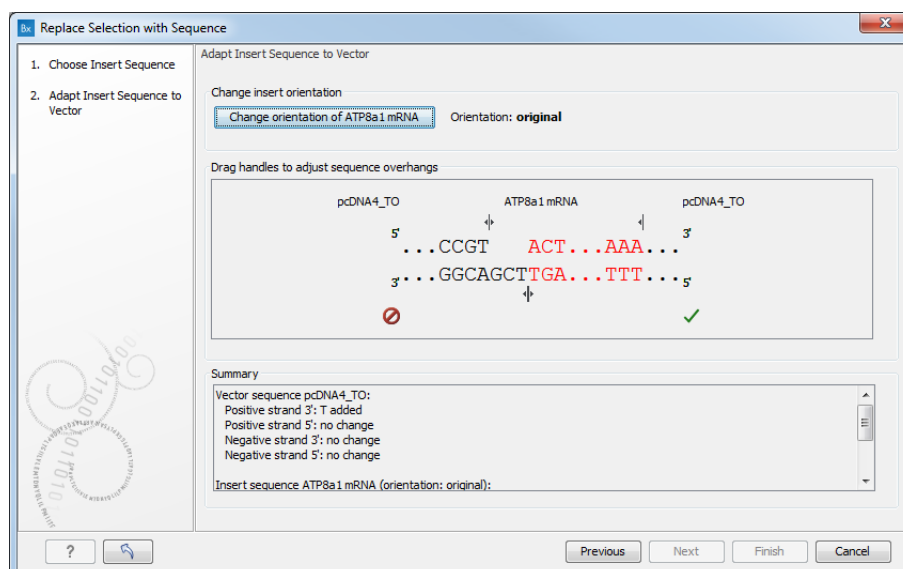


Figure 23.26: Drag the handles to adjust overhangs.

Select the sequence you wish to insert and click **Next** to adapt insert sequence to vector dialog (figure 23.26).

At the top is a button to reverse complement the inserted sequence.

Below is a visualization of the insertion details. The inserted sequence is at the middle shown in red, and the vector has been split at the insertion point and the ends are shown at each side of the inserted sequence.

If the overhangs of the sequence and the vector do not match (⊘), you can blunt end or fill in the overhangs using the **drag handles** (⊞) until it does (✓).

At the bottom of the dialog is a summary field which records all the changes made to the overhangs. This contents of the summary will also be written in the history (📄) of the cloning experiment.

When you click **Finish**, the sequence is inserted and highlighted by being selected.

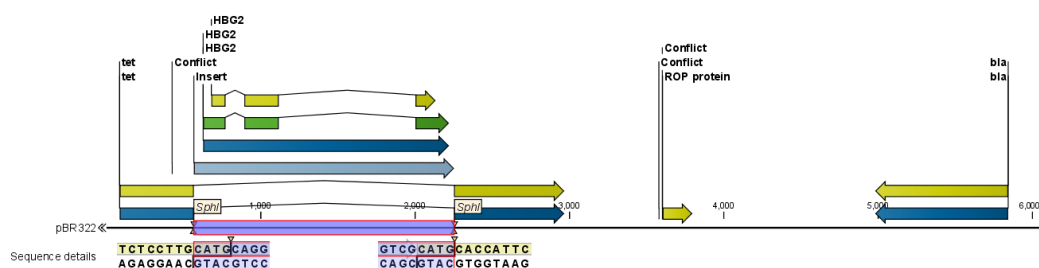


Figure 23.27: One sequence is now inserted into the cloning vector. The sequence inserted is automatically selected.

23.4 Homology Based Cloning

Using **Homology Based Cloning**, cloning experiments can be designed for methods such as Takara In-Fusion® and other ligation independent cloning techniques and Gibson Assembly®,

where sequences with homologous ends need to be assembled.

The tool generates primers with overhangs so that appropriate homologous sequence is added to fragments that should be assembled. Subsequently, primers and overhangs can be inspected and adjusted.

Running the Homology Based Cloning tool

To run the tool, go to:

Molecular Biology Tools (🔧) | **Cloning** (🧬) | **Homology Based Cloning** (🔄)

In the first dialog, select the vector and all the DNA fragments that should be assembled in the cloning reaction (figure 23.28).

The first sequence used as input, will be considered the vector, but the order of sequences can be adjusted in the following wizard step.

You can select 1-50 individual sequences with a maximal combined length of 100,000 base pairs as input to the tool. However, the number of sequences selected should be in accordance with the relevant laboratory protocol.

Note that it is also possible to use an assembled vector that was previously generated by **Homology Based Cloning** as input. In that case, previously designed primers and overhangs will be shown in the following wizard step.

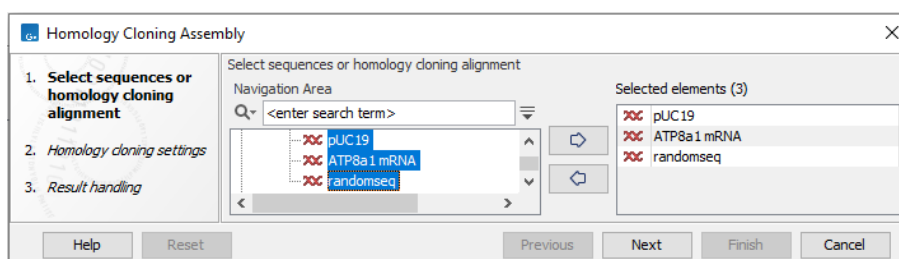


Figure 23.28: Select the vector and fragments that should be assembled in the homology based cloning reaction.

Press **Next** to open the wizard allowing you to inspect and adjust primers and overhangs.

23.4.1 Working with homology based cloning

This section gives a quick overview of the Homology Based Cloning wizard. For a full description, see section 23.4.4.

General options

General options are at the top of the wizard. These include the position of the insertion site in the vector, the maximum primer and overhang lengths as well as option to set the T_m and overhang length for all primers at once. There is also a diagram of the vector including the inserts, where each sequence has a different colour (figure 23.29).

Sequences

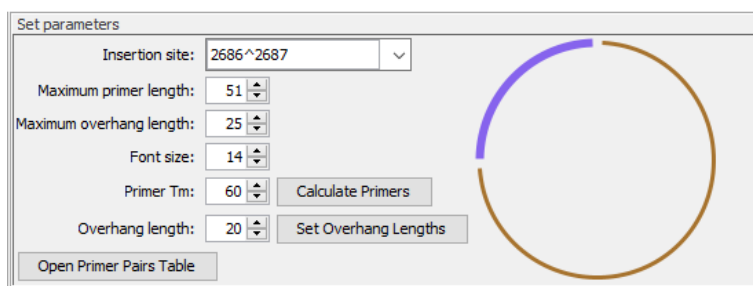


Figure 23.29: The top section of the wizard contains general options.

Each sequence is displayed individually, with a coloured bar to the left and a vertical scroll bar at the bottom. The top sequence is the vector, with the insert sequences displayed further down.

The order of the sequences reflects how they will be assembled into the vector, and the overhangs on the primers support this assembly order.

Vector, inserts, primers and overhangs are color coded (figure 23.30):

- **Black** Vector or insert sequence
- **Grey** Vector or insert sequence not included in the PCR product
- **Brown** Primer sequence
- **Pink** Overhang sequence
- **Blue** Added bases that are inserted between primer and overhang

For each sequence, you can adjust primer and overhang lengths and add bases between primers and overhangs.

The vector sequence is considered circular and primers are depicted as pointing away from each other in order to amplify the circular sequence. Inserts are considered linear, and primers are placed at the ends of the insert sequence pointing towards each other in order to amplify the linear sequence (figure 23.30).

23.4.2 Adjust the homology based cloning design

This section describes how primers and overhangs can be adjusted and inspected.

The overhangs added by **Homology Based Cloning** are 20 base pairs long and are added to vector or insert primers:

- If one insert is assembled into a vector < 8 kb in length, overhangs are added to the vector primers.
- If one insert is assembled into a vector > 8 kb in length, overhangs are added to the insert primers.
- If more inserts are assembled into a vector, the overhangs are added to insert primers.

Primer and overhang lengths should be adjusted, according to the cloning kit used.

Sequence name: **pUC19 (Vector, Circular)**

Reverse primer: Primer length: 17, Tm: 60 °C, Overhang length: 20, Added bases: ATCGAT

Forward primer: Primer length: 17, Tm: 63 °C, Overhang length: 20, Added bases:

Primer pair annealing alignment:

CGCAACCGGCTCTCCC
|||
TCTTCGGCTTCTCGCT

..ACCGAGCGCAGCGAGTCAGTGAGCGAGGAAGCGGAAGA
TCGCTCCTTCGCCTTCT ATCGATTGACGCCCTCTGGATCCGC
GCGCCCAATA CGCAACCGGCTCTCCC GCGGTGGCCGATTACATTA..
AACTTTACCATACATAA CGCAACCGGCTCTCCC

Sequence name: **ATP8a1 mRNA (Insert 1, Linear)**

Reverse primer: Primer length: 24, Tm: 50 °C, Overhang length: 20, Added bases:

Forward primer: Primer length: 23, Tm: 72 °C, Overhang length: 18, Added bases: TAGCTA

Primer pair annealing alignment:

ACTGCGGGGAGACCTAGGCGGCT
|||
TTTATGTAATGGTAAAGTTTATT

GAGCGAGGAAGCGGAAGATAGCTA ACTGCGGGGAGACCTAGGCGGCTCTGCGGACGCAGCTC...GTTATTAATAAATAAAATAAACTTTACCATACATAAA
TTTATTGAAATGGTAAATGTTATT GCGTTTGGCGGAG

Figure 23.30: Top: The vector sequence and primers with overhangs. The grey sequence between the primers is not included in the PCR product. Bottom: An insert sequence and primers with overhangs.

Change the insertion site

Change the insertion site in the vector by typing a position or a range of positions directly into **Insertion site** text field. Alternatively, choose the start, end or complete range of an annotation from the drop down menu (figure 23.31). If you specify a range of bases, primers will be placed so that bases are replaced by the insert(s).

Set parameters

Insertion site: 1..10

Maximum primer length: 2686^2687: genomic DNA

Maximum overhang length: 232^233: polylinker o... (+

Font size: 233..289: polylinker o... (+

Primer Tm: 289^290: polylinker o... (+

Overhang length: 289^290: Lac-Operon (+ M

290..447: Lac-Operon (+ M

447^448: Lac-Operon (+ M

447^448: pBR322 (- Misc. t

Open Primer Pairs Table

Figure 23.31: Choose an insertion site from the drop down menu or type position(s) directly in the Insertion site text field.

Change the assembly order

Use the arrows to the left of the sequence names to move a sequence up or down in the order. The sequence of primer overhangs is automatically updated to reflect the new order. The length of overhangs is not changed.

Adjust the length of primers and overhangs

You can adjust primers and overhangs using the following options:

- To adjust all primers at once, change the **Primer Tm** in the top section and press **Calculate primers** (figure 23.29). This will update primers on all sequences.
- To adjust all overhangs at once, change the **Overhang length** in the top section and press **Set Overhang Lengths** (figure 23.29). This will update overhangs on all sequences.
- To adjust the length of individual primers and overhangs use the **Primer length** and **Overhang length** options available for the forward and reverse primer on each sequence. You can also extend or shorten the the primer and overhang sequences by dragging the **arrow symbols** at the ends of the primers and overhangs (figure 23.30).

Insert additional bases

Insert additional bases between the primer and the overhang by typing directly in to the **Added bases** text fields. You can also choose the sequence of a restriction site from the drop down menu.

Inspect primer pairs

The designed primer pairs can be closely inspected in the table that opens when pressing **Open Primer Pairs Table**. The table contains information about secondary structure for each primer pair, both with and without overhang. For a description of each of the columns in the Primer Pairs table see section 21.5.3. Close the table before returning to the wizard to make further adjustments or complete the design.

The outputs created by **Homology Based Cloning** are described in section 23.4.3.

23.4.3 Homology Based Cloning outputs

Homology Based Cloning can create the following outputs:

- **Report** A report containing the following sections:
 - **Summary** Contains the number of fragments and primers used in the cloning reaction as well as their lengths and any warnings.
 - **Fragments** Lists the vector and fragments used in the cloning reaction.
 - **Warnings** Lists the warnings given for primer pairs.
 - **Primer pairs** Lists fragments for which primers were designed together with pair annealing and pair end annealing values for the primer pairs. See section 21.5.3 for information about annealing values.
 - **Primers** Lists individual primers and their sequence. Primer sequence is written with capital letters, whereas added bases and overhangs are in lowercase.
 - **Primer parts** Lists full and subparts of designed primers with characteristics such as length and G/C content. The following terms are used:
 - * **Full** The full primer including overhang and added bases.
 - * **Anneal** The part of the primer annealing to the original fragment (primer without overhang or any added bases).

- * **Overhang** The overhang added to primers to generate homologous sequence between fragments.
- * **Added** The bases inserted between primer and overhang.
- **Assembled vector** The vector as it will appear after all fragments have been assembled. The assembled vector will be annotated with the positions of primers, added bases and overhangs as well as with inserts and vector sequence. When the vector is opened, you can select which annotations should be shown on the sequence in the side panel under Annotation types.
Note: the assembled vector can be used as input to **Homology Based Cloning** if you wish to adjust a previous design.
- **Primers sequence list** A sequence list containing the designed primers. The primers are annotated with primer, added bases and overhang, where primer is the part of the sequence that originally aligned to the insert or vector that was amplified.
- **PCR fragments sequence list** The PCR fragments generated from input sequences and designed primers including additional bases and overhangs.
- **Primer pairs table** A table providing information about melting temperatures, secondary structure, etc., for primer pairs with and without overhangs. For a description of each of the columns in the Primer Pairs table, see section 21.5.3.

23.4.4 Detailed description of the Homology Based Cloning wizard

This section contains a detailed description of the **Homology Based Cloning** options (figure 23.32).

- **Insertion site** The position where fragments will be inserted in the vector. You can type in a specific position or a range of positions. You can also choose the start, the end, or the entire span of an annotation on the vector using the drop down menu (figure 23.33).

The primers designed to amplify the vector will be placed so that their 5' ends are adjacent to the insertion site. If a range of positions are selected, the primers will be placed so that the selected positions are not included in the PCR product. When the insertion site is changed, the vector primers in the view below are updated accordingly.

Insertion site examples:

- 0 or 0[^]1 Assembles inserts into the vector between the last and the first base.
- 1 or 1[^]2 Assembles inserts into the vector between the first and second base.
- 1..10 Inserts replace bases 1-10 in the vector.
- **Start of an annotation** Assembles inserts into the vector before the first base in the annotated region.
- **Span of an annotation** Inserts replace all bases in the annotated region.
- **End of an annotation** Assembles inserts into the vector after the last base in the annotated region.

Set parameters

Insertion site: 2686^2687

Maximum primer length: 51

Maximum overhang length: 25

Font size: 14

Primer Tm: 60

Overhang length: 20

Calculate Primers

Set Overhang Lengths

Open Primer Pairs Table

Sequence name: pUC19 (Vector, Circular)

Reverse primer: Primer length: 17, Tm: 60 °C, Overhang length: 0, Added bases:

Forward primer: Primer length: 17, Tm: 60 °C, Overhang length: 0, Added bases:

Primer pair annealing alignment:

CGCCCCAATACGCAAAC
| | | | |
TCTTCGCTTCCTCGCT

TCGCTCCTTCGCTTCT

..CCGCAGCCGAACGACCGAGCGCAGTCAGTGAGCGAGGAAGCGGAAGA GCGCCCAATACGCAAACCGCTCTCCCGCGCGTTGGCC
GCGCCCAATACGCAAAC

Sequence name: ATP8a1 mRNA (Insert 1, Linear)

Reverse primer: Primer length: 46, Tm: 60 °C, Overhang length: 20, Added bases:

Forward primer: Primer length: 17, Tm: 59 °C, Overhang length: 20, Added bases:

Primer pair annealing alignment:

ACTG
| |
TTTATGTAATGGTAAAGTTTATTTATTTTAAAT

AGGATT

GTGAGCGAGGAAGCGGAAGA ACTGCGGGGAGACCTAG

Sequence name: randomseq (Insert 2, Linear)

Reverse primer: Primer length: 21, Tm: 61 °C, Overhang length: 20, Added bases:

Forward primer: Primer length: 21, Tm: 61 °C, Overhang length: 20, Added bases:

Primer pair annealing alignment:

GTAGTAGTGTCGCGTAGCTGA
| | | | |
TGCATCAGCTAGCTAGCTAGC

AAACTTTACCATTACATAAA GTAGTAGTGTCGCGTAGCTGA

Figure 23.32: General options for the cloning experiment are provided at the top of the wizard, followed by sections for the vector and insert sequences, where many options relevant to the cloning experiment can be adjusted.

- **Maximum primer length** The maximum length that primers for vectors and inserts can be. This is reflected in the number of nucleotides visible for each sequence in the views below.
- **Maximum overhang length** The maximum length that overhangs for vectors and inserts can be.

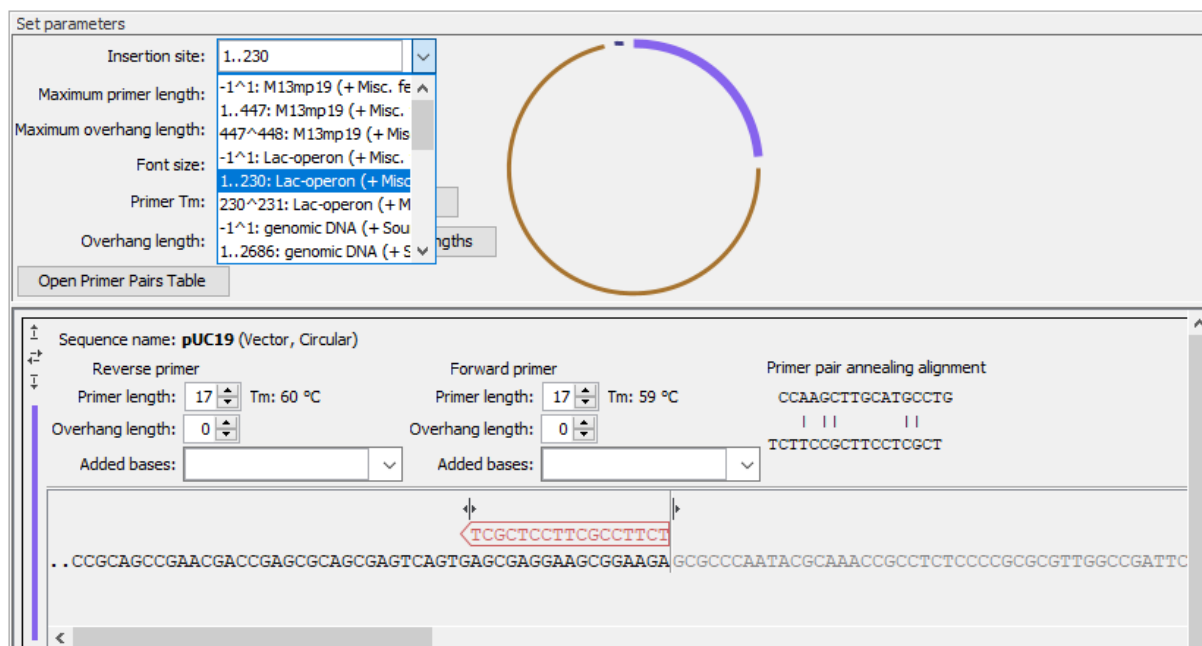


Figure 23.33: Specify the insertion site in the vector. Here the entire *Lac*-operon has been selected from the drop down menu. Notice that when a span of bases are chosen as insertion site, the vector sequence between the primers is grey and not included in the PCR product.

- **Font size** The font size to use for vector and insert sequences, and for primers and overhangs.
- **Primer Tm** The primer melting temperature. This value does not take into account any added bases or overhangs. Click on **Calculate primers** to update all primers after changing this value.
- **Overhang length** The length of the overhang added to primers not including added bases. Click on **Set Overhang Lengths** to update overhangs after changing this value.
- **Open Primer Pairs Table** Opens a table listing each of the primer pairs shown on the sequences below. The primer pairs table contains primer pairs, both with and without overhangs and added bases. It also provides information about melting temperatures, secondary structure, etc. For a description of each of the columns in the Primer Pairs table, see section 21.5.3.
- **Vector map** A vector map showing the assembled vector. Each original fragment has its own color that matches the side bars of the sequences in the views below. If you hover over the sequence of the vector or an insert, it will become bold in the vector map. If you hover over a primer, it will appear on the vector map. The fragments, but not the primers are drawn to scale.
- **Sequence Name: n (vector, circular)** and **Sequence Name: n (insert y, linear)**. The sequences identified as the vector and inserts.
- **Arrows to the left of Sequence Name** Change the order of the sequences in the list using the up and down arrows. Reverse complement the sequence using the horizontal arrows.
- **Primer length** and **Overhang length** The primer and overhang lengths for the forward and reverse primer, respectively. These lengths can be adjusted by typing new values into the

dialogs, or by using the up and down arrows to the right of the dialogs. Changes to the lengths are immediately updated in the sequence view below. The T_m and primer pair annealing alignment are also updated.

- **T_m** The primer melting temperature. This value does not include overhangs or any added bases.
- **Primer pair annealing alignment** Predicted primer-primer annealing of the forward and reverse primers. Overhangs and added bases are not included. The same plot is also available in the primer pairs table.
- **Added bases** Insert additional bases between the primer and overhang. You can either type the bases directly into the dialog, or you can choose the sequence of a specific restriction enzyme from the drop down menu.
- **Sequence and primer views** For each sequence included in the homology cloning reaction, you can see the part of the sequence that primers are designed to, as well as the primers and their overhangs. For the vector, the fragment is considered circular and the primers are placed pointing in opposite directions from the insertion site (figure 23.34). Inserts are considered linear and primers are placed at the ends (figure 23.35).

Vector, inserts, primers and overhangs are color coded (figure 23.30):

- **Black** Vector or insert sequence
- **Grey** Vector or insert sequence not included in the PCR product
- **Brown** Primer sequence
- **Pink** Overhang sequence
- **Blue** Added bases that are inserted between primer and overhang

The overhang of a primer for a given sequence is identical to the sequence that it will be adjacent to in the assembled vector. Figures 23.34 and 23.35 show an example where the linear sequence can be inserted into the circular sequence. Pink overhang bases on the primers for the circular fragment are either the same sequence or complementary to the black sequence of the linear DNA fragment. Overhangs are designed to assemble the fragments in the order they appear in the wizard. In this example, two sequences are assembled, but more than two can be used for homology based cloning.

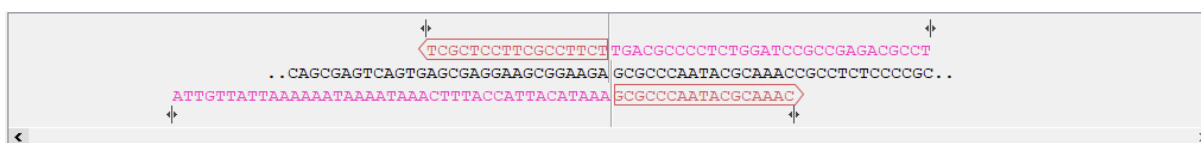


Figure 23.34: Vector sequence and corresponding primers.

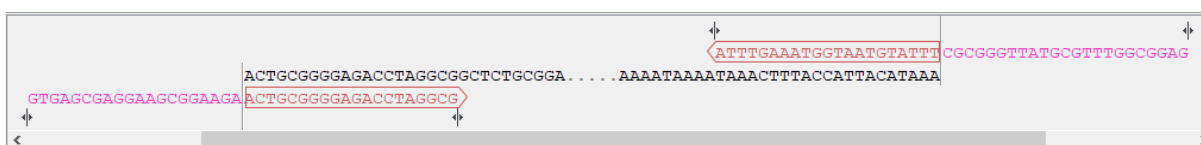


Figure 23.35: Insert sequence and corresponding primers.

- **Warnings in sequence views** A yellow or red exclamation mark next to the sequence name warns of any problems 23.36. Hover over the primer to get more information from the

tooltip or click on the warning to open a dialog showing the warning message. Examples of when warnings appear include:

- A primer does not have G/C content between 40 and 60 %.
- Only one primer is designed for a given fragment.
- An insert with a length that is not a multiple of 3 is placed inside a vector region annotated as a CDS and so may disrupt the reading frame.

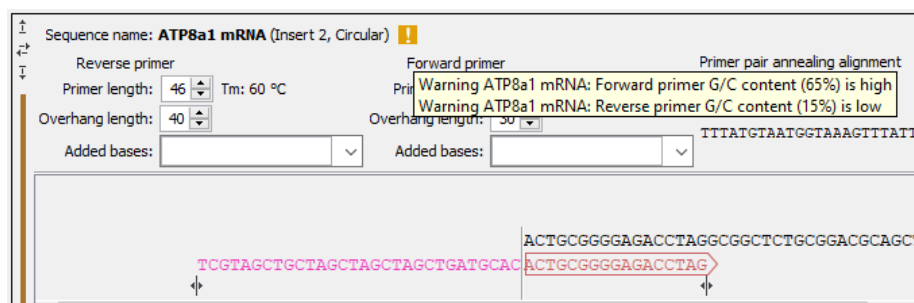


Figure 23.36: Hover over the yellow exclamation mark to see the warnings in the tooltip.

23.4.5 Working with mutations

Mutations can be introduced using PCR amplification with primers containing the mutations. This can be mimicked in the software:

- Introduce mutations manually in the sequences before running Homology Based Cloning. Place primers over mutated sites to ensure the mutations are included in the primer sequence.
- Run Homology Based Cloning using the original sequences, and then introduce mutations into the assembled vector. Re-run Homology Based Cloning using the vector containing the mutations.

23.5 Gateway cloning

CLC Genomics Workbench offers tools to perform *in silico* Gateway cloning (Thermo Fisher Scientific), including Multi-site Gateway cloning.

The three tools for doing Gateway cloning in the CLC Genomics Workbench mimic the procedure followed in the lab:

- First, attB sites are added to a sequence fragment
- Second, the attB-flanked fragment is recombined into a donor vector (the BP reaction) to construct an entry clone
- Finally, the target fragment from the entry clone is recombined into an expression vector (the LR reaction) to construct an expression clone. For Multi-site gateway cloning, multiple entry clones can be created that can recombine in the LR reaction.

During this process, both the attB-flanked fragment and the entry clone can be saved.

For more information about the Gateway technology, please visit <https://www.thermofisher.com/us/en/home/life-science/cloning/gateway-cloning/gateway-technology.html>. To perform these analyses in *CLC Genomics Workbench*, you need to import donor and expression vectors. These can be found on the Thermo Fisher Scientific's website: find the relevant vector sequences, copy them, and paste them in the field that opens when you choose **New | Sequence** in the workbench. Fill in additional information appropriately (enter a "Name", check the "Circular" option) and save the sequences in the Navigation Area.

23.5.1 Add attB sites

The first step in the Gateway cloning process is to amplify the target sequence with primers including so-called attB sites:

Tools | Molecular Biology Tools (🔧) | Cloning (🧬) | Gateway Cloning (📁) | Add attB Sites (🔗)

This will open a dialog where you can select the input sequence, sequences, or sequence lists containing fewer than 10,000 sequences. Note that if a fragment is part of a longer sequence, you will need to extract it prior to starting the tool: select the relevant region (or an annotation) of the original sequence, right-click the selection and choose to **Open Annotation in New View**. **Save** (📁) the new sequence in the Navigation Area.

When you have selected your fragment(s), click **Next**.

This will allow you to choose which attB sites you wish to add to each end of the fragment as shown in figure 23.37.

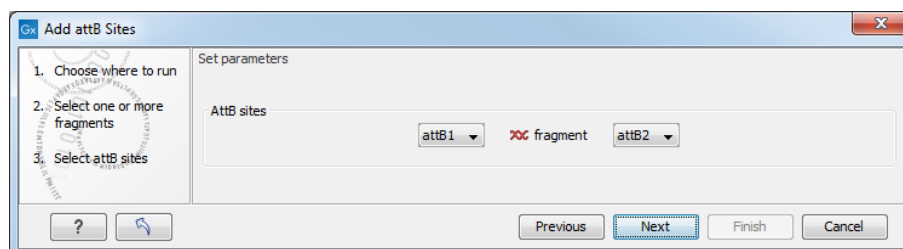


Figure 23.37: Selecting which attB sites to add.

The default option is to use the attB1 and attB2 sites. If you have selected several fragments and wish to add different combinations of sites, you will have to run this tool once for each combination.

Next, you are given the option to extend the fragment with additional sequences by extending the primers 5' of the template-specific part of the primer, i.e., between the template specific part and the attB sites.

You can manually type or paste in a sequence of your choice, but it is also possible to click in the text field and press **Shift + F1 (Shift + Fn + F1 on Mac)** to show some of the most common additions (see figure 23.38). Use the up and down arrow keys to select a tag and press **Enter**. To learn how to modify the default list of primer additions, see section 23.5.1.

At the bottom of the dialog, you can see a preview of what the final PCR product will look like. In the middle there is the sequence of interest. In the beginning is the attB1 site, and at the end is

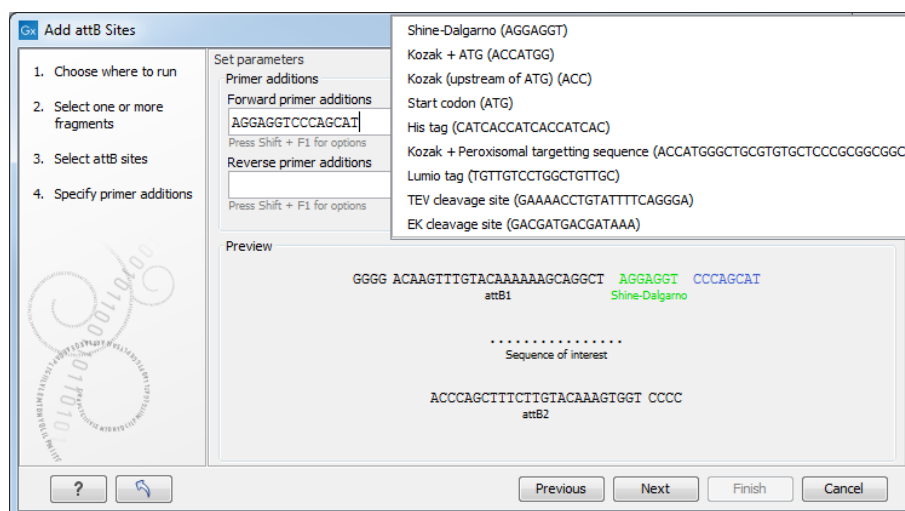


Figure 23.38: Primer additions 5' of the template-specific part of the primer where a Shine-Dalgarno site has been added between the attB site and the gene of interest.

the attB2 site. The primer additions that you have inserted are shown in colors.

In the next step, specify the length of the template-specific part of the primers as shown in figure 23.39.

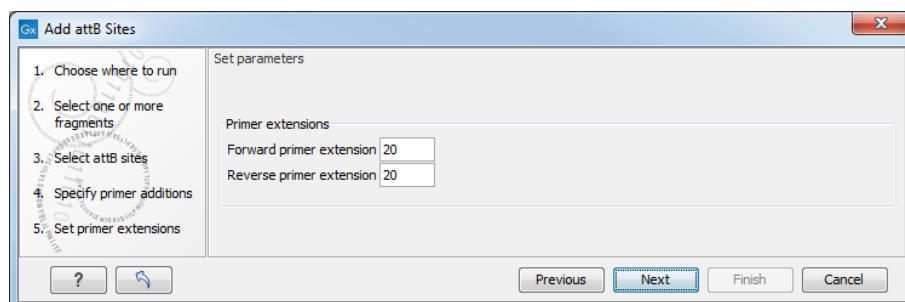


Figure 23.39: Specifying the length of the template-specific part of the primers.

The Workbench is not doing any kind of primer design when adding the attB sites. As a user, you simply specify the length of the template-specific part of the primer, and together with the attB sites and optional primer additions, this will be the primer. The primer region will be annotated in the resulting attB-flanked sequence. You can also choose to get a list of primers in the Result handling dialog (see figure 23.40).

The attB sites, the primer additions and the primer regions are annotated in the final result as shown in figure 23.41 (you may need to switch on the relevant annotation types to show the sites and primer additions).

There will be one output sequence for each sequence you have selected for adding attB sites. **Save** (⌘) the resulting sequence as it will be the input to the next part of the Gateway cloning workflow (see section 23.5.2).

Extending the pre-defined list of primer additions

The list of primer additions shown when pressing **Shift+F1** (on Mac: Shift + fn + F1) in the dialog shown in figure 23.38 can be configured and extended. If there is a tag that you use a lot, you

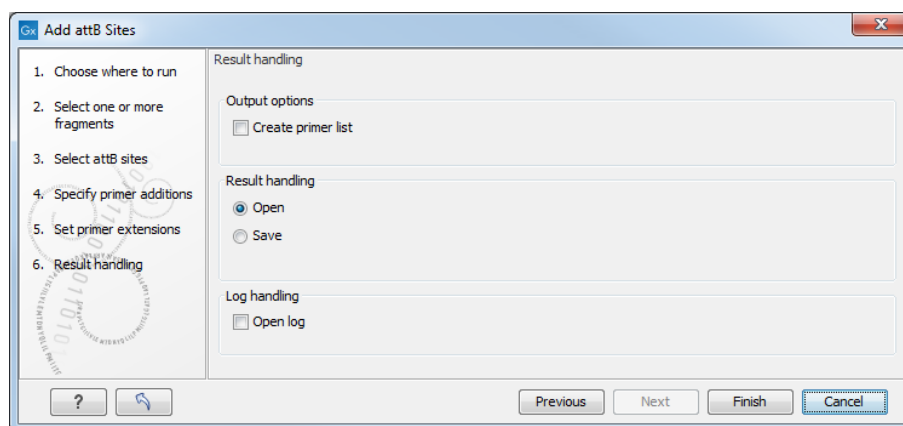


Figure 23.40: Besides the main output which is a copy of the input sequence(s) now including attB sites and primer additions, you can get a list of primers as output.

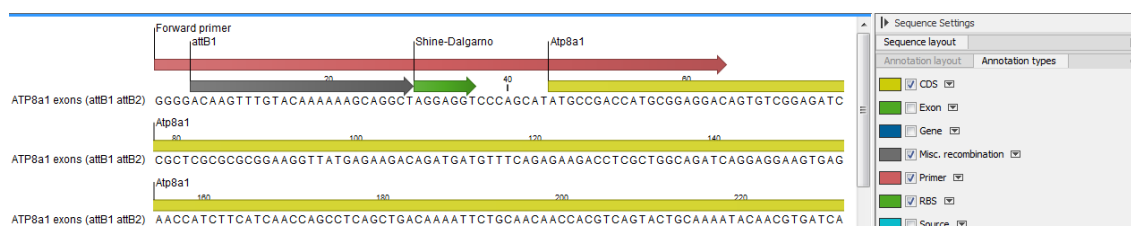


Figure 23.41: the attB site plus the Shine-Dalgarno primer addition is annotated.

can add it to the list for convenient and easy access later on. This is done in the **Preferences**:

Edit | Preferences | Data

In the table **Multisite Gateway Cloning primer additions** (see figure 23.42), select which primer addition options you want to add to forward or reverse primers. You can edit the existing elements in the table by double-clicking any of the cells, or you can use the buttons below to **Add Row** or **Delete Row**. If you by accident have deleted or modified some of the default primer additions, you can press **Add Default Rows**. Note that this will not reset the table but only add all the default rows to the existing rows.

Each element in the list has the following information:

Name When the sequence fragment is extended with a primer addition, an annotation will be added displaying this name.

Sequence The actual sequence to be inserted, defined on the sense strand (although the reverse primer would be reverse complement).

Annotation type The annotation type of the primer that is added to the fragment.

Forward primer addition Whether this addition should be visible in the list of additions for the forward primer.

Reverse primer addition Whether this addition should be visible in the list of additions for the reverse primer.

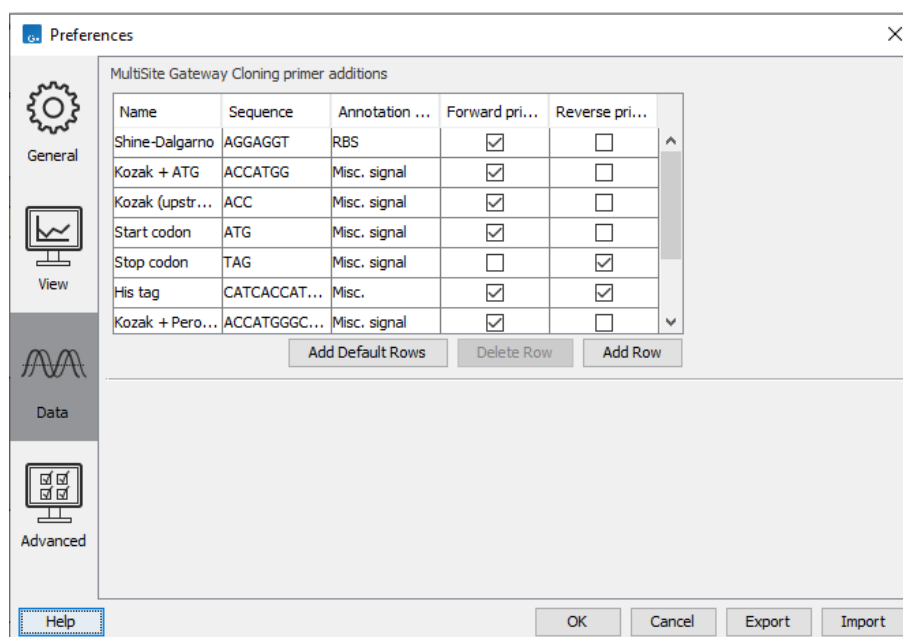


Figure 23.42: Configuring the list of primer additions available when adding attB sites.

23.5.2 Create entry clones (BP)

The next step in the Gateway cloning work flow is to recombine the attB-flanked sequence of interest into a donor vector to create an entry clone. Before proceeding to this step, make sure that the sequence of the destination vector was saved in the Navigation Area: find the relevant vector sequence on the Thermo Fisher Scientific's website, copy it, and paste it in the field that opens when you choose **New | Sequence** in the workbench. Fill in additional information appropriately (enter a "Name", check the "Circular" option) and save the sequence in the Navigation Area.

Tools | Molecular Biology Tools (🔧) | Cloning (🧬) | Gateway Cloning (📁) | Create Entry Clone (🔍)

In the first wizard window, select one or more sequences to be recombined into your donor vector. Note that the sequences you select should be flanked with attB sites (see section 23.5.1). You can select more than one sequence as input, and the corresponding number of entry clones will be created.

In the following dialog (figure 23.43), you can specify a donor vector.

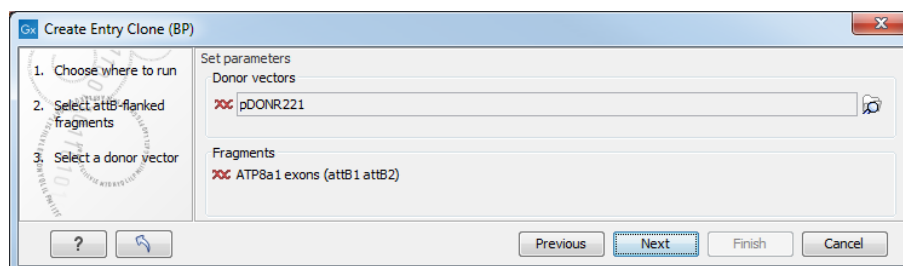


Figure 23.43: Selecting one or more donor vectors.

Once the vector is selected, a preview of the fragments selected and the attB sites that they contain is shown. This can be used to get an overview of which entry clones should be used and

check that the right attB sites have been added to the fragments. Also note that the workbench looks for the attP sites (see how to change the definition of sites in appendix F), but it does not check that they correspond to the attB sites of the selected fragments at this step. If the right combination of attB and attP sites is not found, no entry clones will be produced.

The output is one entry clone per sequence selected. The attB and attP sites have been used for the recombination, and the entry clone is now equipped with attL sites as shown in figure 23.44.

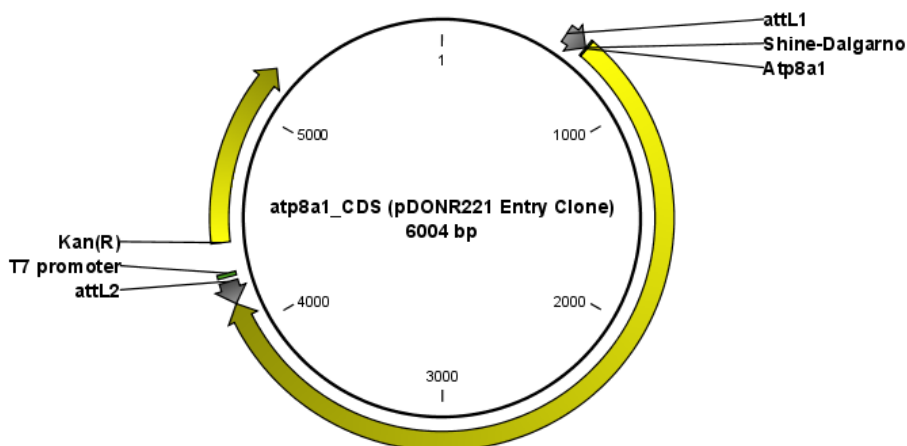


Figure 23.44: The resulting entry vector opened in a circular view.

Note that the bi-product of the recombination is not part of the output.

23.5.3 Create expression clones (LR)

The final step in the Gateway cloning work flow is to recombine the entry clone into a destination vector to create an expression clone. Before proceeding to this step, make sure that the sequence of the destination vector was saved in the Navigation Area: find the relevant vector sequence on the Thermo Fisher Scientific's website, copy it, and paste it in in the field that opens when you choose **New | Sequence** in the workbench. Fill in additional information appropriately (enter a "Name", check the "Circular" option) and save the sequence in the Navigation Area.

Note also that for a destination vector to be recognized, it must contain appropriate att sites and the *ccdB* gene. This gene must be present either as a 'ccdB' annotation, or as the exact sequence:

```
ATGCAGTTTAAGGTTTACACCTATAAAAGAGAGAGCCGTTATCGTCTGTTTGTGGATGTACAGAGTGATATT
ATTGACACGCCCCGGGCGACGGATGGTGATCCCCCTGGCCAGTGACGTCTGCTGTCAGATAAAGTCTCC
CGTGAACCTTACCCGGTGGTGCATATCGGGGATGAAAGCTGGCGCATGATGACCACCGATATGGCCAGT
GTGCCGGTCTCCGTTATCGGGGAAGAAGTGGCTGATCTCAGCCACCGCGAAAATGACATCAAAAACGCC
ATTAACCTGATGTTCTGGGGAATATAA
```

If the *ccdB* gene is not present or if the sequence is not identical to the above, a solution is to simply add a 'ccdB' annotation. Select part of the vector sequence, right-click and choose 'Add Annotation'. Name the annotation 'ccdB'.

You can now start the tool:

Tools | Molecular Biology Tools (🔧) | Cloning (🧬) | Gateway Cloning (📁) | Create Expression Clone (🔄)

In the first step, select one or more entry clones (see how to create an entry clone in section 23.5.2). If you wish to perform separate LR reactions with multiple entry clones, you should run the **Create Expression Clone** in batch mode (see section 12.3).

In the second step, select the destination vector that was previously saved in the Navigation Area (fig 23.45).

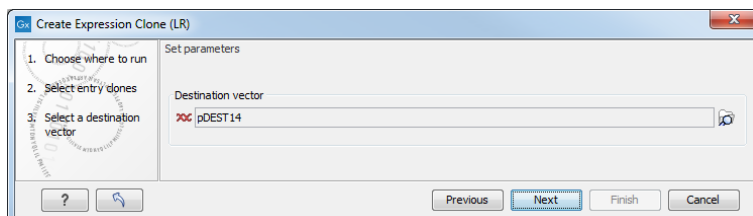


Figure 23.45: Selecting one or more destination vectors.

Note that the workbench looks for the specific sequences of the attR sites in the sequences that you select in this dialog (see how to change the definition of sites in appendix F), but it does not check that they correspond to the attL sites of the selected fragments. If the right combination of attL and attR sites is not found, no entry clones will be produced.

When performing multi-site gateway cloning, *CLC Genomics Workbench* will insert the fragments (contained in entry clones) by matching the sites that are compatible. If the sites have been defined correctly, an expression clone containing all the fragments will be created. You can find an explanation of the multi-site gateway system at <https://www.thermofisher.com/dk/en/home/life-science/cloning/gateway-cloning/multisite-gateway-technology.html?SID=fr-gwcloning-3>

The output is a number of expression clones depending on how many entry clones and destination vectors that you selected. The attL and attR sites have been used for the recombination, and the expression clone is now equipped with attB sites as shown in figure 23.46.

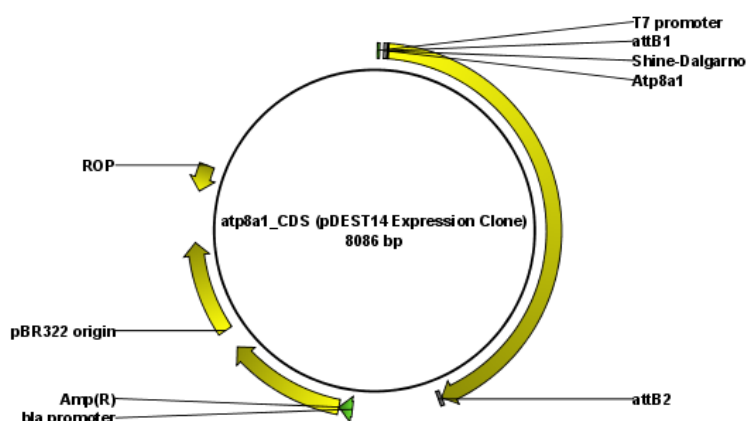


Figure 23.46: The resulting expression clone opened in a circular view.

You can choose to create a sequence list with the bi-products as well.

23.6 Gel electrophoresis

CLC Genomics Workbench enables the user to simulate the separation of nucleotide sequences on a gel. This feature is useful when designing an experiment which will allow the differentiation of a successful and an unsuccessful cloning experiment on the basis of a restriction map.

There are several ways to simulate gel separation of nucleotide sequences:

- When using the **Restriction Site Analysis** tool, available under the Tools menu, you can choose to create a restriction map which can be shown as a gel (see section 23.1.2).
- From all the graphical views of sequences, you can right-click the name of the sequence and choose **Digest and Create Restriction Map** (☰). The sequence will be digested with the enzymes that are selected in the Side Panel. The views where this option is available are listed below:
 - Circular view (see section 15.2.5).
 - Ordinary sequence view (see section 15.2).
 - Graphical view of sequence lists (see section 15.1).
 - Cloning editor (see section 23.3).
 - Primer designer (see section 21.3).
- **Separate sequences on gel:** To separate sequences without restriction enzyme digestion, first create a sequence list of the sequences in question, then click the **Gel** button (☰) at the bottom of the view of the sequence list (figure 23.47).

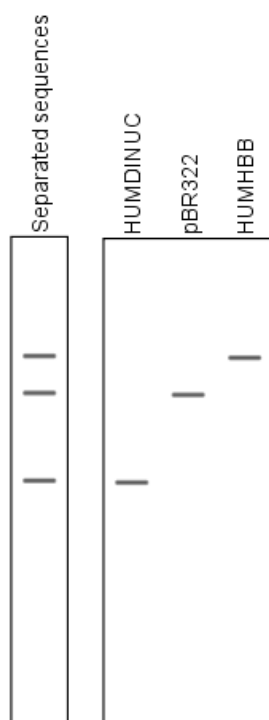


Figure 23.47: A sequence list shown as a gel.

23.6.1 Gel view

In figure 23.48 you can see a simulation of a gel with its Side Panel to the right.

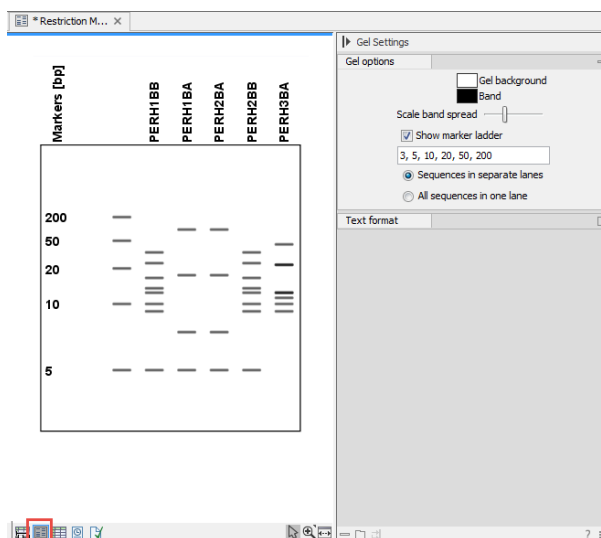


Figure 23.48: Five lanes showing fragments of five sequences cut with restriction enzymes.

Information on bands / fragments You can get information about the individual bands by hovering the mouse cursor on the band of interest. This will display a tool tip with the following information:

- Fragment length
- Fragment region on the original sequence
- Enzymes cutting at the left and right ends, respectively

For gels comparing whole sequences, you will see the sequence name and the length of the sequence.

Note! You have to be in **Selection** () or **Pan** () mode in order to get this information.

It can be useful to add markers to the gel which enables you to compare the sizes of the bands. This is done by clicking **Show marker ladder** in the **Side Panel**.



Markers can be entered into the text field, separated by commas.

Modifying the layout The background of the lane and the colors of the bands can be changed in the Side Panel. Click the colored box to display a dialog for picking a color. The slider **Scale band spread** can be used to adjust the effective time of separation on the gel, i.e. how much the bands will be spread over the lane. In a real electrophoresis experiment this property will be determined by several factors including time of separation, voltage and gel density.

You can also choose how many lanes should be displayed:

- **Sequences in separate lanes.** This simulates that a gel is run for each sequence.

- **All sequences in one lane.** This simulates that one gel is run for all sequences.

You can also modify the layout of the view by zooming in or out. Click **Zoom in** () or **Zoom out** () in the Toolbar and click the view.

Finally, you can modify the format of the text heading each lane in the Text format preferences in the Side Panel.

Chapter 24

Sequence alignment

Contents

24.1 Create an alignment	625
24.1.1 Gap costs	626
24.1.2 Fast or accurate alignment algorithm	627
24.1.3 Aligning alignments	628
24.1.4 Fixpoints	628
24.2 View alignments	630
24.2.1 Bioinformatics explained: Sequence logo	634
24.3 Edit alignments	636
24.3.1 Realignment	638
24.4 Join alignments	640
24.5 Pairwise comparison	641
24.5.1 The pairwise comparison table	643
24.5.2 Bioinformatics explained: Multiple alignments	644

CLC Genomics Workbench can align nucleotides and proteins using a *progressive alignment* algorithm (see section 24.5.2).

This chapter describes how to use the program to align sequences, and alignment algorithms in more general terms.

24.1 Create an alignment

Alignments can be created from sequences, sequence lists (see section 15.1), existing alignments and from any combination of the three.

To create an alignment, run the **Create Alignment** tool, available at:

Tools | Classical Sequence Analysis () | **Alignments and Trees** () | **Create Alignment** ()

This opens the dialog shown in figure 24.1.

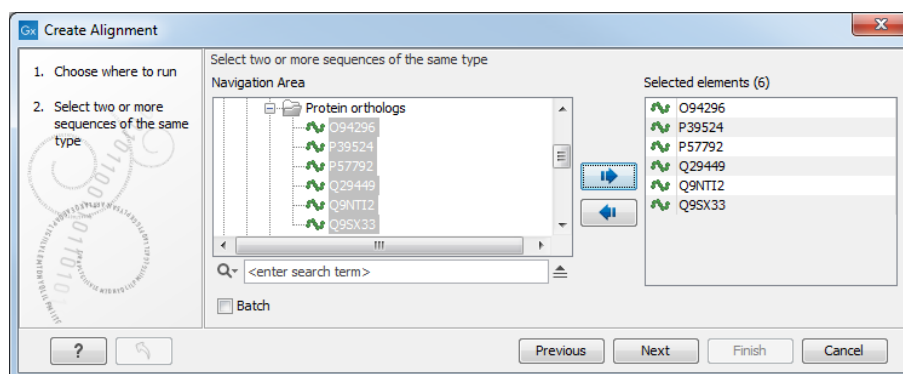


Figure 24.1: Creating an alignment.

After selecting the elements to align, you are presented with options that can be configured (figure 24.2).

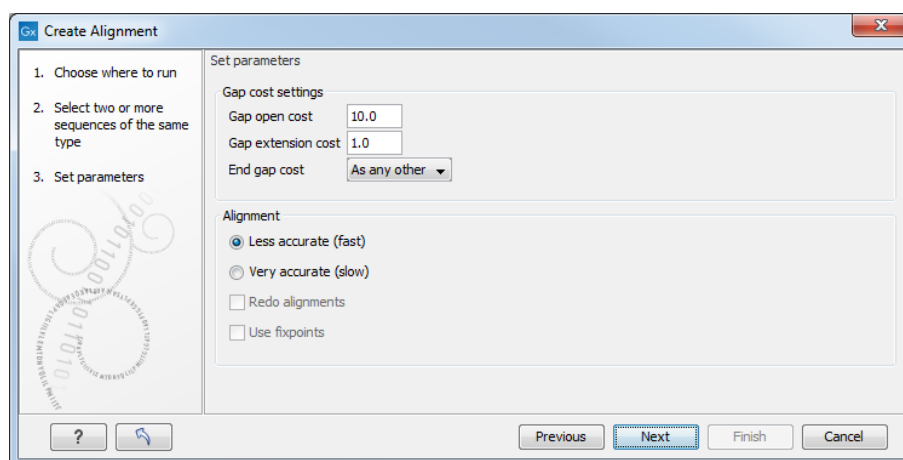


Figure 24.2: Adjusting alignment algorithm parameters.

24.1.1 Gap costs

The alignment algorithm has three parameters concerning gap costs: Gap open cost, Gap extension cost and End gap cost. The precision of these parameters is to one place of decimal.

- **Gap open cost.** The price for introducing gaps in an alignment.
- **Gap extension cost.** The price for every extension past the initial gap.

If you expect a lot of small gaps in your alignment, the Gap open cost should equal the Gap extension cost. On the other hand, if you expect few but large gaps, the Gap open cost should be set significantly higher than the Gap extension cost.

However, for most alignments it is a good idea to make the Gap open cost quite a bit higher than the Gap extension cost. The default values are 10.0 and 1.0 for the two parameters, respectively.

- **End gap cost.** The price of gaps at the beginning or the end of the alignment. One of the advantages of the *CLC Genomics Workbench* alignment method is that it provides flexibility in the treatment of gaps at the ends of the sequences. There are three possibilities:

- **Free end gaps.** Any number of gaps can be inserted in the ends of the sequences without any cost.
- **Cheap end gaps.** All end gaps are treated as gap extensions and any gaps past 10 are free.
- **End gaps as any other.** Gaps at the ends of sequences are treated like gaps in any other place in the sequences.

When aligning a long sequence with a short partial sequence, it is ideal to use free end gaps, since this will be the best approximation to the situation. The many gaps inserted at the ends are not due to evolutionary events, but rather to partial data.

Many homologous proteins have quite different ends, often with large insertions or deletions. This confuses alignment algorithms, but using the **Cheap end gaps** option, large gaps will generally be tolerated at the sequence ends, improving the overall alignment. This is the default setting of the algorithm.

Finally, treating end gaps like any other gaps is the best option when you know that there are no biologically distinct effects at the ends of the sequences.

Figures 24.3 and 24.4 illustrate the differences between the different gap scores at the sequence ends.

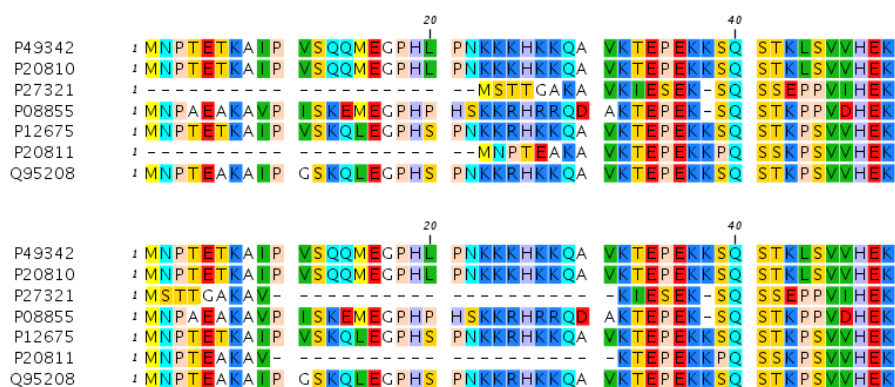


Figure 24.3: The first 50 positions of two different alignments of seven calpastatin sequences. The top alignment is made with cheap end gaps, while the bottom alignment is made with end gaps having the same price as any other gaps. In this case it seems that the latter scoring scheme gives the best result.

24.1.2 Fast or accurate alignment algorithm

CLC Genomics Workbench has two algorithms for calculating alignments:

- **Fast (less accurate).** Use an optimized alignment algorithm that is very fast. This is particularly useful for data sets with very long sequences.
- **Slow (very accurate).** The recommended choice unless the processing time is too long.

Both algorithms use progressive alignment. The faster algorithm builds the initial tree by doing more approximate pairwise alignments than the slower option.

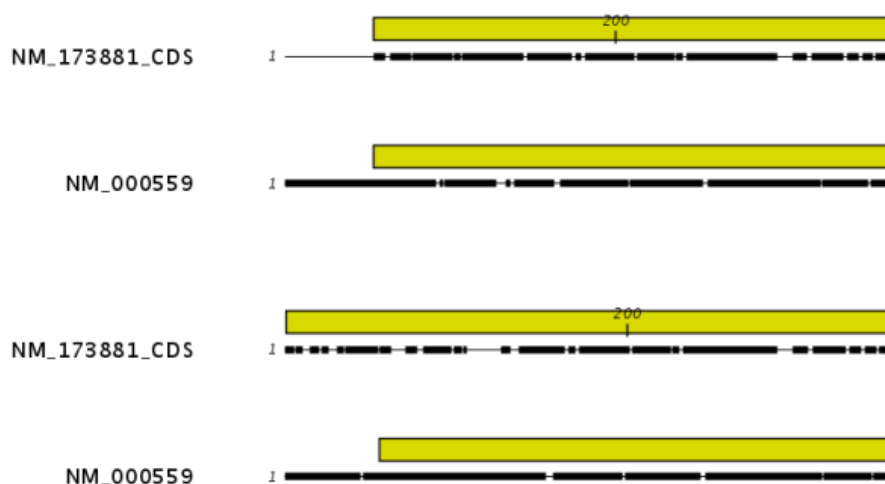


Figure 24.4: The alignment of the coding sequence of bovine myoglobin with the full mRNA of human gamma globin. The top alignment is made with free end gaps, while the bottom alignment is made with end gaps treated as any other. The yellow annotation is the coding sequence in both sequences. It is evident that free end gaps are ideal in this situation as the start codons are aligned correctly in the top alignment. Treating end gaps as any other gaps in the case of aligning distant homologs where one sequence is partial leads to a spreading out of the short sequence as in the bottom alignment.

24.1.3 Aligning alignments

The **Redo alignment** option becomes available when an alignment is provided as input to the tool (figure 24.1).

- **Leave this box unchecked** when aligning additional sequences to the original alignment. Equal sized gaps may be inserted in all sequences of the original alignment to accommodate the alignment of the new sequences (figure 24.5), but apart from this, positions in the original alignment are fixed.
- **Check this box** to realign the sequences in the alignment provided as input. This can be useful, for example, if you wish to realign using different gap costs than used originally.

A new alignment is output, whether or not **Redo alignment** option is checked.

24.1.4 Fixpoints

To force particular regions of an alignment to be aligned to each other, there are two steps:

1. Add fixpoint annotations to these regions.
2. Check the "Use fixpoints" option when launching the Create Alignment tool.

To add a fixpoint, open the sequence or alignment and:

Select the region you want to use as a fixpoint | Right-click on the selection | Set Alignment Fixpoint Here

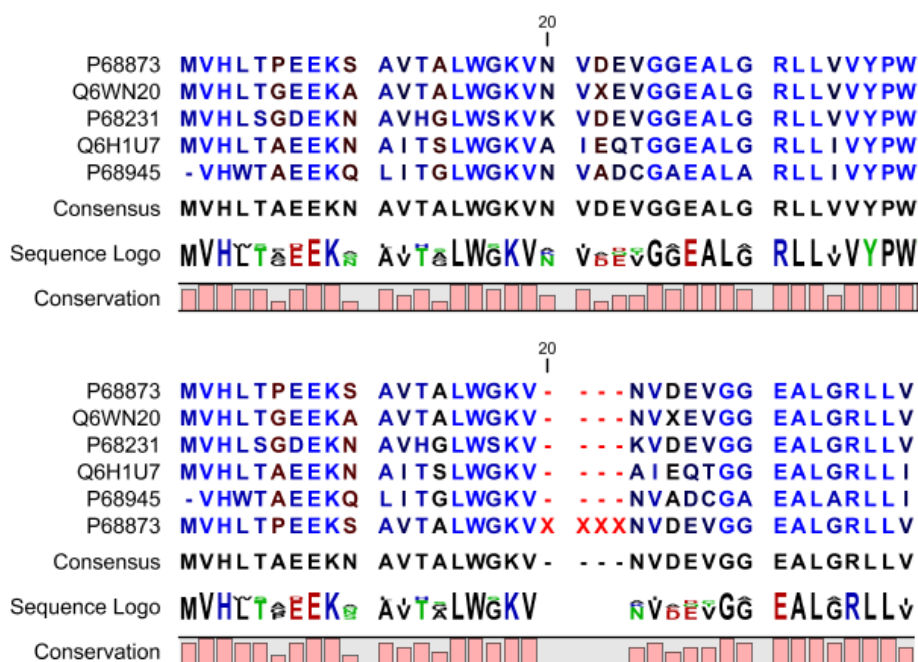


Figure 24.5: The original alignment is shown at the top. That alignment and a single additional sequence, with four Xs added for illustrative purposes, were used as input to Create Alignment. The "Redo alignment" option was left unchecked. The resulting alignment is shown at the bottom. Gaps have been added, compared to the original alignment, to accommodate the new sequence. All other positions are aligned as they were in the original alignment.

This will add an annotation of type "Alignment fixpoint", with name "Fixpoint" to the sequence (figure 24.6).

Regions with fixpoint annotations with the same name are aligned to each other. Where there are multiple fixpoints of the same name on sequences, the first fixpoints on each sequence will be aligned to each other, the second on each sequence will be aligned to each other, and so on.

To adjust the name of a fixpoint annotation:

Right-click the Fixpoint annotation | Edit Annotation (🔧) | Type the name in the 'Name' field

An example where assigning different names to fixpoints is useful: Given three sequences, A, B and C, where A and B each have one copy of a domain while sequence C has two copies of the domain, you can force sequence A to align to the first copy of the domain in sequence C and sequence B to align to the second copy of the domain in sequence C by naming the fixpoints accordingly. E.g. if the fixpoints in sequence C were named 'fp1' and 'fp2', the fixpoint in sequence A was named 'fp1' and the fixpoint in sequence B was named 'fp2', then when these sequences are aligned using fixpoints, the fixpoint in sequence A would be aligned to the first copy of the domain in sequence C, while the fixpoint in sequence B would be aligned to the second copy of the domain in sequence C.

The result of an alignment using fixpoints is shown in figure 24.7.



Figure 24.6: Select a region and right-click on it to see the option to set a fixpoint. The second sequence in the list already has a Fixpoint annotation.

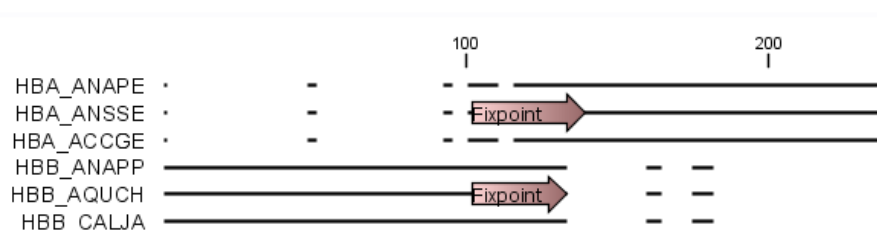
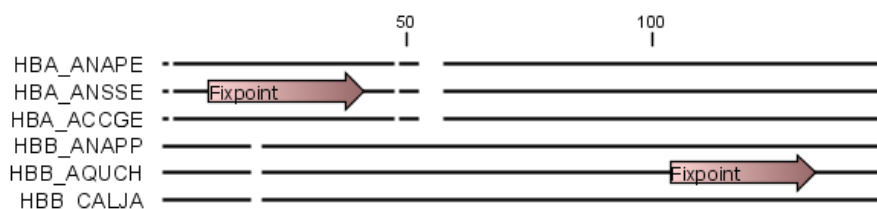


Figure 24.7: Fixpoints have been added to 2 sequences in an alignment, where the first 3 sequences are very similar to each other and the last 3 sequences are very similar to each other (top). After realigning using just these 2 fixpoints (bottom), the alignment now shows clearly the 2 groups of sequences.

24.2 View alignments

The basic options for viewing alignments are the same as for viewing sequences, described in section 15.2.

Alignment specific view options in the **Sequence layout**, **Nucleotide info**, **Alignment info**, and **Positional stats** side panel tabs, to the right of the view, are described here.

Sequence layout

In the side panel tab **Sequence layout** the option **Alignments on top** supports moving the aligned

sequences relative to other elements shown in the alignment view. When checked, the alignment is shown at the top of the view, when unchecked the alignment is shown underneath other included summary information such as Consensus, Conservation and Sequence logo.

Nucleotide info

In the side panel tab **Nucleotide info** under **Translation**, there is an extra checkbox: **Relative to top sequence**. Checking this box will make the reading frames for the translation align with the top sequence so that you can compare the effect of nucleotide differences on the protein level.

Alignment info

The entire side panel tab **Alignment info** is specific to alignments. Each of the options in the **Alignment info** relate to each column in the alignment.

The data points for graph representations can be exported (see section 8.3).

Consensus Shows a consensus sequence at the bottom of the alignment. The consensus sequence is based on every single position in the alignment and reflects an artificial sequence which resembles the sequence information of the alignment, but only as one single sequence. If all sequences of the alignment is 100% identical the consensus sequence will be identical to all sequences found in the alignment. If the sequences of the alignment differ the consensus sequence will reflect the most common sequences in the alignment. Parameters for adjusting the consensus sequences are described below.

- **Limit** This option determines how conserved the sequences must be in order to agree on a consensus. Here you can also choose **IUPAC** which will display the ambiguity code when there are differences between the sequences. For example, an alignment with **A** and a **G** at the same position will display an **R** in the consensus line if the **IUPAC** option is selected. The IUPAC codes can be found in section **H** and **G**. Please note that the IUPAC codes are only available for nucleotide alignments.
- **No gaps** Checking this option will not show gaps in the consensus.
- **Ambiguous symbol** Select how ambiguities should be displayed in the consensus line (as **N**, **?**, *****, **.** or **-**). This option has no effect if **IUPAC** is selected in the **Limit** list above.

The Consensus Sequence can be opened in a new view, simply by right-clicking the Consensus Sequence and click **Open Consensus in New View**.

Conservation Displays the level of conservation at each position in the alignment. The conservation shows the conservation of all sequence positions. The height of the bar, or the gradient of the color reflect how conserved that particular position is in the alignment. If one position is 100% conserved the bar will be shown in full height, and it is colored in the color specified at the right side of the gradient slider.

- **Foreground color** Colors the letters using a gradient, where the right side color is used for highly conserved positions and the left side color is used for positions that are less conserved.
- **Background color.** Sets a background color of the residues using a gradient in the same way as described above.
- **Graph** Displays the conservation level as a graph at the bottom of the alignment. The bar (default view) show the conservation of all sequence positions. The height of the graph reflects how conserved that particular position is in the alignment. If one position is 100% conserved the graph will be shown in full height.
 - **Height** Specifies the height of the graph.
 - **Type** The type of the graph: **Line plot**, **Bar plot**, or **Colors**, in which case the graph is seen as a color bar using a gradient like the foreground and background colors.
 - **Color box** Specifies the color of the graph for line and bar plots, and specifies a gradient for colors.

Gap fraction Which fraction of the sequences in the alignment that have gaps. The gap fraction is only relevant if there are gaps in the alignment.

- **Foreground color** Colors the letter using a gradient, where the left side color is used if there are relatively few gaps, and the right side color is used if there are relatively many gaps.
- **Background color** Sets a background color of the residues using a gradient in the same way as described above.
- **Graph** Displays the gap fraction as a graph at the bottom of the alignment.
 - **Height** Specifies the height of the graph.
 - **Type** The type of the graph: **Line plot**, **Bar plot**, or **Colors**, in which case the graph is seen as a color bar using a gradient like the foreground and background colors.
 - **Color box** Specifies the color of the graph for line and bar plots, and specifies a gradient for colors.

Color different residues Indicates differences in aligned residues.

- **Foreground color** Colors the letter.
- **Background color.** Sets a background color of the residues.

Sequence logo A sequence logo displays the frequencies of residues at each position in an alignment. This is presented as the relative heights of letters, along with the degree of sequence conservation as the total height of a stack of letters, measured in bits of information. The vertical scale is in bits, with a maximum of 2 bits for nucleotides and approximately 4.32 bits for amino acid residues. See section [24.2.1](#) for more details.

- **Foreground color** Color the residues using a gradient according to the information content of the alignment column. Low values indicate columns with high variability whereas high values indicate columns with similar residues.
- **Background color** Sets a background color of the residues using a gradient in the same way as described above.
- **Logo** Displays sequence logo at the bottom of the alignment.
 - **Height** Specifies the height of the sequence logo graph.
 - **Color** The sequence logo can be displayed in black or Rasmol colors. For protein alignments, a polarity color scheme is also available, where hydrophobic residues are shown in black color, hydrophilic residues as green, acidic residues as red and basic residues as blue.

Positional stats

The side panel tab **Positional stats** provides site specific information about the alignment. Hover the mouse cursor over a position in the alignment or make a selection to populate the tab with information (figure 24.8).



Figure 24.8: Contents of the *Positional stats* tab when a single sequence is selected (Left) and when three sequences are selected (Right). Note that the side panel can be dragged into the alignment view.

When one position is selected, the information provided is calculated from all the sequences at the position. If more sequences at the same position are selected, the information is calculated for the selected sequences only.

The following information is provided:

- **Alignment position** The selected position in the alignment.
- **Pairwise % identity** Average percent identity. All pairs of bases at the same position are compared. The number of identical pairs is counted and divided by the total number

of pairs. The count of ambiguity characters is scaled to the number of bases they can represent, for example a G compared to an R (A or G) is given the value 0.5.

Example calculation for an alignment with the nucleotides A, A and G in the tested position: There are three pairwise comparisons, A to A = 1, A to G = 0, and A to G = 0. The pairwise % identity is then 1/3.

- **A, C, G, T and** - Percentage of A, C, G, T and "no nucleotide" at the selected position.

24.2.1 Bioinformatics explained: Sequence logo

In the search for homologous sequences, researchers are often interested in conserved sites/residues or positions in a sequence which tend to differ a lot. Most researches use alignments (see Bioinformatics explained: *multiple alignments*) for visualization of homology on a given set of either DNA or protein sequences. In proteins, active sites in a given protein family are often highly conserved. Thus, in an alignment these positions (which are not necessarily located in proximity) are fully or nearly fully conserved. On the other hand, antigen binding sites in the F_{ab} unit of immunoglobulins tend to differ quite a lot, whereas the rest of the protein remains relatively unchanged.

In DNA, promoter sites or other DNA binding sites are highly conserved (see figure 24.9). This is also the case for repressor sites as seen for the Cro repressor of bacteriophage λ .

When aligning such sequences, regardless of whether they are highly variable or highly conserved at specific sites, it is very difficult to generate a consensus sequence which covers the actual variability of a given position. In order to better understand the information content or significance of certain positions, a sequence logo can be used. The sequence logo displays the information content of all positions in an alignment as residues or nucleotides stacked on top of each other (see figure 24.9). The sequence logo provides a far more detailed view of the entire alignment than a simple consensus sequence. Sequence logos can aid to identify protein binding sites on DNA sequences and can also aid to identify conserved residues in aligned domains of protein sequences and a wide range of other applications.

Each position of the alignment and consequently the sequence logo shows the sequence information in a computed score based on Shannon entropy [Schneider and Stephens, 1990]. The height of the individual letters represent the sequence information content in that particular position of the alignment.

A sequence logo is a much better visualization tool than a simple consensus sequence. An example hereof is an alignment where in one position a particular residue is found in 70% of the sequences. If a consensus sequence is used, it typically only displays the single residue with 70% coverage. In figure 24.9 an un-gapped alignment of 11 *E. coli* start codons including flanking regions are shown. In this example, a consensus sequence would only display ATG as the start codon in position 1, but when looking at the sequence logo it is seen that a GTG is also allowed as a start codon.

Calculation of sequence logos A comprehensive walk-through of the calculation of the information content in sequence logos is beyond the scope of this document but can be found in the original paper by [Schneider and Stephens, 1990]. Nevertheless, the conservation of every position is defined as R_{seq} which is the difference between the maximal entropy (S_{max}) and the observed entropy for the residue distribution (S_{obs}),

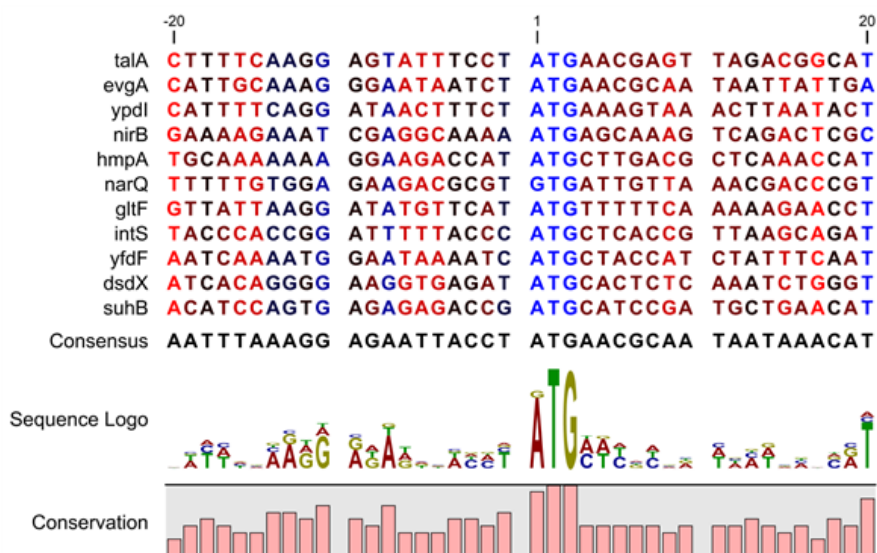


Figure 24.9: Ungapped sequence alignment of eleven *E. coli* sequences defining a start codon. The start codons start at position 1. Below the alignment is shown the corresponding sequence logo. As seen, a GTG start codon and the usual ATG start codons are present in the alignment. This can also be visualized in the logo at position 1.

$$R_{seq} = S_{max} - S_{obs} = \log_2 N - \left(- \sum_{n=1}^N p_n \log_2 p_n \right)$$

p_n is the observed frequency of a amino acid residue or nucleotide of symbol n at a particular position and N is the number of distinct symbols for the sequence alphabet, either 20 for proteins or four for DNA/RNA. This means that the maximal sequence information content per position is $\log_2 4 = 2$ bits for DNA/RNA and $\log_2 20 \approx 4.32$ bits for proteins.

The original implementation by Schneider does not handle sequence gaps.

We have slightly modified the algorithm so an estimated logo is presented in areas with sequence gaps.

If amino acid residues or nucleotides of one sequence are found in an area containing gaps, we have chosen to show the particular residue as the fraction of the sequences. Example; if one position in the alignment contain 9 gaps and only one alanine (A) the A represented in the logo has a hight of 0.1.

Other useful resources

The website of Tom Schneider

<http://www-lmmb.ncifcrf.gov/~toms/>

WebLogo

<http://weblogo.berkeley.edu/>

[Crooks et al., 2004]

24.3 Edit alignments

Move residues and gaps The placement of gaps in the alignment can be changed by modifying the parameters when creating the alignment (see section 24.1). However, gaps and residues can also be moved after the alignment is created:

select one or more gaps or residues in the alignment | drag the selection to move

This can be done both for single sequences, but also for multiple sequences by making a selection covering more than one sequence. When you have made the selection, the mouse pointer turns into a horizontal arrow indicating that the selection can be moved (see figure 24.10).

Note! Residues can only be moved when they are next to a gap.

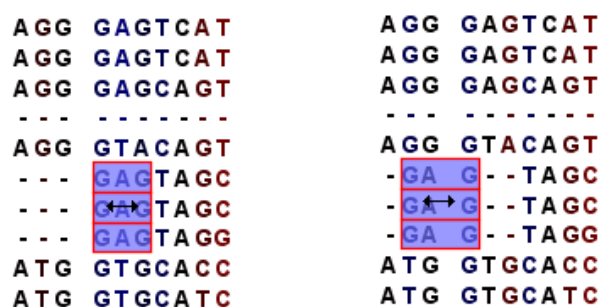


Figure 24.10: Moving a part of an alignment. Notice the change of mouse pointer to a horizontal arrow.

Insert gaps The placement of gaps in the alignment can be changed by modifying the parameters when creating the alignment. However, gaps can also be added manually after the alignment is created.

To insert extra gaps:

select a part of the alignment | right-click the selection | Add gaps before/after

If you have made a selection covering five residues for example, a gap of five will be inserted. In this way you can easily control the number of gaps to insert. Gaps will be inserted in the sequences that you selected. If you make a selection in two sequences in an alignment, gaps will be inserted into these two sequences. This means that these two sequences will be displaced compared to the other sequences in the alignment.

Delete residues and gaps Residues or gaps can be deleted for individual sequences or for the whole alignment. For individual sequences:

select the part of the sequence you want to delete | right-click the selection | Edit Selection (📏) | Delete the text in the dialog | Replace

The selection shown in the dialog will be replaced by the text you enter. If you delete the text, the selection will be replaced by an empty text, i.e. deleted.

In order to delete entire columns:

manually select the columns to delete | right-click the selection | click 'Delete Selection'

Copy annotations to other sequences Annotations on one sequence can be transferred to other sequences in the alignment:

right-click the annotation | Copy Annotation to other Sequences

This will display a dialog listing all the sequences in the alignment. Next to each sequence is a checkbox which is used for selecting which sequences the annotation should be copied to. Click **Copy** to copy the annotation.

If you wish to copy all annotations on the sequence, click the **Copy All Annotations to other Sequences**.

Copied/transferred annotations will contain the same qualifier text as the original, i.e., the text is not updated. As an example, if the annotation contains 'translation' as qualifier text, this translation will be copied to the new sequence and will thus reflect the translation of the original sequence, and not the new sequence which may differ.

Move sequences up and down Sequences can be moved up and down in the alignment:

drag the name of the sequence up or down

When you move the mouse pointer over the label, the pointer will turn into a vertical arrow indicating that the sequence can be moved.

The sequences can also be sorted automatically to let you save time moving the sequences around. To sort the sequences alphabetically:

Right-click the name of a sequence | Sort Sequences Alphabetically

If you change the Sequence name (in the **Sequence Layout** view preferences), you will have to ask the program to sort the sequences again.

If you have one particular sequence that you would like to use as a reference sequence, it can be useful to move this to the top. This can be done manually, but it can also be done automatically:

Right-click the name of a sequence | Move Sequence to Top

The sequences can also be sorted by similarity, grouping similar sequences together:

Right-click the name of a sequence | Sort Sequences by Similarity

Delete, rename and add sequences Sequences can be removed from the alignment by right-clicking the label of a sequence:

right-click label | Delete Sequence

If you wish to delete several sequences, you can check all the sequences, right-click and choose **Delete Marked Sequences**. To show the checkboxes, you first have to click the **Show Selection Boxes** in the **Side Panel**.

A sequence can also be renamed:

right-click label | Rename Sequence

This will show a dialog, letting you rename the sequence. This will not affect the sequence that the alignment is based on.

Extra sequences can be added to the alignment by creating a new alignment where you select the current alignment and the extra sequences (see section 24.1).

The same procedure can be used for joining two alignments.

24.3.1 Realignment

This section describes realigning parts of an existing alignment. To realign an entire alignment, consider using the "Redo alignment" option of the **Create Alignment** tool, described in section 24.1.3

Examples where realigning part of an alignment can be helpful include:

- **Adjusting the number of gaps** If a region has more gaps than is useful, select the region of interest and realign using a higher gap cost.
- **Combine with fixpoints** When you have an alignment where two residues are not aligned although they should have been, you can set an alignment fixpoint on each of those residues. and then realign the section of interest using those fixpoints, as described in section 24.1.4. This should result in the two residues being aligned, and everything in the selected region around them being adjusted to accommodate that change.

Realigning a subsection of an alignment

There are two steps to realigning a subsection of an alignment:

1. Select the region of interest.
2. Realign the selected region.

Selecting a region

Click and drag to select the regions of interest. For small regions in a small number of sequences, this may be easiest while zoomed in fully, such that each residue is visible. For realigning entire sequences, zooming out fully may be helpful.

As selection involves clicking and dragging the mouse, all regions of interest must be contiguous. That is, you must be able to drag over the relevant regions in a single motion. This may mean gathering particular sequences into a block. There are two ways to achieve this:

1. Click on the name of an individual sequence and drag it to the desired location in the alignment. Do this with each relevant sequence until all those of interest are placed as desired.
2. Check the option "Show selection boxes" in the Alignment settings section of the side panel settings (figure 24.11). Click in the checkbox next to the names of the sequences you wish to select. Then right-click on the name of one of the sequences and choose the option "Sort Sequences by Marked Status". This will bring all selected sequences to the top of the alignment.

If you have many sequences to select, it can be easiest to select the few that are not of interest, and then invert the selection by right-clicking on any of the checkboxes and choosing the option "Invert All Marks".

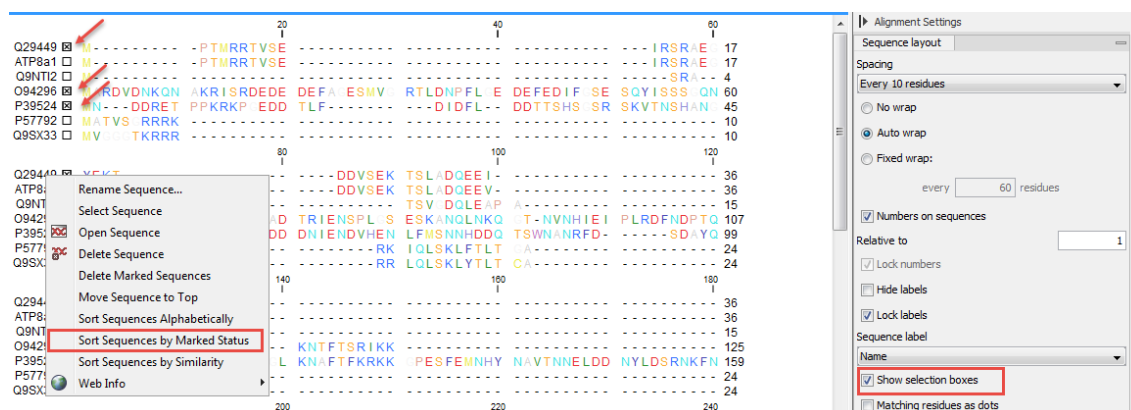


Figure 24.11: Marking and sorting sequences by marked status in an alignment

You can then easily click-and-drag your selection of sequences (this is made easier if you select the "No wrap" setting in the right-hand side panel). By right-clicking on the selected sequences (not on their names, but on the sequences themselves as seen in figure 24.12), you can choose the option "Open selection in a new view", with the ability to run any relevant tool on that sub-alignment.

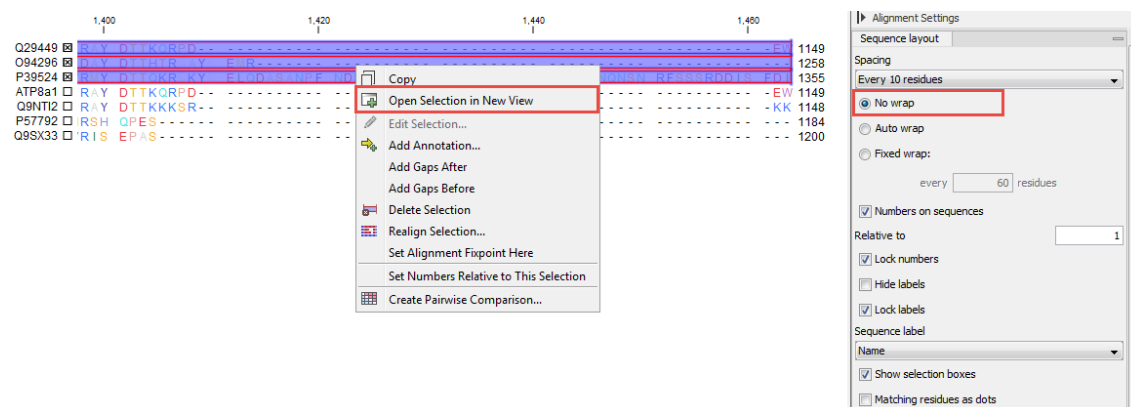


Figure 24.12: Open the selected sequences in a new window to realign them.

Realign the selected region

To realign the selected region:

Right-click the selection | Choose the option "Realign selection"

This will open a window allowing you to set the parameters for the realignment (see figure 24.13).

It is possible for an alignment to become shorter or longer as a result of the realignment of a region. This is because gaps may have to be inserted in, or deleted from, the sequences not selected for realignment. This will only occur for entire columns of gaps in these sequences, ensuring that their relative alignment is unchanged.

Learn more about the options available to you in the sections 24.1.1 and 24.1.2.

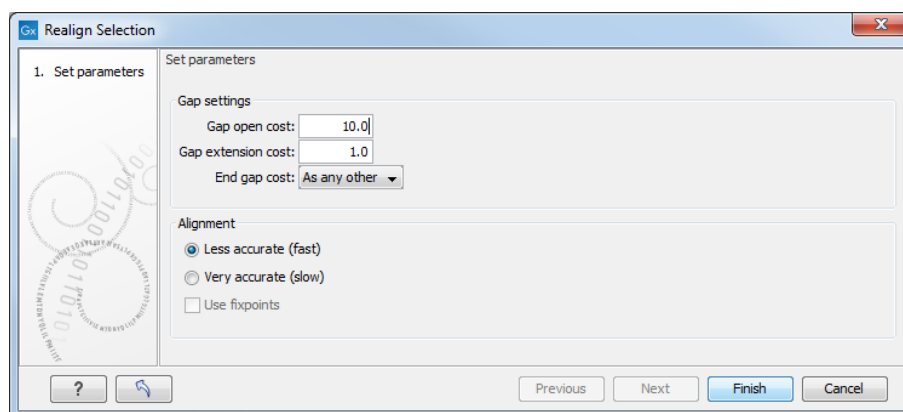


Figure 24.13: Options available when realigning a subset of an alignment.

24.4 Join alignments

CLC Genomics Workbench can join several alignments into one. This feature can for example be used to construct "supergenes" for phylogenetic inference by joining alignments of several disjoint genes into one spliced alignment. Note, that when alignments are joined, all their annotations are carried over to the new spliced alignment.

Alignments can be joined using the **Join Alignments** tool, available at:

Tools | Classical Sequence Analysis (📁) | Alignments and Trees (📁) | Join Alignments (🔗)

This opens the dialog shown in figure 24.14.

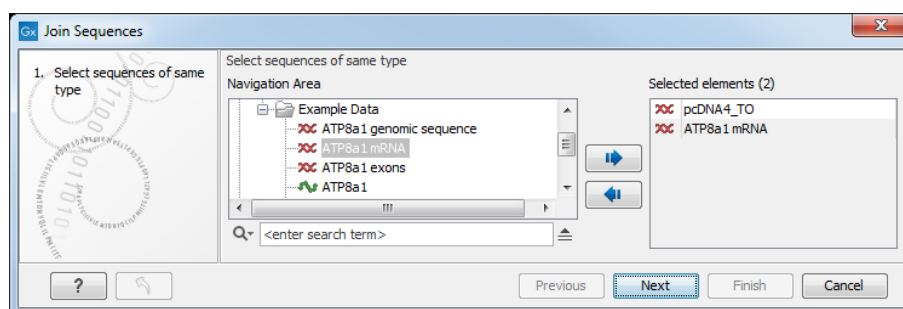


Figure 24.14: Selecting two alignments to be joined.

If you have selected some alignments before launching the tool, they will be pre-selected in the **Selected Elements** window of the dialog. Use the arrows to add or remove alignments from the selected elements. In this example seven alignments are selected. Each alignment represents one gene that have been sequenced from five different bacterial isolates from the genus *Nisseria*. Clicking **Next** opens the dialog shown in figure 24.15.

To adjust the order of concatenation, click the name of one of the alignments, and move it up or down using the arrow buttons.

The result is seen in the lower part of figure 24.16.

How alignments are joined Alignments are joined by considering the sequence names in the individual alignments. If two sequences from different alignments have identical names, they are

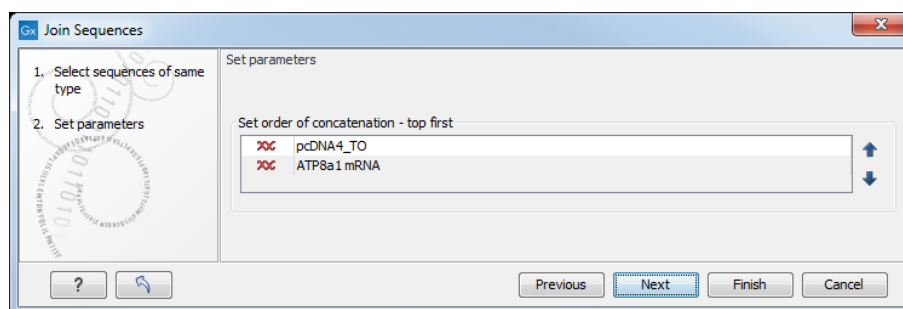


Figure 24.15: Selecting order of concatenation.

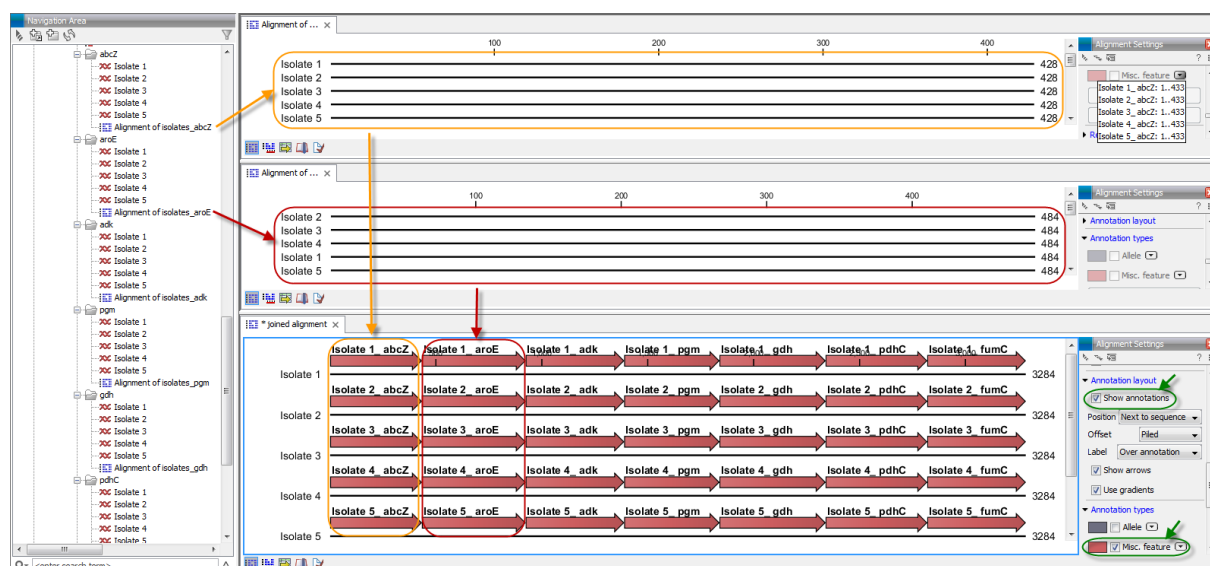


Figure 24.16: The upper part of the figure shows two of the seven alignments for the genes "abcZ" and "aroE" respectively. Each alignment consists of sequences from one gene from five different isolates. The lower part of the figure shows the result of "Join Alignments". Seven genes have been joined to an artificial gene fusion, which can be useful for construction of phylogenetic trees in cases where only fractions of a genome is available. Joining of the alignments results in one row for each isolate consisting of seven fused genes. Each fused gene sequence corresponds to the number of uniquely named sequences in the joined alignments.

considered to have the same origin and are thus joined. Consider the joining of the alignments shown in figure 24.16 "Alignment of isolates_abcZ", "Alignment of isolates_aroE", "Alignment of isolates_adk" etc. If a sequence with the same name is found in the different alignments (in this case the name of the isolates: Isolate 1, Isolate 2, Isolate 3, Isolate 4, and Isolate 5), a joined alignment will exist for each sequence name. In the joined alignment the selected alignments will be fused with each other in the order they were selected (in this case the seven different genes from the five bacterial isolates). Note that annotations have been added to each individual sequence before aligning the isolates for one gene at the time in order to make it clear which sequences were fused to each other.

24.5 Pairwise comparison

For a given set of aligned sequences it is possible to make a pairwise comparison in which each pair of sequences are compared to each other. This provides an overview of the diversity among

the sequences in the alignment.

In *CLC Genomics Workbench* this is done by creating a comparison table:

Tools | Classical Sequence Analysis (📁) | Alignments and Trees (📁) | Create Pairwise Comparison (📊)

This opens the dialog displayed in figure 24.17:

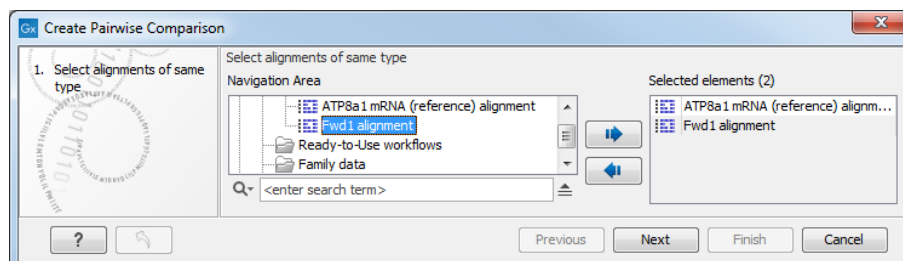


Figure 24.17: Creating a pairwise comparison table.

Select at least two alignments alignment to compare. A pairwise comparison can also be performed for a selected part of an alignment:

right-click on an alignment selection | Pairwise Comparison (📊)

There are five kinds of comparison that can be made between the sequences in the alignment, as shown in figure 24.18.

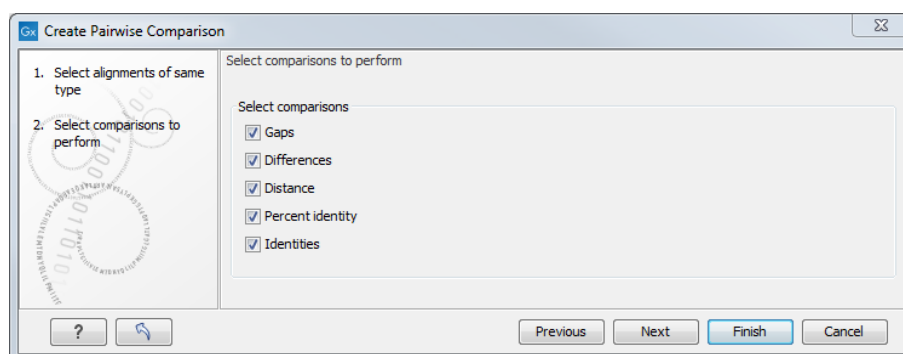


Figure 24.18: Adjusting parameters for pairwise comparison.

- **Gaps** Calculates the number of alignment positions where one sequence has a gap and the other does not.
- **Identities** Calculates the number of identical alignment positions to overlapping alignment positions between the two sequences. An overlapping alignment position is a position where at least one residue is present, rather than only gaps.
- **Differences** Calculates the number of alignment positions where one sequence is different from the other. This includes gap differences as in the Gaps comparison.
- **Distance** Calculates the Jukes-Cantor distance between the two sequences. This number is given as the Jukes-Cantor correction of the proportion between identical and overlapping alignment positions between the two sequences.

- **Percent identity** Calculates the percentage of identical residues in alignment positions to overlapping alignment positions between the two sequences.

24.5.1 The pairwise comparison table

The table shows the results of selected comparisons (see an example in figure 24.19). Since comparisons are often symmetric, the table can show the results of two comparisons at the same time, one in the upper-right and one in the lower-left triangle.

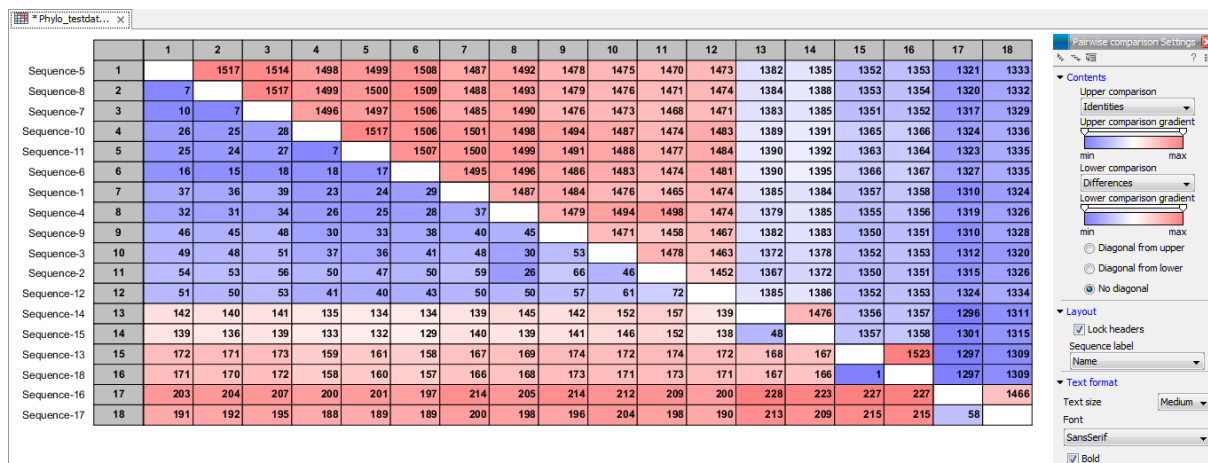


Figure 24.19: A pairwise comparison table.

Note that you can change the minimum and maximum values of the gradient coloring by sliding the corresponding cursor along the gradient in the right side panel of the comparison table. The values that appears when you slide the cursor reflect the percentage of the range of values in the table, and not absolute values.

The following settings are present in the side panel:

- **Contents**

- **Upper comparison** Selects the comparison to show in the upper triangle of the table.
- **Upper comparison gradient** Selects the color gradient to use for the upper triangle.
- **Lower comparison** Selects the comparison to show in the lower triangle. Choose the same comparison as in the upper triangle to show all the results of an asymmetric comparison.
- **Lower comparison gradient** Selects the color gradient to use for the lower triangle.
- **Diagonal from upper** Use this setting to show the diagonal results from the upper comparison.
- **Diagonal from lower** Use this setting to show the diagonal results from the lower comparison.
- **No Diagonal.** Leaves the diagonal table entries blank.

- **Layout**

- **Lock headers** Locks the sequence labels and table headers when scrolling the table.

- **Sequence label** Changes the sequence labels.
- **Text format**
 - **Text size** Changes the size of the table and the text within it.
 - **Font** Changes the font in the table.
 - **Bold** Toggles the use of boldface in the table.

24.5.2 Bioinformatics explained: Multiple alignments

Multiple alignments are at the core of bioinformatical analysis. Often the first step in a chain of bioinformatical analyses is to construct a multiple alignment of a number of homologs DNA or protein sequences. However, despite their frequent use, the development of multiple alignment algorithms remains one of the algorithmically most challenging areas in bioinformatical research.

Constructing a multiple alignment corresponds to developing a hypothesis of how a number of sequences have evolved through the processes of character substitution, insertion and deletion. The input to multiple alignment algorithms is a number of homologous sequences, i.e., sequences that share a common ancestor and most often also share molecular function. The generated alignment is a table (see figure 24.20) where each row corresponds to an input sequence and each column corresponds to a position in the alignment. An individual column in this table represents residues that have all diverged from a common ancestral residue. Gaps in the table (commonly represented by a '-') represent positions where residues have been inserted or deleted and thus do not have ancestral counterparts in all sequences.

Use of multiple alignments

Once a multiple alignment is constructed it can form the basis for a number of analyses:

- The phylogenetic relationship of the sequences can be investigated by tree-building methods based on the alignment.
- Annotation of functional domains, which may only be known for a subset of the sequences, can be transferred to aligned positions in other un-annotated sequences.
- Conserved regions in the alignment can be found which are prime candidates for holding functionally important sites.
- Comparative bioinformatical analysis can be performed to identify functionally important regions.

Constructing multiple alignments

Whereas the optimal solution to the pairwise alignment problem can be found in reasonable time, the problem of constructing a multiple alignment is much harder.

The first major challenge in the multiple alignment procedure is how to rank different alignments, i.e., which *scoring function* to use. Since the sequences have a shared history they are correlated through their *phylogeny* and the scoring function should ideally take this into account. Doing so is, however, not straightforward as it increases the number of model parameters considerably. It is therefore commonplace to either ignore this complication and assume sequences to be unrelated, or to use heuristic corrections for shared ancestry.

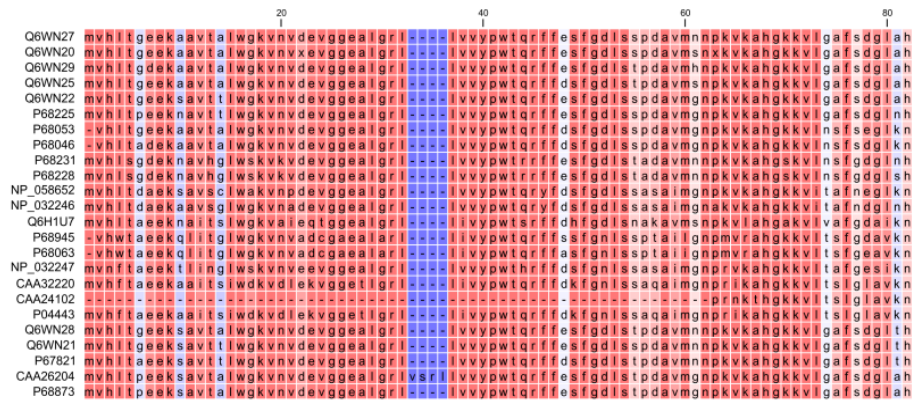


Figure 24.20: The tabular format of a multiple alignment of 24 Hemoglobin protein sequences. Sequence names appear at the beginning of each row and the residue position is indicated by the numbers at the top of the alignment columns. The level of sequence conservation is shown on a color scale with blue residues being the least conserved and red residues being the most conserved.

The second challenge is to find the optimal alignment given a scoring function. For pairs of sequences this can be done by *dynamic programming* algorithms, but for more than three sequences this approach demands too much computer time and memory to be feasible.

A commonly used approach is therefore to do *progressive alignment* [Feng and Doolittle, 1987] where multiple alignments are built through the successive construction of pairwise alignments. These algorithms provide a good compromise between time spent and the quality of the resulting alignment

The method has the inherent drawback that once two sequences are aligned, there is no way of changing their relative alignment based on the information that additional sequences may contribute later in the process. It is therefore important to make the best possible alignments early in the procedure, to avoid accumulating errors. To accomplish this, a tree of the sequences is usually constructed to guide the progressive alignment algorithm. And to overcome the problem of a time consuming tree construction step, we are using word matching, a method that group sequences in a very efficient way, saving much time, without reducing the resulting alignment accuracy significantly.

Our algorithm (developed by QIAGEN Aarhus) has two speed settings: "standard" and "fast". The **standard method** makes a fairly standard progressive alignment using the fast method of generating a guide tree. When aligning two alignments to each other, two matching columns are scored as the average of all the pairwise scores of the residues in the columns. The gap cost is affine, allowing a different cost for the first gapped position and for the consecutive gaps. This ensures that gaps are not spread out too much.

The **fast method** of alignment uses the same overall method, except that it uses fixpoints in the alignment algorithm based on short subsequences that are identical in the sequences that are being aligned. This allows similar sequences to be aligned much more efficiently, without reducing accuracy very much.

Chapter 25

Phylogenetic trees

Contents

25.1 K-mer Based Tree Construction	648
25.2 Create tree	649
25.3 Model Testing	650
25.4 Maximum Likelihood Phylogeny	652
25.4.1 Bioinformatics explained	655
25.5 Tree Settings	660
25.5.1 Minimap	661
25.5.2 Tree layout	661
25.5.3 Node settings	663
25.5.4 Label settings	663
25.5.5 Background settings	665
25.5.6 Branch layout	666
25.5.7 Bootstrap settings	666
25.5.8 Visualizing metadata	667
25.5.9 Node right click menu	667
25.6 Metadata and phylogenetic trees	669
25.6.1 Table Settings and Filtering	671
25.6.2 Add or modify metadata on a tree	671
25.6.3 Undefined metadata values on a tree	672
25.6.4 Selection of specific nodes	673

Phylogenetics describes the taxonomic classification of organisms based on their evolutionary history i.e. their phylogeny. Phylogenetics is therefore an integral part of the science of systematics that aims to establish the phylogeny of organisms based on their characteristics. Furthermore, phylogenetics is central to evolutionary biology as a whole as it is the condensation of the overall paradigm of how life arose and developed on earth. The focus of this module is the reconstruction and visualization of phylogenetic trees. Phylogenetic trees illustrate the inferred evolutionary history of a set of organisms, and makes it possible to e.g. identify groups of closely related organisms and observe clustering of organisms with common traits. See [25.4.1](#) for a more detailed introduction to phylogenetic trees.

The viewer for visualizing and working with phylogenetic trees allows the user to create high-quality, publication-ready figures of phylogenetic trees. Large trees can be explored in two alternative tree layouts; circular and radial. The viewer supports importing, editing and visualization of metadata associated with nodes in phylogenetic trees.

Below is an overview of the main features of the phylogenetic tree editor. Further details can be found in the subsequent sections.

Main features of the phylogenetic tree editor:

- Circular and radial layouts.
- Import of metadata in Excel and CSV format.
- Tabular view of metadata with support for editing.
- Options for collapsing nodes based on bootstrap values.
- Re-ordering of tree nodes.
- Legends describing metadata.
- Visualization of metadata though e.g. node color, node shape, branch color, etc.
- Minimap navigation.
- Coloring and labeling of subtrees.
- Curved edges.
- Editable node sizes and line width.
- Intelligent visualization of overlapping labels and nodes.

For a given set of aligned sequences (see section 24.1) it is possible to infer their evolutionary relationships. In *CLC Genomics Workbench* this may be done either by using a distance based method or by using maximum likelihood (ML) estimation, which is a statistical approach (see Bioinformatics explained in section 25.4.1). Both approaches generate a phylogenetic tree.

Three tools are available for generating phylogenetic trees:

- **K-mer Based Tree Construction** (🌲): Is a distance-based method that can create trees based on multiple single sequences. K-mers are used to compute distance matrices for distance-based phylogenetic reconstruction tools such as neighbor joining and UPGMA (see section 25.4.1). This method is less precise than the Create Tree tool but it can cope with a very large number of long sequences as it does not require a multiple alignment. The k-mer based tree construction tool is especially useful for whole genome phylogenetic reconstruction where the genomes are closely related, i.e. they differ mainly by SNPs and contain no or few structural variations.
- **Maximum Likelihood Phylogeny** (🌳): The most advanced and time consuming method of the three mentioned. The maximum likelihood tree estimation is performed under the assumption of one of five substitution models: the Jukes-Cantor, the Kimura 80, the HKY and the GTR (also known as the REV model) models (see section 25.4 for further information

about the models). Prior to using the Maximum Likelihood Phylogeny tool for creating a phylogenetic tree it is recommended to run the Model Testing tool (see section 25.3) in order to identify the best suitable models for creating a tree.

- **Create Tree** (🔗) Is a tool that uses distance estimates computed from a multiple sequence alignment to create a tree. The user can select whether to use Jukes-Cantor distance correction or Kimura distance correction (Kimura 80 for nucleotides/Kimura protein for proteins) in combination with either the neighbor joining or UPGMA method (see section 25.4.1).

25.1 K-mer Based Tree Construction

The K-mer Based Tree Construction tool uses single sequences or sequence lists as input and is the simplest way of creating a distance-based phylogenetic tree. To run the K-mer Based Tree Construction tool:

Tools | Classical Sequence Analysis (🔗) | Alignments and Trees (🔗) | K-mer Based Tree Construction (🔗)

Select sequences or a sequence list (figure 25.1):

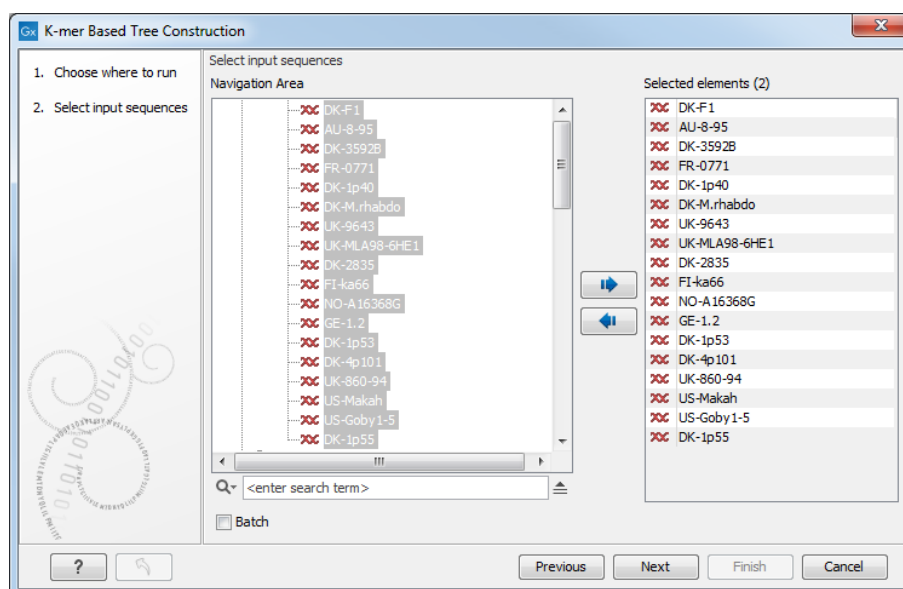


Figure 25.1: Select sequences needed for creating a tree with K-mer based tree construction.

Next, select the construction method, specify the k-mer length and select a distance measure for tree construction (figure 25.2):

• Tree construction

- **Tree construction method** The user is asked to specify which distance-based method to use for tree construction. There are two options (see section 25.4.1):

- * The **UPGMA** method. Assumes constant rate of evolution.
- * The **Neighbor Joining** method. Well suited for trees with varying rates of evolution.

• K-mer settings

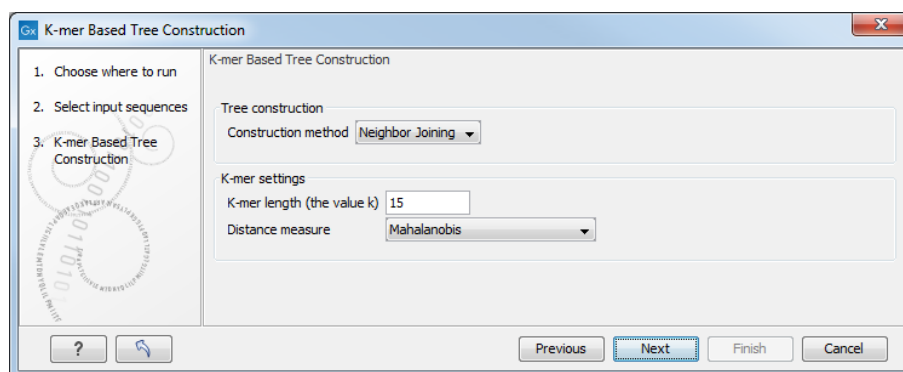


Figure 25.2: Select the construction method, and specify the k-mer length and a distance measure.

- **K-mer length (the value k)** Allows specification of the k-mer length, which can be a number between 3 and 50.
- **Distance measure** The distance measure is used to compute the distances between two counts of k-mers. Three options exist: Euclidian squared, Mahalanobis, and Fractional common K-mer count. See section 25.4.1 for further details.

25.2 Create tree

The Create tree tool can be used to generate a distance-based phylogenetic tree from an input alignment:

Tools | Classical Sequence Analysis (🔍) | Alignments and Trees (📁) | Create Tree (🌳)

This will open the dialog displayed in figure 25.3:

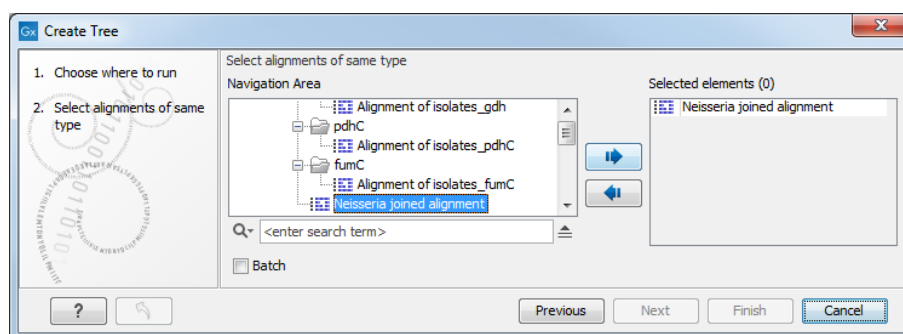


Figure 25.3: Creating a tree.

If an alignment was selected before the tool was launched, that alignment will be listed in the **Selected Elements** window of the dialog. Use the arrows to add or remove elements from the **Navigation Area**. Click **Next** to adjust parameters.

Figure 25.4 shows the parameters that can be set for this distance-based tree creation:

- Tree construction (see section 25.4.1)
 - Tree construction method
 - * The **UPGMA** method. Assumes constant rate of evolution.

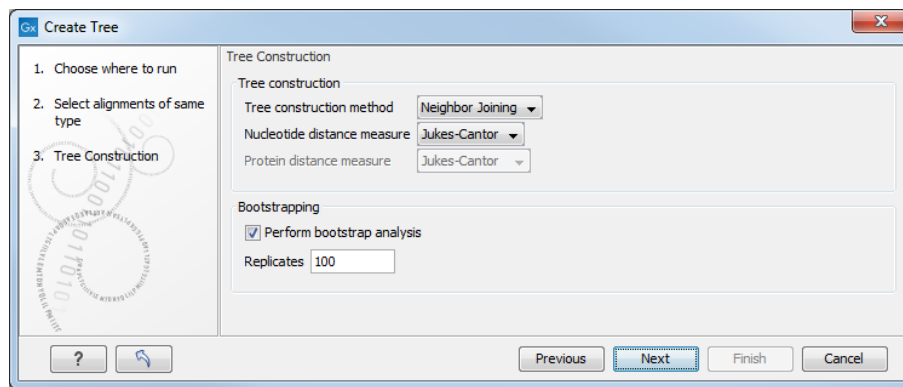


Figure 25.4: Adjusting parameters for distance-based methods.

- * The **Neighbor Joining** method. Well suited for trees with varying rates of evolution.
- Nucleotide distance measure
 - * **Jukes-Cantor**. Assumes equal base frequencies and equal substitution rates.
 - * **Kimura 80**. Assumes equal base frequencies but distinguishes between transitions and transversions.
- Protein distance measure
 - * **Jukes-Cantor**. Assumes equal amino acid frequency and equal substitution rates.
 - * **Kimura protein**. Assumes equal amino acid frequency and equal substitution rates. Includes a small correction term in the distance formula that is intended to give better distance estimates than Jukes-Cantor.
- Bootstrapping.
 - Perform bootstrap analysis. To evaluate the reliability of the inferred trees, *CLC Genomics Workbench* allows the option of doing a **bootstrap** analysis (see section 25.4.1). A bootstrap value will be attached to each node, and this value is a measure of the confidence in the subtree rooted at the node. The number of replicates used in the bootstrap analysis can be adjusted in the wizard. The default value is 100 replicates which is usually enough to distinguish between reliable and unreliable nodes in the tree. The bootstrap value assigned to each inner node in the output tree is the percentage (0-100) of replicates which contained the same subtree as the one rooted at the inner node.

For a more detailed explanation, see Bioinformatics explained in section 25.4.1.

25.3 Model Testing

As the Model Testing tool can help identify the best substitution model (25.4.1) to be used for Maximum Likelihood Phylogeny tree construction, it is recommended to run Model Testing before running the Maximum Likelihood Phylogeny tool.

The Model Testing tool uses four different statistical analyses:

- Hierarchical likelihood ratio test (hLRT)
- Bayesian information criterion (BIC)

- Minimum theoretical information criterion (AIC)
- Minimum corrected theoretical information criterion (AICc)

to test the substitution models:

- Jukes-Cantor [Jukes and Cantor, 1969]
- Felsenstein 81 [Felsenstein, 1981]
- Kimura 80 [Kimura, 1980]
- HKY [Hasegawa et al., 1985]
- GTR (also known as the REV model) [Yang, 1994a]

To do model testing:

Tools | Classical Sequence Analysis (📁) | Alignments and Trees (📁) | Model Testing (📁)

Select the alignment that you wish to use for the tree construction (figure 25.5):

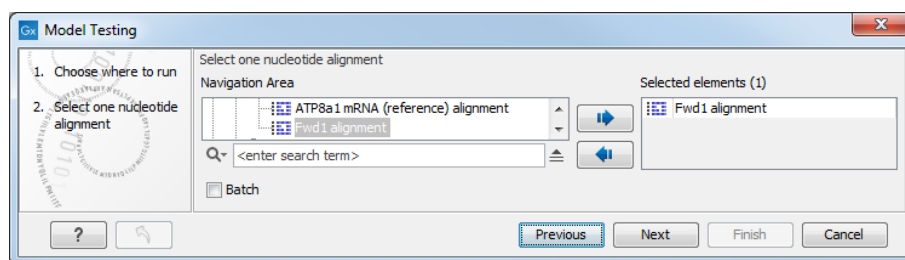


Figure 25.5: Select alignment for model testing.

Specify the parameters to be used for model testing (figure 25.6):

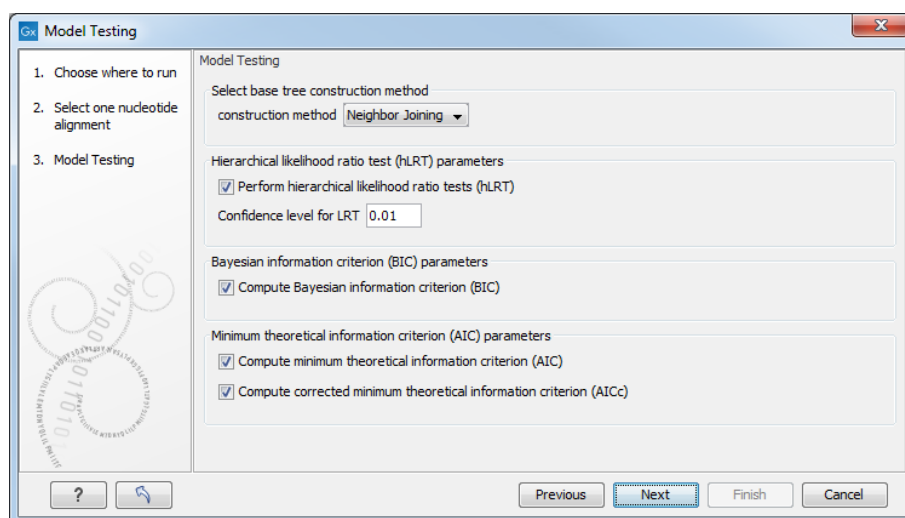


Figure 25.6: Specify parameters for model testing.

- **Select base tree construction method**

A base tree (a guiding tree) is required in order to be able to determine which model(s) would be the most appropriate to use to make the best possible phylogenetic tree from a specific alignment. The topology of the base tree is used in the hierarchical likelihood ratio test (hLRT), and the base tree is used as starting point for topology exploration in Bayesian information criterion (BIC), Akaike information criterion (or minimum theoretical information criterion) (AIC), and AICc (AIC with a correction for the sample size) ranking.

- **Construction method** A base tree is created automatically using one of two methods from the Create Tree tool:
 - * The **UPGMA** method. Assumes constant rate of evolution.
 - * The **Neighbor Joining** method. Well suited for trees with varying rates of evolution.
- **Hierarchical likelihood ratio test (hLRT) parameters** A statistical test of the goodness-of-fit between two models that compares a relatively more complex model to a simpler model to see if it fits a particular dataset significantly better.
 - **Perform hierarchical likelihood ratio test (hLRT)**
 - **Confidence level for LRT** The confidence level used in the likelihood ratio tests.
- **Bayesian information criterion (BIC) parameters**
 - **Compute Bayesian information criterion (BIC)** Rank substitution models based on Bayesian information criterion (BIC). Formula used is $BIC = -2\ln(L) + K\ln(n)$, where $\ln(L)$ is the log-likelihood of the best tree, K is the number of parameters in the model, and $\ln(n)$ is the logarithm of the length of the alignment.
- **Minimum theoretical information criterion (AIC) parameters**
 - **Compute minimum theoretical information criterion (AIC)** Rank substitution models based on minimum theoretical information criterion (AIC). Formula used is $AIC = -2\ln(L) + 2K$, where $\ln(L)$ is the log-likelihood of the best tree, K is the number of parameters in the model.
 - **Compute corrected minimum theoretical information criterion (AICc)** Rank substitution models based on minimum corrected theoretical information criterion (AICc). Formula used is $AICc = -2\ln(L) + 2K + 2K(K+1)/(n-K-1)$, where $\ln(L)$ is the log-likelihood of the best tree, K is the number of parameters in the model, n is the length of the alignment. AICc is recommended over AIC roughly when n/K is less than 40.

The output from model testing is a report that lists all test results in table format. For each tested model the report indicate whether it is recommended to use rate variation or not. Topology variation is recommended in all cases.

From the listed test results, it is up to the user to select the most appropriate model. The different statistical tests will usually agree on which models to recommend although variations may occur. Hence, in order to select the best possible model, it is recommended to select the model that has proven to be the best by most tests.

25.4 Maximum Likelihood Phylogeny

To generate a maximum likelihood based phylogenetic tree, go to:

Tools | Classical Sequence Analysis (🔍) | Alignments and Trees (📁) | Maximum Likelihood Phylogeny (🔍)

First, select the alignment to be used for the reconstruction (figure 25.7).

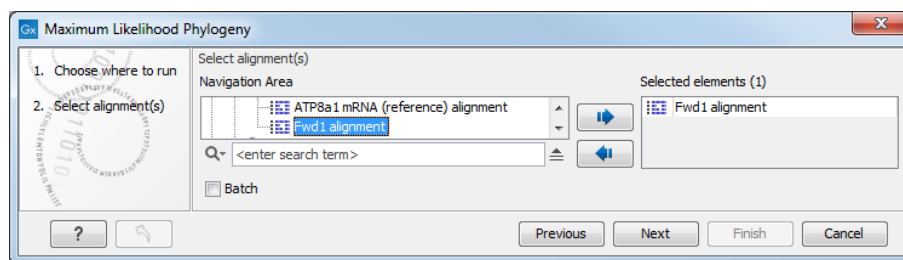


Figure 25.7: Select the alignment for tree construction.

You can then set up the following parameters (figure 25.8):

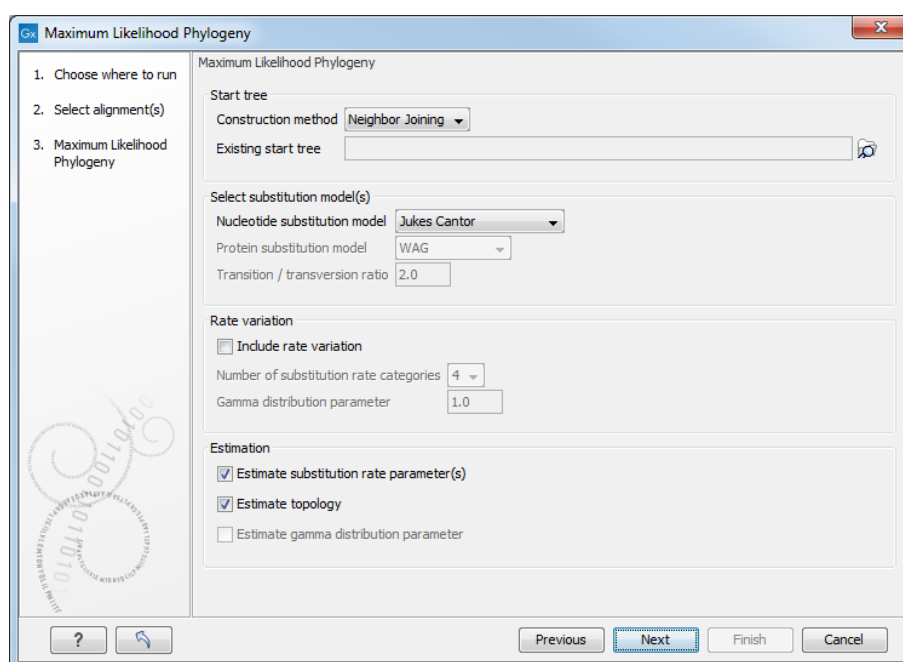


Figure 25.8: Adjusting parameters for maximum likelihood phylogeny

- **Start tree**

- **Construction method** Specify the tree construction method which should be used to create the initial tree, Neighbor Joining or UPGMA
- **Existing start tree** Alternatively, an existing tree can be used as starting tree for the tree reconstruction. Click on the folder icon to the right of the text field to specify the desired starting tree.

- **Select substitution model**

- **Nucleotide substitution model** CLC Genomics Workbench allows maximum likelihood tree estimation to be performed under the assumption of one of five nucleotide substitution models:

- * Jukes-Cantor [Jukes and Cantor, 1969]
- * Felsenstein 81 [Felsenstein, 1981]
- * Kimura 80 [Kimura, 1980]
- * HKY [Hasegawa et al., 1985]
- * General Time Reversible (GTR) (also known as the REV model) [Yang, 1994a]

All models are time-reversible. In the Kimura 80 and HKY models, the user may set a transition/transversion ratio value, which will be used as starting value for optimization or as a fixed value, depending on the level of estimation chosen by the user. For further details, see 25.4.1.

- **Protein substitution model** *CLC Genomics Workbench* allows maximum likelihood tree estimation to be performed under the assumption of one of four protein substitution models:

- * Bishop-Friday [Bishop and Friday, 1985]
- * Dayhoff (PAM) [Dayhoff et al., 1978]
- * JTT [Jones et al., 1992]
- * WAG [Whelan and Goldman, 2001]

The Bishop-Friday substitution model is similar to the Jukes-Cantor model for nucleotide sequences, i.e. it assumes equal amino acid frequencies and substitution rates. This is an unrealistic assumption and we therefore recommend using one of the remaining three models. The Dayhoff, JTT and WAG substitution models are all based on large scale experiments where amino acid frequencies and substitution rates have been estimated by aligning thousands of protein sequences. For these models, the maximum likelihood tool does not estimate parameters, but simply uses those determined from these experiments.

- **Rate variation**

To enable variable substitution rates among individual nucleotide sites in the alignment, select the **include rate variation** box. When selected, the discrete gamma model of Yang [Yang, 1994b] is used to model rate variation among sites. The number of categories used in the discretization of the gamma distribution as well as the gamma distribution parameter may be adjusted by the user (as the gamma distribution is restricted to have mean 1, there is only one parameter in the distribution).

- **Estimation**

Estimation is done according to the maximum likelihood principle, that is, a search is performed for the values of the free parameters in the model assumed that results in the highest likelihood of the observed alignment [Felsenstein, 1981]. By ticking the **Estimate substitution rate parameters** box, maximum likelihood values of the free parameters in the rate matrix describing the assumed substitution model are found. If the **Estimate topology** box is selected, a search in the space of tree topologies for that which best explains the alignment is performed. If left un-ticked, the starting topology is kept fixed at that of the starting tree.

The **Estimate Gamma distribution parameter** is active if rate variation has been included in the model and in this case allows estimation of the Gamma distribution parameter to be switched on or off. If the box is left un-ticked, the value is fixed at that given in the **Rate variation** part. In the absence of rate variation estimation of substitution

parameters and branch lengths are carried out according to the expectation maximization algorithm [Dempster et al., 1977]. With rate variation the maximization algorithm is performed. The topology space is searched according to the PHYML method [Guindon and Gascuel, 2003], allowing efficient search and estimation of large phylogenies. **Branch lengths are given in terms of expected numbers of substitutions per nucleotide site.**

In the next step of the wizard it is possible to perform bootstrapping (figure 25.9).

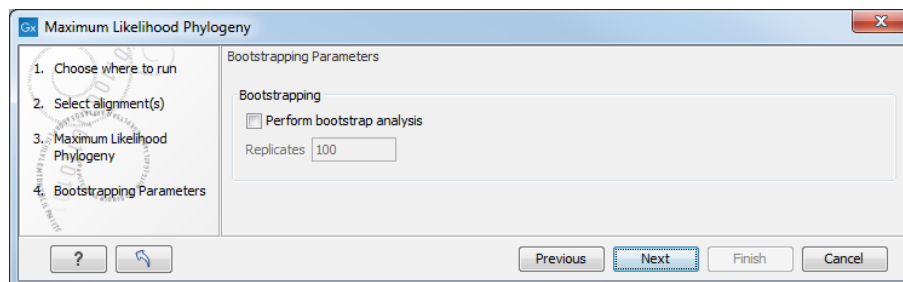


Figure 25.9: Adjusting parameters for ML phylogeny

To evaluate the reliability of the inferred trees, *CLC Genomics Workbench* allows the option of doing a **bootstrap** analysis (see section 25.4.1). A bootstrap value will be attached to each node, and this value is a measure of the confidence in the subtree rooted at the node. The number of replicates in the bootstrap analysis can be adjusted in the wizard by specifying the number of times to resample the data. The default value is 100 resamples. The bootstrap value assigned to a node in the output tree is the percentage (0-100) of the bootstrap resamples which resulted in a tree containing the same subtree as that rooted at the node.

25.4.1 Bioinformatics explained

The phylogenetic tree

The evolutionary hypothesis of a phylogeny can be graphically represented by a phylogenetic tree.

Figure 25.10 shows a proposed phylogeny for the great apes, *Hominidae*, taken in part from Purvis [Purvis, 1995]. The tree consists of a number of nodes (also termed vertices) and branches (also termed edges). These nodes can represent either an individual, a species, or a higher grouping and are thus broadly termed taxonomic units. In this case, the terminal nodes (also called leaves or tips of the tree) represent extant species of *Hominidae* and are the *operational taxonomic units* (OTUs). The internal nodes, which here represent extinct common ancestors of the great apes, are termed *hypothetical taxonomic units* since they are not directly observable.

The ordering of the nodes determine the tree *topology* and describes how lineages have diverged over the course of evolution. The branches of the tree represent the amount of evolutionary divergence between two nodes in the tree and can be based on different measurements. A tree is completely specified by its topology and the set of all edge lengths.

The phylogenetic tree in figure 25.10 is rooted at the most recent common ancestor of all *Hominidae* species, and therefore represents a hypothesis of the direction of evolution e.g. that the common ancestor of gorilla, chimpanzee and man existed before the common ancestor of chimpanzee and man. In contrast, an unrooted tree would represent relationships without assumptions about ancestry.

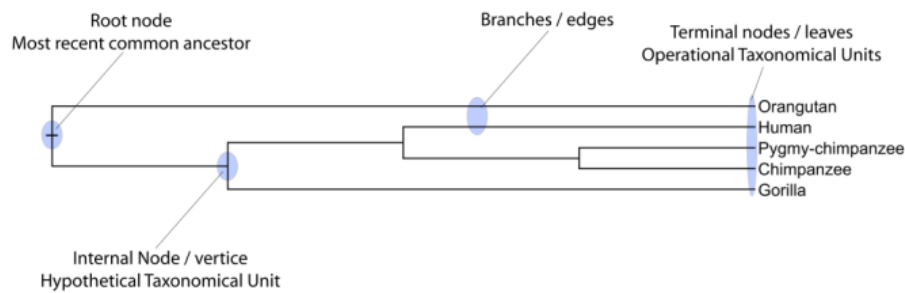


Figure 25.10: A proposed phylogeny of the great apes (Hominidae). Different components of the tree are marked, see text for description.

Besides evolutionary biology and systematics the inference of phylogenies is central to other areas of research.

As more and more genetic diversity is being revealed through the completion of multiple genomes, an active area of research within bioinformatics is the development of comparative machine learning algorithms that can simultaneously process data from multiple species [Siepel and Haussler, 2004]. Through the comparative approach, valuable evolutionary information can be obtained about which amino acid substitutions are functionally tolerant to the organism and which are not. This information can be used to identify substitutions that affect protein function and stability, and is of major importance to the study of proteins [Knudsen and Miyamoto, 2001]. Knowledge of the underlying phylogeny is, however, paramount to comparative methods of inference as the phylogeny describes the underlying correlation from shared history that exists between data from different species.

In molecular epidemiology of infectious diseases, phylogenetic inference is also an important tool. The very fast substitution rate of microorganisms, especially the RNA viruses, means that these show substantial genetic divergence over the time-scale of months and years. Therefore, the phylogenetic relationship between the pathogens from individuals in an epidemic can be resolved and contribute valuable epidemiological information about transmission chains and epidemiologically significant events [Leitner and Albert, 1999], [Forsberg et al., 2001].

Substitution models and distance estimation

When estimating the evolutionary distance between organisms, one needs a model of how frequently different mutations occur in the DNA. Such models are known as substitution models. Our Model Testing and Maximum Likelihood Phylogeny tools currently support the five nucleotide substitution models listed here:

- Jukes-Cantor [Jukes and Cantor, 1969]
- Felsenstein 81 [Felsenstein, 1981]
- Kimura 80 [Kimura, 1980]
- HKY [Hasegawa et al., 1985]
- GTR (also known as the REV model) [Yang, 1994a]

Common to all these models is that they assume mutations at different sites in the genome occur independently and that the mutations at each site follow the same common probability

distribution. Thus all five models provide relative frequencies for each of the 16 possible DNA substitutions (e.g. $C \rightarrow A$, $C \rightarrow C$, $C \rightarrow G$,...).

The Jukes-Cantor and Kimura 80 models assume equal base frequencies and the HKY and GTR models allow the frequencies of the four bases to differ (they will be estimated by the observed frequencies of the bases in the alignment). In the Jukes-Cantor model all substitutions are assumed to occur at equal rates, in the Kimura 80 and HKY models transition and transversion rates are allowed to differ (substitution between two purines ($A \leftrightarrow G$) or two pyrimidines ($C \leftrightarrow T$) are transitions and purine - pyrimidine substitutions are transversions). The GTR model is the general time reversible model that allows all substitutions to occur at different rates. For the substitution rate matrices describing the substitution models we use the parametrization of Yang [Yang, 1994a].

For protein sequences, our Maximum Likelihood Phylogeny tool supports four substitution models:

- Bishop-Friday [Bishop and Friday, 1985]
- Dayhoff (PAM) [Dayhoff et al., 1978]
- JTT [Jones et al., 1992]
- WAG [Whelan and Goldman, 2001]

As with nucleotide substitution models, it is assumed that mutations at different sites in the genome occur independently and according to the same probability distribution.

The Bishop-Friday model assumes all amino acids occur with same frequency and that all substitutions are equally likely. This is the simplest model, but also the most unrealistic. The remaining three models use amino acid frequencies and substitution rates which have been determined from large scale experiments where huge sets of protein sequences have been aligned and rates have been estimated. These three models reflect the outcome of three different experiments. We recommend using WAG as these rates were estimated from the largest experiment.

K-mer based distance estimation

K-mer based distance estimation is an alternative to estimating evolutionary distance based on multiple alignments. At a high level, the distance between two sequences is defined by first collecting the set of k-mers (subsequences of length k) occurring in the two sequences. From these two sets, the evolutionary distance between the two organisms is now defined by measuring how different the two sets are. The more the two sets look alike, the smaller is the evolutionary distance. The main motivation for estimating evolutionary distance based on k-mers, is that it is computationally much faster than first constructing a multiple alignment. Experiments show that phylogenetic tree reconstruction using k-mer based distances can produce results comparable to the slower multiple alignment based methods [Blaisdell, 1989].

All of the k-mer based distance measures completely ignores the ordering of the k-mers inside the input sequences. Hence, if the selected k value (the length of the sequences) is too small, very distantly related organisms may be assigned a small evolutionary distance (in the extreme case where k is 1, two organisms will be treated as being identical if the frequency of each nucleotide/amino-acid is the same in the two corresponding sequences). In the other extreme,

the k-mers should have a length (k) that is somewhat below the average distance between mismatches if the input sequences were aligned (in the extreme case of k=the length of the sequences, two organisms have a maximum distance if they are not identical). Thus the selected k value should not be too large and not too small. A general rule of thumb is to only use k-mer based distance estimation for organisms that are not too distantly related.

Formal definition of distance. In the following, we give a more formal definition of the three supported distance measures: Euclidian-squared, Mahalanobis and Fractional common k-mer count. For all three, we first associate a point $p(s)$ to every input sequence s . Each point $p(s)$ has one coordinate for every possible length k sequence (e.g. if s represent nucleotide sequences, then $p(s)$ has 4^k coordinates). The coordinate corresponding to a length k sequence x has the value: "number of times x occurs as a subsequence in s ". Now for two sequences s_1 and s_2 , their evolutionary distance is defined as follows:

- **Euclidian squared:** For this measure, the distance is simply defined as the (squared Euclidian) distance between the two points $p(s_1)$ and $p(s_2)$, i.e.

$$\text{dist}(s_1, s_2) = \sum_i (p(s_1)_i - p(s_2)_i)^2.$$

- **Mahalanobis:** This measure is essentially a fine-tuned version of the Euclidian squared distance measure. Here all the counts $p(s_j)_i$ are "normalized" by dividing with the standard deviation σ_j of the count for the k-mer. The revised formula thus becomes:

$$\text{dist}(s_1, s_2) = \sum_i (p(s_1)_i/\sigma_1 - p(s_2)_i/\sigma_2)^2.$$

Here the standard deviations can be computed directly from a set of equilibrium frequencies for the different bases, see [Gentleman and Mullin, 1989].

- **Fractional common k-mer count:** For the last measure, the distance is computed based on the minimum count of every k-mer in the two sequences, thus if two sequences are very different, the minimums will all be small. The formula is as follows:

$$\text{dist}(s_1, s_2) = \log(0.1 + \sum_i (\min(p(s_1)_i, p(s_2)_i) / (\min(n, m) - k + 1))).$$

Here n is the length of s_1 and m is the length of s_2 . This method has been described in [Edgar, 2004].

In experiments performed in [Höhl et al., 2007], the Mahalanobis distance measure seemed to be the best performing of the three supported measures.

Distance based reconstruction methods

Distance based phylogenetic reconstruction methods use a pairwise distance estimate between the input organisms to reconstruct trees. The distances are an estimate of the evolutionary distance between each pair of organisms which are usually computed from DNA or amino acid sequences. Given two homologous sequences a distance estimate can be computed by aligning the sequences and then counting the number of positions where the sequences differ. The number of differences is called the observed number of substitutions and is usually an

underestimate of the real distance as multiple mutations could have occurred at any position. To correct for these hidden substitutions a substitution model, such as Jukes-Cantor or Kimura 80, can be used to get a more precise distance estimate (see section 25.4.1).

To correct for these hidden substitutions a substitution model, such as Jukes-Cantor or Kimura 80, can be used to get a more precise distance estimate.

Alternatively, k-mer based methods or SNP based methods can be used to get a distance estimate without the use of substitution models.

After distance estimates have been computed, a phylogenetic tree can be reconstructed using a distance based reconstruction method. Most distance based methods perform a bottom up reconstruction using a greedy clustering algorithm. Initially, each input organism is put in its own cluster which corresponds to a leaf node in the resulting tree. Next, pairs of clusters are iteratively joined into higher level clusters, which correspond to connecting two nodes in the tree with a new parent node. When a single node remains, the tree is reconstructed.

The *CLC Genomics Workbench* provides two of the most widely used distance based reconstruction methods:

- The **UPGMA** method [Michener and Sokal, 1957] which assumes a constant rate of evolution (molecular clock hypothesis) in the different lineages. This method reconstructs trees by iteratively joining the two nearest clusters until there is only one cluster left. The result of the UPGMA method is a rooted bifurcating tree annotated with branch lengths.
- The **Neighbor Joining** method [Saitou and Nei, 1987] attempts to reconstruct a minimum evolution tree (a tree where the sum of all branch lengths is minimized). Opposite to the UPGMA method, the neighbor joining method is well suited for trees with varying rates of evolution in different lineages. A tree is reconstructed by iteratively joining clusters which are close to each other but at the same time far from all other clusters. The resulting tree is a bifurcating tree with branch lengths. Since no particular biological hypothesis is made about the placement of the root in this method, the resulting tree is unrooted.

Maximum Likelihood reconstruction methods

Maximum Likelihood (ML) based reconstruction methods [Felsenstein, 1981] seek to identify the most probable tree given the data available, i.e. maximize $P(\text{tree}|\text{data})$ where the *tree* refers to a tree topology with branch lengths while *data* is usually a set of sequences. However, it is not possible to compute $P(\text{tree}|\text{data})$ so instead ML based methods have to compute the probability of the data given a tree, i.e. $P(\text{data}|\text{tree})$. The ML tree is then the tree which makes the data most probable. In other words, ML methods search for the tree that gives the highest probability of producing the observed sequences. This is done by searching through the space of all possible trees while computing an ML estimate for each tree. Computing an ML estimate for a tree is time consuming and since the number of tree topologies grows exponentially with the number of leaves in a tree, it is infeasible to explore all possible topologies. Consequently, ML methods must employ search heuristics that quickly converge towards a tree with a likelihood close to the real ML tree.

The likelihood of trees are computed using an explicit model of evolution such as the Jukes-Cantor or Kimura 80 models. Choosing the right model is often important to get a good result. To help users choose the correct model for a data set, the Model Testing tool (see section 25.3) can be

used to test a range of different models for input nucleotide sequences.

The search heuristics which are commonly used in ML methods requires an initial phylogenetic tree as a starting point for the search. An initial tree which is close to the optimal solution, can reduce the running time of ML methods and improve the chance of finding a tree with a large likelihood. A common way of reconstructing a good initial tree is to use a distance based method such as UPGMA or neighbor-joining to produce a tree based on a multiple alignment.

Bootstrap tests

Bootstrap tests [Felsenstein, 1985] is one of the most common ways to evaluate the reliability of the topology of a phylogenetic tree. In a bootstrap test, trees are evaluated using Efron's re-sampling technique [Efron, 1982], which samples nucleotides from the original set of sequences as follows:

Given an alignment of n sequences (rows) of length l (columns), we randomly choose l columns in the alignment with replacement and use them to create a new alignment. The new alignment has n rows and l columns just like the original alignment but it may contain duplicate columns and some columns in the original alignment may not be included in the new alignment. From this new alignment we reconstruct the corresponding tree and compare it to the original tree. For each subtree in the original tree we search for the same subtree in the new tree and add a score of one to the node at the root of the subtree if the subtree is present in the new tree. This procedure is repeated a number of times (usually around 100 times). The result is a counter for each interior node of the original tree, which indicate how likely it is to observe the exact same subtree when the input sequences are sampled. A bootstrap value is then computed for each interior node as the percentage of resampled trees that contained the same subtree as that rooted at the node.

Bootstrap values can be seen as a measure of how reliably we can reconstruct a tree, given the sequence data available. If all trees reconstructed from resampled sequence data have very different topologies, then most bootstrap values will be low, which is a strong indication that the topology of the original tree cannot be trusted.

Scale bar

The scale bar unit depends on the distance measure used and the tree construction algorithm used. The trees produced using the Maximum Likelihood Phylogeny tool has a very specific interpretation: A distance of x means that the expected number of substitutions/changes per nucleotide (amino acid for protein sequences) is x . i.e. if the distance between two taxa is 0.01, you expected a change in each nucleotide independently with probability 1 %. For the remaining algorithms, there is not as nice an interpretation. The distance depends on the weight given to different mutations as specified by the distance measure.

25.5 Tree Settings

The Tree Settings Side Panel found in the left side of the view area can be used to adjust the tree layout and to visualize metadata that is associated with the tree nodes. The following section describes the visualization options available from the Tree Settings side panel. Note however that editing legend boxes related to metadata can be done directly from editing the metadata

table (see section 25.6).

The **preferred tree layout settings** (user defined tree settings) can be saved and applied via the top right **Save Tree Settings** (figure 25.11). Settings can either be saved **For This Tree Only** or for all saved phylogenetic trees (**For Tree View in General**). The first option will save the layout of the tree for that tree only and it ensures that the layout is preserved even if it is exported and opened by a different user. The second option stores the layout globally in the Workbench and makes it available to other trees through the **Apply Saved Settings** option.

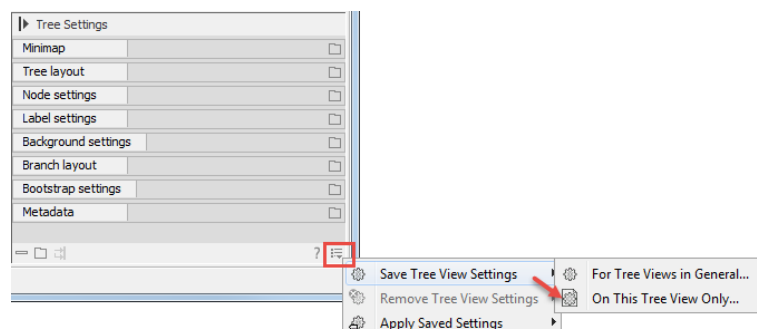


Figure 25.11: Save, remove or apply preferred layout settings.

25.5.1 Minimap

The Minimap is a navigation tool that shows a small version of the tree. A grey square indicates the specific part of the tree that is visible in the View Area (figure 25.12). To navigate the tree using the Minimap, click on the Minimap with the mouse and move the grey square around within the Minimap.

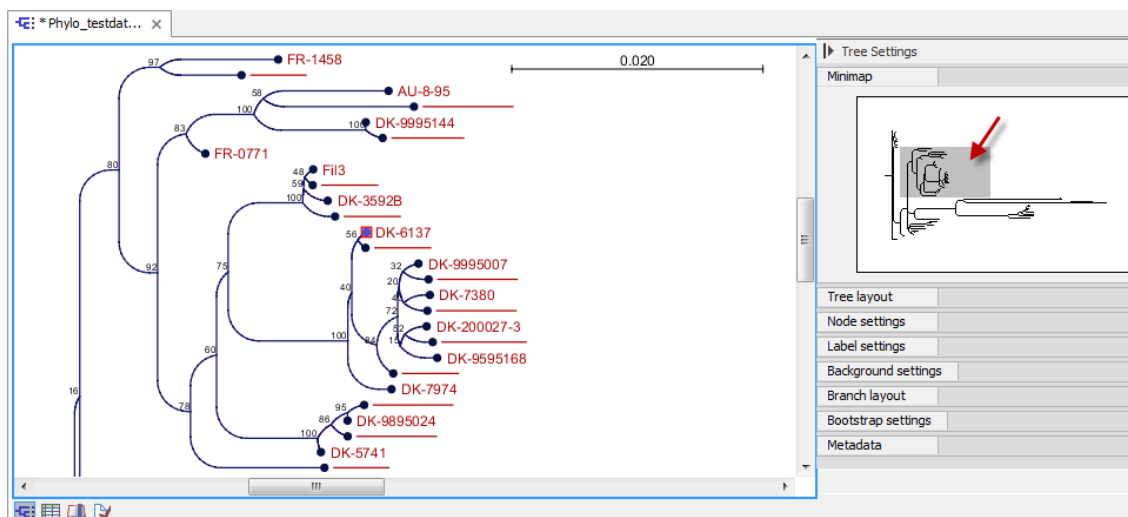


Figure 25.12: Visualization of a phylogenetic tree. The grey square in the Minimap shows the part of the tree that is shown in the View Area.

25.5.2 Tree layout

The **Tree Layout** can be adjusted in the Side Panel (figure 25.13).

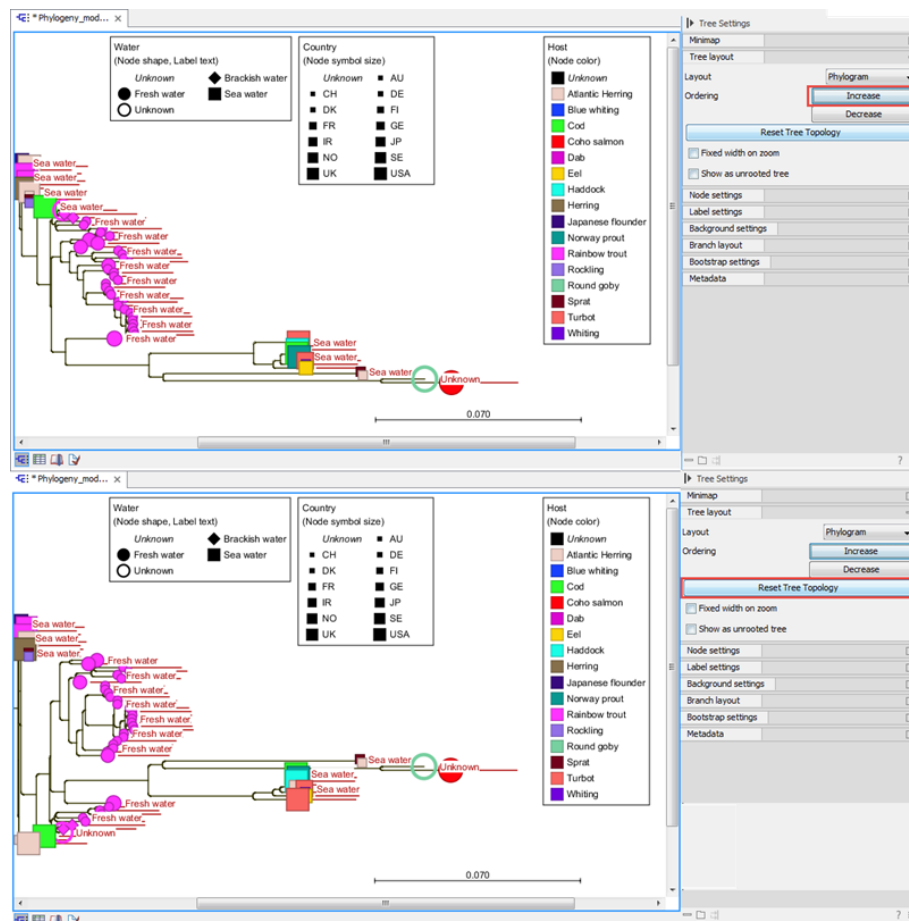


Figure 25.13: The tree layout can be adjusted in the Side Panel. The top part of the figure shows a tree with increasing node order. In the bottom part of the figure the tree has been reverted to the original tree topology.

- **Layout** Selects one of the five layout types: Phylogram, Cladogram, Circular Phylogram, Circular Cladogram or Radial. Note that only the Cladogram layouts are available if all branches in the tree have zero length.
 - **Phylogram** is a rooted tree where the edges have "lengths", usually proportional to the inferred amount of evolutionary change to have occurred along each branch.
 - **Cladogram** is a rooted tree without branch lengths which is useful for visualizing the topology of trees.
 - **Circular Phylogram** is also a phylogram but with the leaves in a circular layout.
 - **Circular Cladogram** is also a cladogram but with the leaves in a circular layout.
 - **Radial** is an unrooted tree that has the same topology and branch lengths as the rooted styles, but lacks any indication of evolutionary direction.
- **Ordering** The nodes can be ordered after the branch length; either **Increasing** (shown in figure 25.13) or **Decreasing**.
- **Reset Tree Topology** Resets to the default tree topology and node order (see figure 25.13). Any previously collapsed nodes will be uncollapsed.

- **Fixed width on zoom** Locks the horizontal size of the tree to the size of the main window. Zoom is therefore only performed on the vertical axis when this option is enabled.
- **Show as unrooted tree** The tree can be shown with or without a root.

25.5.3 Node settings

The nodes can be manipulated in several ways.

- **Leaf node symbol** Leaf nodes can be shown as a range of different symbols (Dot, Box, Circle, etc.).
- **Internal node symbols** The internal nodes can also be shown with a range of different symbols (Dot, Box, Circle, etc.).
- **Max. symbol size** The size of leaf- and internal node symbols can be adjusted.
- **Avoid overlapping symbols** The symbol size will be automatically limited to avoid overlaps between symbols in the current view.
- **Node color** Specify a fixed color for all nodes in the tree.

The node layout settings in the Side Panel are shown in figure 25.14.

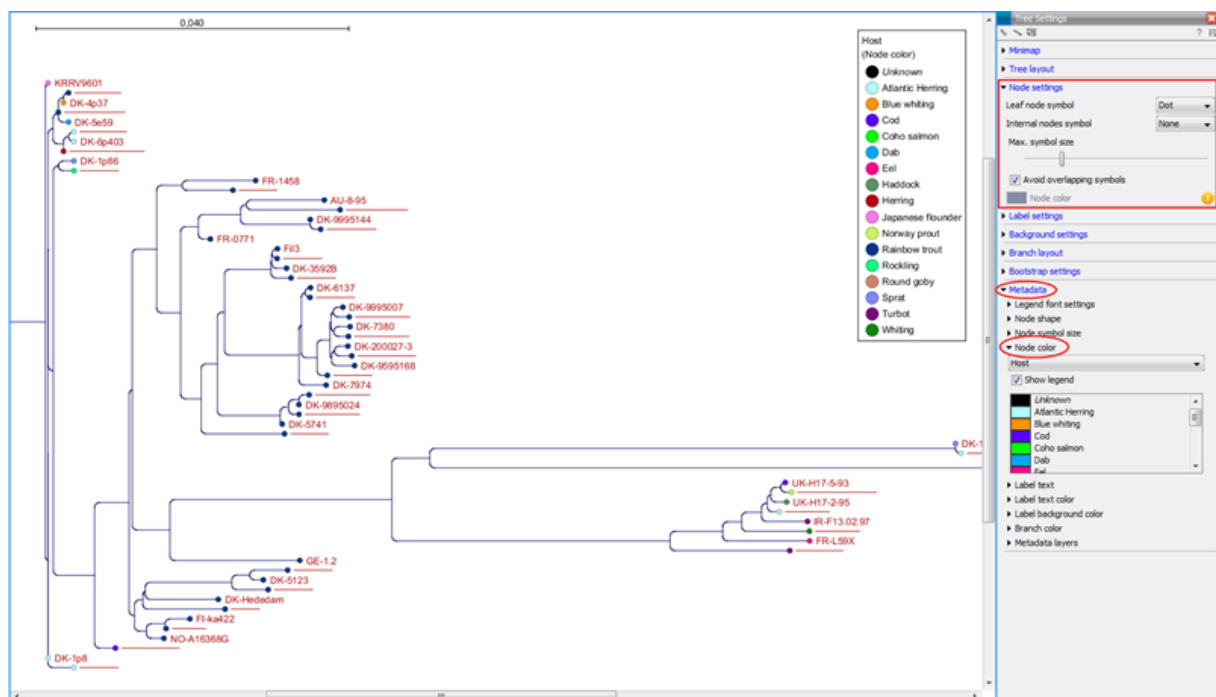


Figure 25.14: The Node Layout settings. Node color is specified by metadata and is therefore inactive in this example.

25.5.4 Label settings

- **Label font settings** Can be used to specify/adjust font type, size and typography (Bold, Italic or normal).

- **Hide overlapping labels** Disable automatic hiding of overlapping labels and display all labels even if they overlap.
- **Show internal node labels** Labels for internal nodes of the tree (if any) can be displayed. Please note that subtrees and nodes can be labeled with a custom text. This is done by right clicking the node and selecting **Edit Label** (see figure 25.15).
- **Show leaf node labels** Leaf node labels can be shown or hidden.
- **Rotate Subtree labels** Subtree labels can be shown horizontally or vertically. Labels are shown vertically when "Rotate subtree labels" has been selected. Subtree labels can be added with the right click option "Set Subtree Label" that is enabled from "Decorate subtree" (see section 25.5.9).
- **Align labels** Align labels to the node furthest from the center of the tree so that all labels are positioned next to each other. The exact behavior depends on the selected tree layout.
- **Connect labels to nodes** Adds a thin line from the leaf node to the aligned label. Only possible when Align labels option is selected.

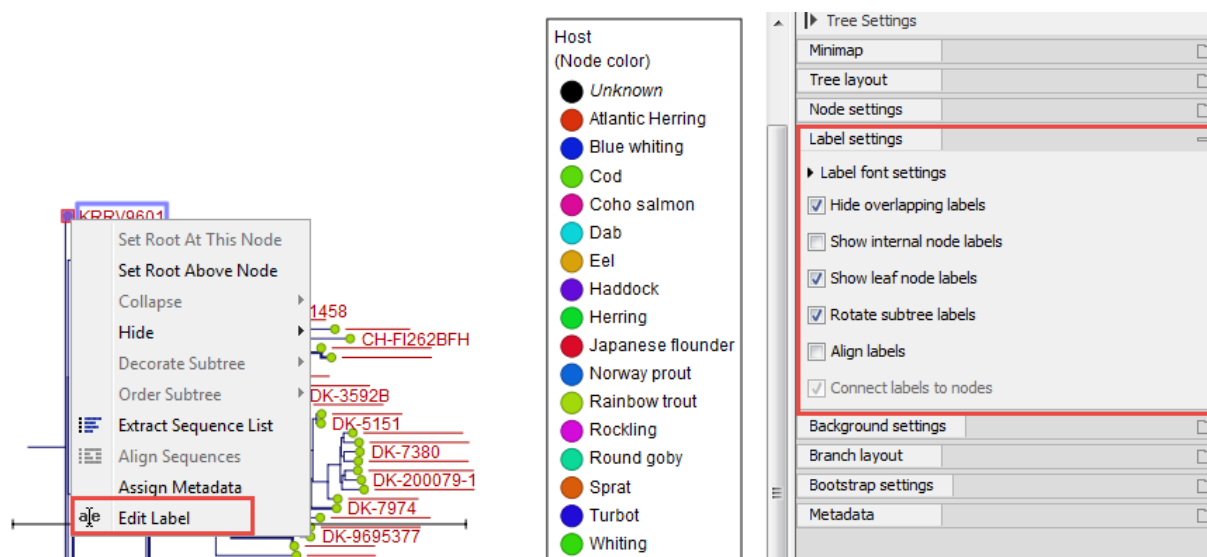


Figure 25.15: "Edit label" in the right click menu can be used to customize the label text. The way node labels are displayed can be controlled through the labels settings in the right side panel.

When working with big trees there is typically not enough space to show all labels. As illustrated in figure 25.15, only some of the labels are shown. The hidden labels are illustrated with thin horizontal lines (figure 25.16).

There are different ways of showing more labels. One way is to reduce the font size of the labels, which can be done under **Label font settings** in the Side Panel. Another option is to zoom in on specific areas of the tree (figure 25.16 and figure 25.17). The last option is to disable **Hide overlapping labels** under "Label settings" in the right side panel. When this option is unchecked all labels are shown even if the text overlaps. When allowing overlapping labels it is usually a good idea to disable **Show label background** under "Background settings" (see section 25.5.5).

Note! When working with a tree with hidden labels, it is possible to make the hidden label text appear by moving the mouse over the node with the hidden label.

Note! The text within labels can be edited by editing the metadata table values directly.

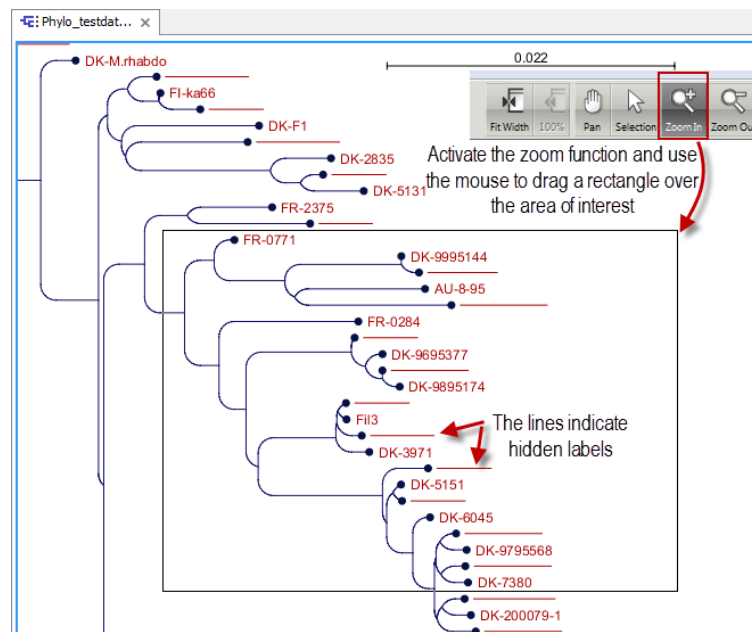


Figure 25.16: The zoom function in the upper right corner of the Workbench can be used to zoom in on a particular region of the tree. When the zoom function has been activated, use the mouse to drag a rectangle over the area that you wish to zoom in at.

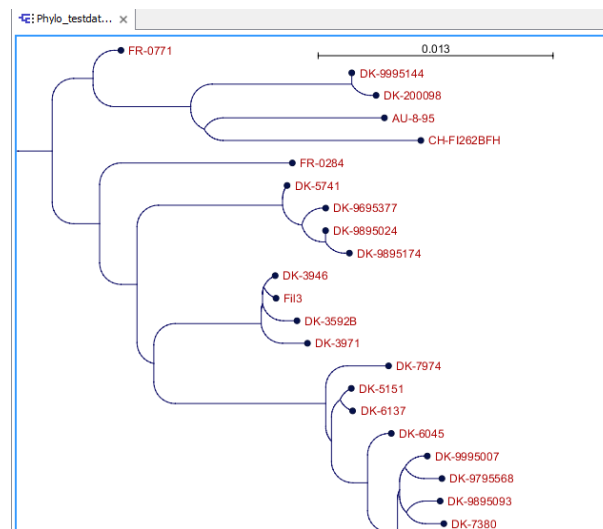


Figure 25.17: After zooming in on a region of interest more labels become visible. In this example all labels are now visible.

25.5.5 Background settings

- **Show label background** Show a background color for each label. Once ticked, it is possible to specify whether to use a fixed color or to use the color that is associated with the selected metadata category.

25.5.6 Branch layout

- **Branch length font settings** Specify/adjust font type, size and typography (Bold, Italic or normal).
- **Line color** Select the default line color.
- **Line width** Select the width of branches (1.0-3.0 pixels).
- **Curvature** Adjust the degree of branch curvature to get branches with round corners.
- **Min. length** Select a minimum branch length. This option can be used to prevent nodes connected with a short branch to cluster at the parent node.
- **Show branch lengths** Show or hide the branch lengths.

The branch layout settings in the Side Panel are shown in figure 25.18.

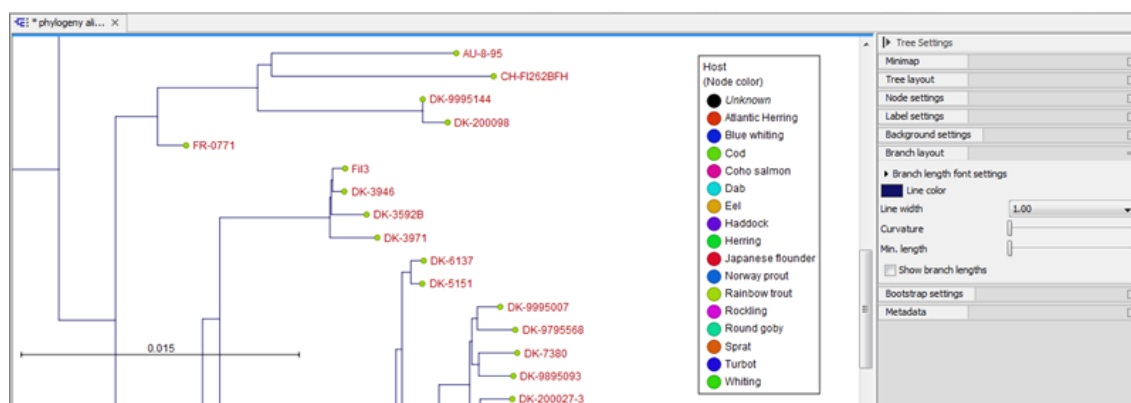


Figure 25.18: Branch Layout settings.

25.5.7 Bootstrap settings

Bootstrap values can be shown on the internal nodes. The bootstrap values are shown in percent and can be interpreted as confidence levels where a bootstrap value close to 100 indicate a clade, which is strongly supported by the data from which the tree was reconstructed. Bootstrap values are useful for identifying clades in the tree where the topology (and branch lengths) should not be trusted.

Some branches in rooted trees may not have bootstrap values. Trees constructed with neighbour joining are unrooted and to correctly visualize them, the "Radial" view is required. In all other tree views we need a root to visualize the tree. An "artificial node" and therefore an extra branch are created for such visualization to achieve this, which makes it look like a bootstrap value is missing

- **Bootstrap value font settings** Specify/adjust font type, size and typography (Bold, Italic or normal).
- **Show bootstrap values (%)** Show or hide bootstrap values. When selected, the bootstrap values (in percent) will be displayed on internal nodes if these have been computed during the reconstruction of the tree.

- **Bootstrap threshold (%)** When specifying a bootstrap threshold, the branch lengths can be controlled manually by collapsing internal nodes with bootstrap values under a certain threshold.
- **Highlight bootstrap \geq (%)** Highlights branches where the bootstrap value is above the user defined threshold.

25.5.8 Visualizing metadata

Metadata associated with a phylogenetic tree (described in detail in section 25.6) can be visualized in a number of different ways:

- **Node shape** Different node shapes are available to visualize metadata.
- **Node symbol size** Change the node symbol size to visualize metadata.
- **Node color** Change the node color to visualize metadata.
- **Label text** The metadata can be shown directly as text labels as shown in figure 25.19.
- **Label text color** The label text can be colored and used to visualize metadata (see figure 25.19).
- **Label background color** The background color of node text labels can be used to visualize metadata.
- **Branch color** Branch colors can be changed according to metadata.
- **Metadata layers** Color coded layers shown next to leaf nodes.

Please note that when visualizing metadata through a tree property that can be adjusted in the right side panel (such as node color or node size), an exclamation mark will appear next to the control for that property to indicate that the setting is inactive because it is defined by metadata (see figure 25.14).

25.5.9 Node right click menu

Additional options for layout and extraction of subtree data are available when right clicking the nodes (figure 25.15):

- **Set Root At This Node** Re-root the tree using the selected node as root. Please note that re-rooting will change the tree topology. This option is only available for internal nodes, not leaf nodes.
- **Set Root Above Node** Re-root the tree by inserting a node between the selected node and its parent. Useful for rooting trees using an outgroup.
- **Collapse** Branches associated with a selected node can be collapsed with or without the associated labels. Collapsed branches can be uncollapsed using the *Uncollapse* option in the same menu.

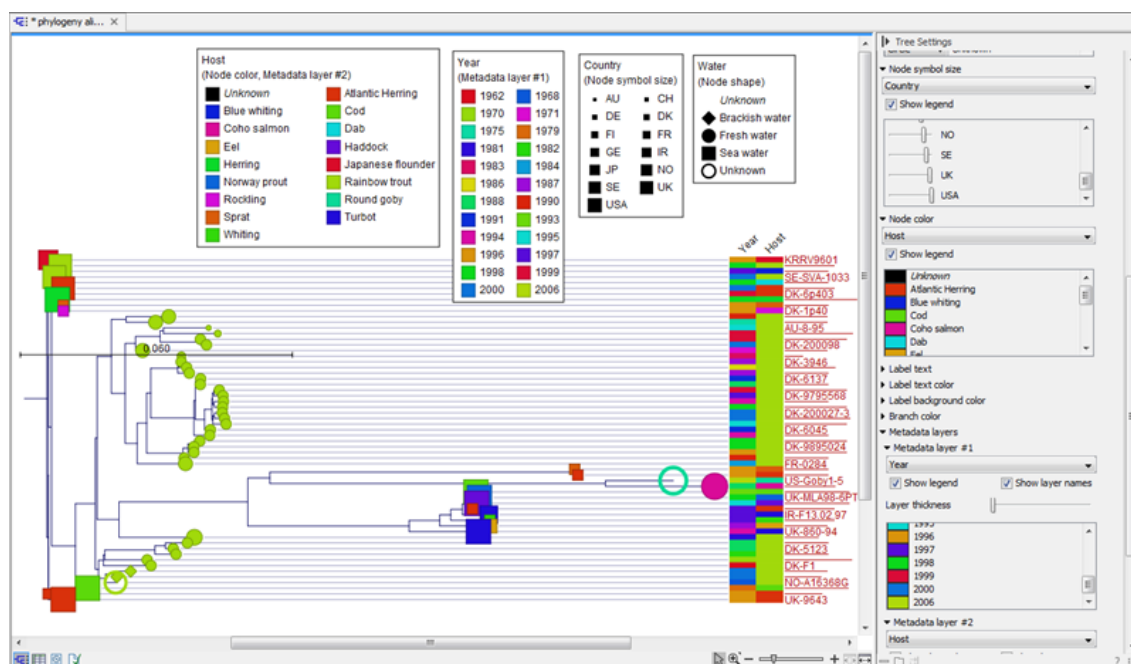


Figure 25.19: Different types of metadata can be visualized by adjusting node size, shape, and color. Two color-code metadata layers (Year and Host) are shown in the right side of the tree.

- **Hide** Can be used to hide a node or a subtree. Hidden nodes or subtrees can be shown again using the *Show Hidden Subtree* function on a node which is root in a subtree containing hidden nodes (see figure 25.20). When hiding nodes, a new button appears labeled "Show X hidden nodes" in the Side Panel under "Tree Layout" (figure 25.21). When pressing this button, all hidden nodes are shown again.
- **Decorate Subtree** A subtree can be labeled with a customized name, and the subtree lines and/or background can be colored. To save the decoration, see figure 25.11 and use option: **Save/Restore Settings | Save Tree View Settings On This Tree View only**.
- **Order Subtree** Rearrange leaves and branches in a subtree by Increasing/Decreasing depth, respectively. Alternatively, change the order of a node's children by left clicking and dragging one of the node's children.
- **Extract Sequence List** Sequences associated with selected leaf nodes are extracted to a new sequence list.
- **Align Sequences** Sequences associated with selected leaf nodes are extracted and used as input to the *Create Alignment* tool.
- **Assign Metadata** Metadata can be added, deleted or modified. To add new metadata categories a new "Name" must be assigned. (This will be the column header in the metadata table). To add a new metadata category, enter a value in the "Value" field. To delete values, highlight the relevant nodes and right click on the selected nodes. In the dialog that appears, use the drop-down list to select the name of the desired metadata category and leave the value field empty. When pressing "Add" the values for the selected metadata category will be deleted from the selected nodes. Metadata can be modified in the same way, but instead of leaving the value field empty, the new value should be entered.

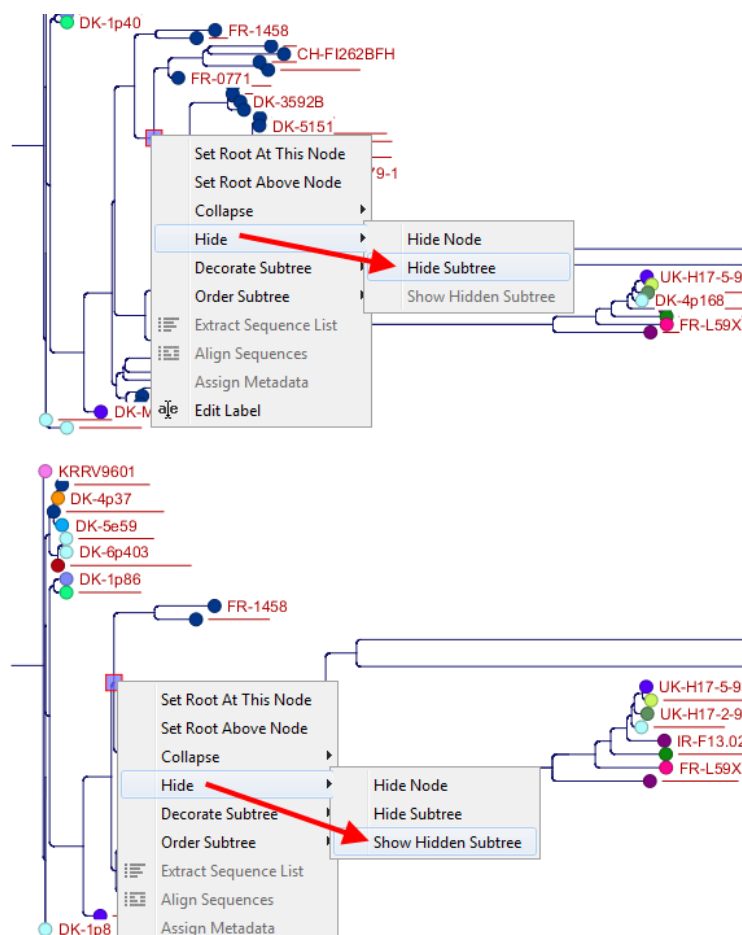


Figure 25.20: A subtree can be hidden by selecting "Hide Subtree" and is shown again when selecting "Show Hidden Subtree" on a parent node.

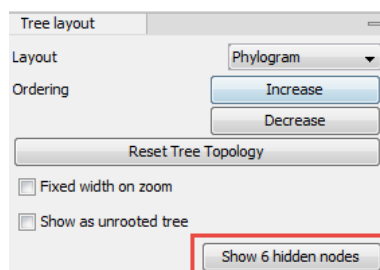




Figure 25.21: When hiding nodes, a new button labeled "Show X hidden nodes" appears in the Side Panel under "Tree Layout". When pressing this button, all hidden nodes are brought back.

- **Edit label** Edit the text in the selected node label. Labels can be shown or hidden by using the Side Panel: **Label settings | Show internal node labels**

25.6 Metadata and phylogenetic trees

When a tree is reconstructed, some mandatory metadata will be added to nodes in the tree. These metadata are special in the sense that the tree viewer has specialized features for visualizing the data and some of them cannot be edited. The mandatory metadata include:

- **Node name** The node name.
- **Branch length** The length of the branch, which connects a node to the parent node.
- **Bootstrap value** The bootstrap value for internal nodes.
- **Size** The length of the sequence which corresponds to each leaf node. This only applies to leaf nodes.
- **Start of sequence** The first 50bp of the sequence corresponding to each leaf node.

To view metadata associated with a phylogenetic tree, click on the table icon () at the bottom of the tree. If you hold down the Ctrl key (or ⌘ on Mac) while clicking on the table icon (), you will be able to see both the tree and the table in a split view (figure 25.22).

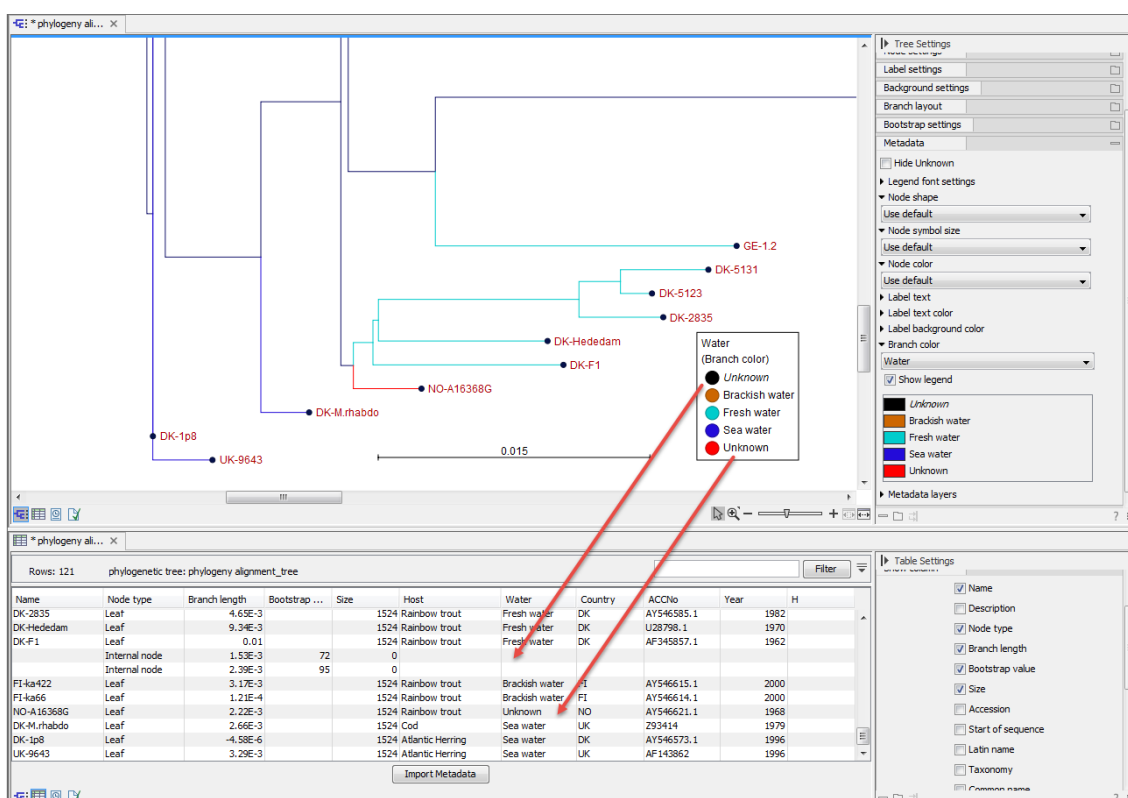


Figure 25.22: Tabular metadata that is associated with an existing tree shown in a split view. Note that *Unknown* written in italics (black branches) refer to missing metadata, while *Unknown* in regular font refers to metadata labeled as "Unknown".

Additional metadata can be associated with a tree by clicking the **Import Metadata** button. This will open up the dialog shown in figure 25.23.

To associate metadata with an existing tree a common denominator is required. This is achieved by mapping the node names in the "Name" column of the metadata table to the names that have been used in the metadata table to be imported. In this example the "Strain" column holds the names of the nodes and this column must be assigned "Name" to allow the importer to associate metadata with nodes in the tree.

It is possible to import a subset of the columns in a set of metadata. An example is given in figure 25.23. The column "H" is not relevant to import and can be excluded simply by leaving the text field at the top row of the column empty.

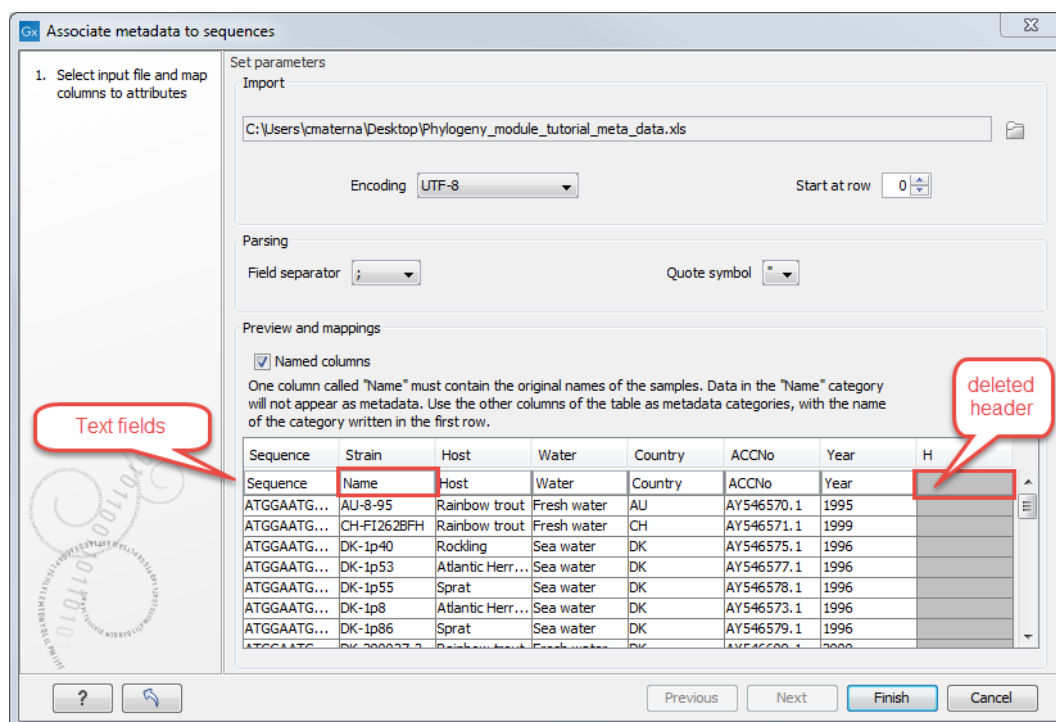


Figure 25.23: Import of metadata for a tree. The second column named "Strain" is chosen as the common denominator by entering "Name" in the text field of the column. The column labeled "H" is ignored by not assigning a column heading to this column.

25.6.1 Table Settings and Filtering

How to use the metadata table (see figure 25.24):

- **Column width** The column width can be adjusted in two ways; *Manually* or *Automatically*.
- **Show column** Selects which metadata categories that are shown in the table layout.
- **Filtering Metadata information** Metadata information in a table can be filtered by a simple- or advanced mode (this is described in section 9.2).

25.6.2 Add or modify metadata on a tree

It is possible to add and modify metadata from both the tree view and the table view.

Metadata can be added and edited in the metadata table by using the following right click options (see figure 25.25):

- **Assign Metadata** The right click option "Assign Metadata" can be used for four purposes.

Rows: 121 phylogenetic tree: phylogeny alignment_tree

Name	Node type	Branch length	Bootstrap v...	Size	Host	Water	Country	ACCNo	Year	H
DK-1p53	Leaf	7.07E-4			1524 Atlantic Herring	Sea water	DK	AY546577.1	1996	
DK-4p168	Leaf	1.53E-3			1524 Atlantic Herring	Sea water	DK	AY546582.1	1997	
DK-1p8	Leaf	4.58E-6			1524 Atlantic Herring	Sea water	DK	AY546573.1	1996	
UK-9643	Leaf	3.29E-3			1524 Atlantic Herring	Sea water	UK	AF143862	1996	
DK-4p37	Leaf	1.11E-5			1524 Blue whiting	Sea water	DK	FJ460590.1	1997	
UK-H17-5-93	Leaf	5.96E-4			1524 Cod	Sea water	UK	AY546630.1	1993	
DK-M.rhabdo	Leaf	2.66E-3			1524 Cod	Sea water	UK	Z93414	1979	
US-Makah	Leaf	0.02			1524 Coho salmon	Fresh water	USA	U28747.1	1988	
DK-9e59	Leaf	1.31E-3			1524 Dab	Sea water	DK	AY546583.1	1998	
FR-L59X	Leaf	0.01			1524 Eel	Sea water	FR	AY546618.1	1987	
DK-3971	Leaf	2.56E-3			1524 Rainbow trout	Fresh water	DK	AY546587.1	1987	
DK-6137	Leaf	6.93E-4			1524 Rainbow trout	Fresh water	DK	AY546593.1	1991	
DK-5151	Leaf	6.20E-4			1524 Rainbow trout	Fresh water	DK	AF345859.1	1988	

Import Metadata

Table Settings
Column width: Manual
Show column:
☒ Name
☐ Description
☒ Node type
☒ Branch length
☒ Bootstrap value
☒ Size
☒ Year
☒ H
Select All
Deselect All
0 rows selected

Figure 25.24: Metadata table. The column width can be adjusted manually or automatically. Under "Show column" it is possible to select which columns should be shown in the table. Filtering using specific criteria can be performed.

Water	Country	ACCNo	Year	H
Unknown	USA	AR672615.1	2006	
Unknown			1968	
Sea water			1996	
Sea water			1998	
Sea water			1997	
Sea water			2000	
Sea water			1998	
Sea water			2000	
Sea water			1999	
Sea water	UK	AY546631.1	1998	

Table filters
Extract Sequence List
Align Sequences
Assign Metadata
Delete Metadata Water
Edit Water

Figure 25.25: Right click options in the metadata table.

- To add new metadata categories (columns). In this case, a new "Name" must be assigned, which will be the column header. To add a new column requires that a value is entered in the "Value" field. This can be done by right clicking anywhere in the table.
 - To add values to one or more rows in an existing column. In this case, highlight the relevant rows and right click on the selected rows. In the dialog that appears, use the drop-down list to select the name of the desired column and enter a value.
 - To delete values from an existing column. This is done in the same way as when adding a new value, with the only exception that the value field should be left empty.
 - To delete metadata columns. This is done by selecting all rows in the table followed by a right click anywhere in the table. Select the name of the column to delete from the drop down menu and leave the value field blank. When pressing "Add", the selected column will disappear.
- **Delete Metadata "column header"** This is the most simple way of deleting a metadata column. Click on one of the rows in the column to delete and select "Delete column header".
 - **Edit "column header"** To modify existing metadata point, right click on a cell in the table and select the "Edit column header". To edit multiple entries at once, select multiple rows in the table, right click a selected cell in the column you want to edit and choose "Edit column header" (see an example in figure 25.26). This will change values in all selected rows in the column that was clicked.

25.6.3 Undefined metadata values on a tree

When visualizing a metadata category where one or more nodes in the tree have undefined values (empty fields in the table), these nodes will be visualized using a default value in **italics** in the

Size	Host	Water	Country	ACCNo
1524	Round goby	Unknown	USA	AB672615.1
1524	Rainbow trout	Fresh water	CH	AY546571.1
1524	Eel	Sea water	FR	AY546618.1
0	0			
1524	Rainbow trout			U28800
1524	Rainbow trout			AF345857.1
1524	Rainbow trout			AY546570.1
0	0			
1524	Rainbow trout			AF143863
1524	Rainbow trout			U28798.1
0	0			
1524	Whiting	Sea water	DK	AY546581.1

Figure 25.26: To modify existing metadata, click on the specific field, select "Edit <column header>" and provide a new value.

top of the legend (see the entry "Unknown" in figure 25.27). To remove this entry in the legend, all nodes must have a value assigned in the corresponding metadata category.

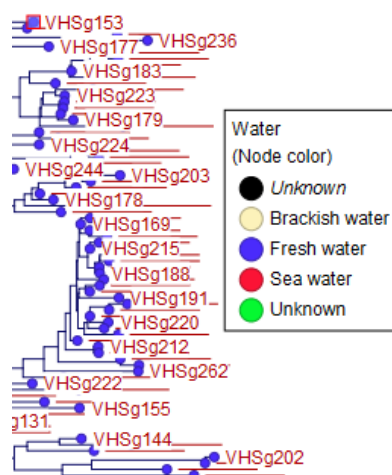


Figure 25.27: A legend for a metadata category where one or more values are undefined. Fill your metadata table with a value of your choice to edit the mention of "Unknown" in the legend. Note that the "Unknown" that is not in italics is used for data that had a value written as "Unknown" in the metadata table.

25.6.4 Selection of specific nodes

Selection of nodes in a tree is automatically synchronized to the metadata table and the other way around. Nodes in a tree can be selected in three ways:

- *Selection of a single node* Click once on a single node. Additional nodes can be added by holding down Ctrl (or ⌘ for Mac) and clicking on them (see figure 25.28).
- *Selecting all nodes in a subtree* Double clicking on a inner node results in the selection of all nodes in the subtree rooted at the node.
- *Selection via the Metadata table* Select one or more entries in the table. The corresponding nodes will now be selected in the tree.

It is possible to extract a subset of the underlying sequence data directly through either the tree viewer or the metadata table as follows. Select one or more nodes in the tree where at least

one node has a sequence attached. Right click one of the selected nodes and choose **Extract Sequence List**. This will generate a new sequence list containing all sequences attached to the selected nodes. The same functionality is available in the metadata table where sequences can be extracted from selected rows using the right click menu. Please note that all extracted sequences are copies and any changes to these sequences will not be reflected in the tree.

When analyzing a phylogenetic tree it is often convenient to have a multiple alignment of sequences from e.g. a specific clade in the tree. A quick way to generate such an alignment is to first select one or more nodes in the tree (or the corresponding entries in the metadata table) and then select **Align Sequences** in the right click menu. This will extract the sequences corresponding to the selected elements and use a copy of them as input to the multiple alignment wizard that pops up and click **Finish**. The multiple alignment will now be generated.

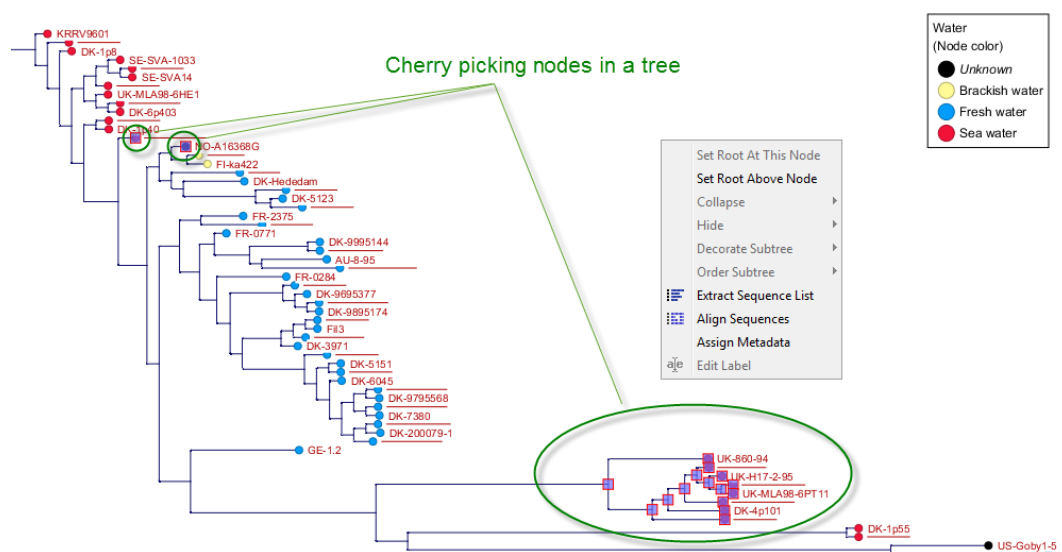


Figure 25.28: Cherry picking nodes in a tree. The selected leaf sequences can be extracted by right clicking on one of the selected nodes and selecting "Extract Sequence List". It is also possible to Align Sequences directly by right clicking on the nodes or leaves.

Chapter 26

RNA structure

Contents

26.1 RNA secondary structure prediction	676
26.1.1 Selecting sequences for prediction	676
26.1.2 Secondary structure prediction parameters	677
26.1.3 Structure as annotation	681
26.2 View and edit secondary structures	682
26.2.1 Graphical view and editing of secondary structure	682
26.2.2 Tabular view of structures and energy contributions	685
26.2.3 Symbolic representation in sequence view	688
26.2.4 Probability-based coloring	689
26.3 Evaluate structure hypothesis	690
26.3.1 Selecting sequences for evaluation	690
26.3.2 Probabilities	691
26.4 Structure scanning plot	692
26.4.1 Selecting sequences for scanning	692
26.4.2 The structure scanning result	693
26.5 Bioinformatics explained: RNA structure prediction by minimum free energy minimization	694
26.5.1 The algorithm	695
26.5.2 Structure elements and their energy contribution	696

Ribonucleic acid (RNA) is a nucleic acid polymer that plays several important roles in the cell.

As for proteins, the three dimensional shape of an RNA molecule is important for its molecular function. A number of tertiary RNA structures are known from crystallography but de novo prediction of tertiary structures is not possible with current methods. However, as for proteins RNA tertiary structures can be characterized by secondary structural elements which are hydrogen bonds within the molecule that form several recognizable "domains" of secondary structure like stems, hairpin loops, bulges and internal loops. A large part of the functional information is thus contained in the secondary structure of the RNA molecule, as shown by the high degree of base-pair conservation observed in the evolution of RNA molecules.

Computational prediction of RNA secondary structure is a well defined problem and a large body of work has been done to refine prediction algorithms and to experimentally estimate the relevant biological parameters.

In *CLC Genomics Workbench* we offer the user a number of tools for analyzing and displaying RNA structures. These include:

- Secondary structure prediction using state-of-the-art algorithms and parameters
- Calculation of full partition function to assign probabilities to structural elements and hypotheses
- Scanning of large sequences to find local structure signal
- Inclusion of experimental constraints to the folding process
- Advanced viewing and editing of secondary structures and structure information

26.1 RNA secondary structure prediction

CLC Genomics Workbench uses a minimum free energy (MFE) approach to predict RNA secondary structure. Here, the stability of a given secondary structure is defined by the amount of free energy used (or released) by its formation. The more negative free energy a structure has, the more likely is its formation since more stored energy is released by the event.

Free energy contributions are considered additive, so the total free energy of a secondary structure can be calculated by adding the free energies of the individual structural elements. Hence, the task of the prediction algorithm is to find the secondary structure with the minimum free energy. As input to the algorithm empirical energy parameters are used. These parameters summarize the free energy contribution associated with a large number of structural elements. A detailed structure overview can be found in section 26.5.

In *CLC Genomics Workbench*, structures are predicted by a modified version of Professor Michael Zukers well known algorithm [Zuker, 1989b] which is the algorithm behind a number of RNA-folding packages including MFOLD. Our algorithm is a dynamic programming algorithm for free energy minimization which includes free energy increments for coaxial stacking of stems when they are either adjacent or separated by a single mismatch. The thermodynamic energy parameters used are from Mfold version 3. Read about Mfold here <http://www.unafold.org/>.

26.1.1 Selecting sequences for prediction

To start the **Predict Secondary Structure** tool, go to:

Tools | Classical Sequence Analysis (📁) | RNA Structure (🔍) | Predict Secondary Structure (🔍)

This opens the dialog shown in figure 26.1.

If you have selected sequences before running the tool, those sequences will be listed in the **Selected Elements** pane of the dialog. Use the arrows to add or remove sequences or sequence lists from the selected elements. You can use both DNA and RNA sequences - DNA will be folded as if it were RNA.

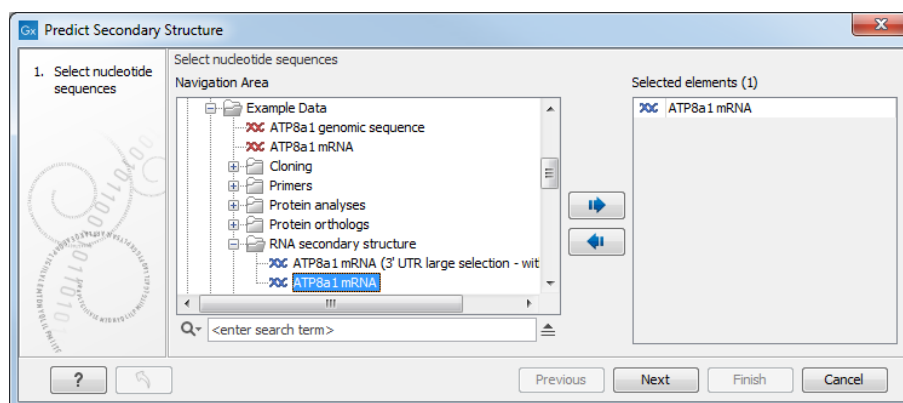


Figure 26.1: Selecting RNA or DNA sequences for structure prediction (DNA is folded as if it was RNA).

26.1.2 Secondary structure prediction parameters

Click **Next** to adjust secondary structure prediction parameters (figure 26.2).

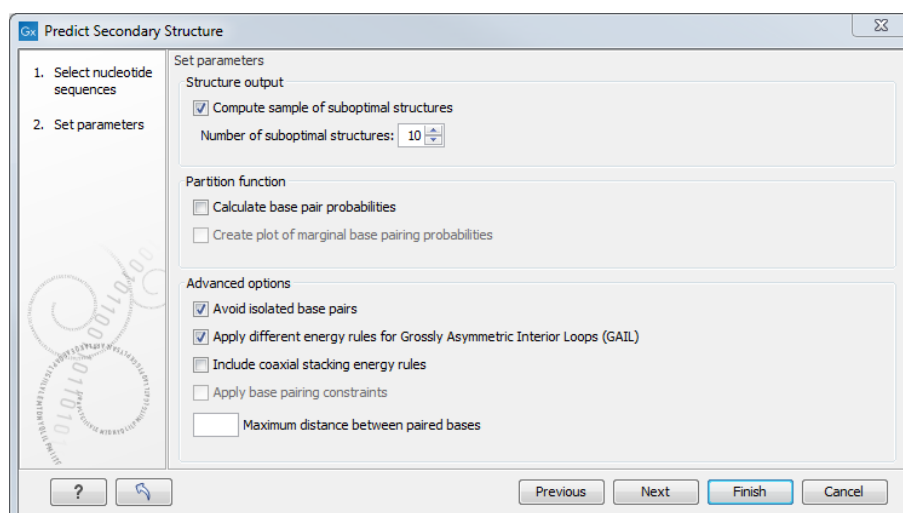


Figure 26.2: Adjusting parameters for secondary structure prediction.

Structure output

The predict secondary structure algorithm always calculates the minimum free energy structure of the input sequence. In addition to this, it is also possible to compute a sample of suboptimal structures by ticking the checkbox **Compute sample of suboptimal structures**.

Subsequently, you can specify how many structures to include in the output. The algorithm then iterates over all permissible canonical base pairs and computes the minimum free energy and associated secondary structure constrained to contain a specified base pair. These structures are then sorted by their minimum free energy and the most optimal are reported given the specified number of structures. Note that two different sub-optimal structures can have the same minimum free energy. Further information about suboptimal folding can be found in [Zuker, 1989a].

Partition function

The predicted minimum free energy structure gives a point-estimate of the structural conformation of an RNA molecule. However, this procedure implicitly assumes that the secondary structure is at equilibrium, that there is only a single accessible structure conformation, and that the parameters and model of the energy calculation are free of errors.

Obvious deviations from these assumptions make it clear that the predicted MFE structure may deviate somewhat from the actual structure assumed by the molecule. This means that rather than looking at the MFE structure it may be informative to inspect statistical properties of the structural landscape to look for general structural properties which seem to be robust to minor variations in the total free energy of the structure (see [Mathews et al., 2004]).

To this end *CLC Genomics Workbench* allows the user to calculate the complete secondary structure partition function using the algorithm described in [Mathews et al., 2004] which is an extension of the seminal work by [McCaskill, 1990].

There are two options regarding the partition function calculation:

- **Calculate base pair probabilities.** This option invokes the partition function calculation and calculates the marginal probabilities of all possible base pairs and the marginal probability that any single base is unpaired.
- **Create plot of marginal base pairing probabilities.** This creates a plot of the marginal base pair probability of all possible base pairs as shown in figure 26.3.

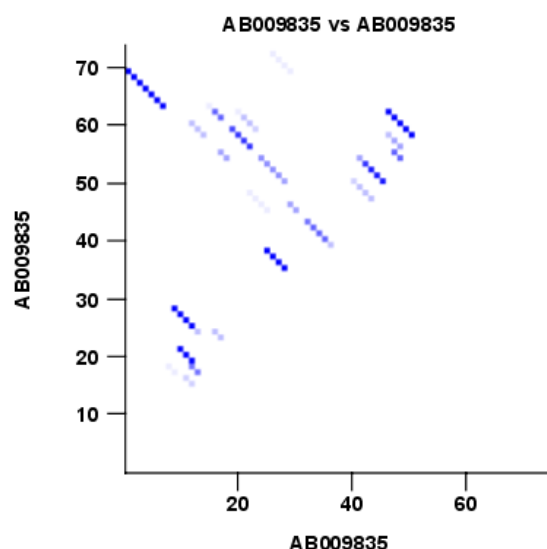


Figure 26.3: The marginal base pair probability of all possible base pairs.

The marginal probabilities of base pairs and of bases being unpaired are distinguished by colors which can be displayed in the normal sequence view using the **Side Panel** - see section 26.2.3 and also in the secondary structure view. An example is shown in figure 26.4. Furthermore, the marginal probabilities are accessible from tooltips when hovering over the relevant parts of the structure.

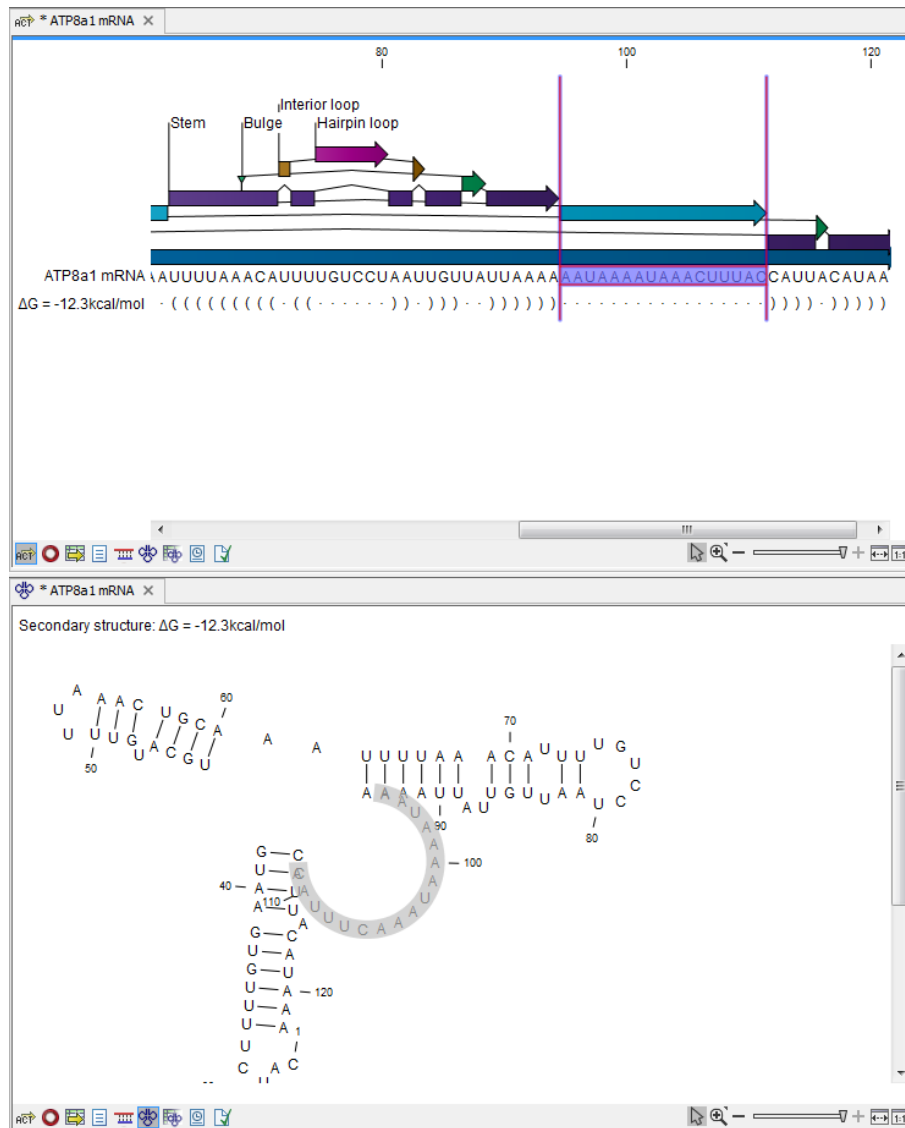


Figure 26.4: Marginal probability of base pairs shown in linear view (top) and marginal probability of being unpaired shown in the secondary structure 2D view (bottom).

Advanced options

The free energy minimization algorithm includes a number of advanced options:

- **Avoid isolated base pairs.** The algorithm filters out isolated base pairs (i.e. stems of length 1).
- **Apply different energy rules for Grossly Asymmetric Interior Loops (GAIL).** Compute the minimum free energy applying different rules for Grossly Asymmetry Interior Loops (GAIL). A Grossly Asymmetry Interior Loop (GAIL) is an interior loop that is $1 \times n$ or $n \times 1$ where $n > 2$ (see <http://www.unafold.org/doc/mfold-manual/node5.php>).
- **Include coaxial stacking energy rules.** Include free energy increments of coaxial stacking for adjacent helices [Mathews et al., 2004].
- **Apply base pairing constraints.** With base pairing constraints, you can easily add

experimental constraints to your folding algorithm. When you are computing suboptimal structures, it is not possible to apply base pair constraints. The possible base pairing constraints are:

- Force two equal length intervals to form a stem.
- Prohibit two equal length intervals to form a stem.
- Prohibit all nucleotides in a selected region to be a part of a base pair.

Base pairing constraints have to be added to the sequence before you can use this option - see below.

- **Maximum distance between paired bases.** Forces the algorithms to only consider RNA structures of a given upper length by setting a maximum distance between the base pair that opens a structure.

Specifying structure constraints

Structure constraints can serve two purposes in *CLC Genomics Workbench*: they can act as experimental constraints imposed on the MFE structure prediction algorithm or they can form a structure hypothesis to be evaluated using the partition function (see section 26.1.2).

To *force* two regions to form a stem, open a normal sequence view and:

Select the two regions you want to force by pressing Ctrl while selecting - (use ⌘ on Mac) | right-click the selection | Add Structure Prediction Constraints | Force Stem Here

This will add an annotation labeled "Forced Stem" to the sequence (see figure 26.5).

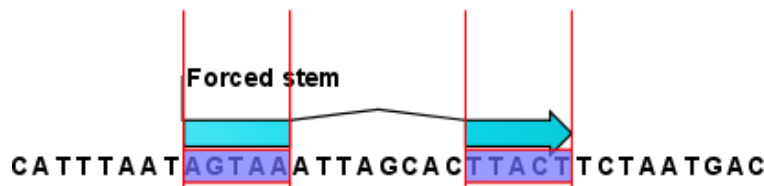


Figure 26.5: Force a stem of the selected bases.

Using this procedure to add base pairing constraints will force the algorithm to compute minimum free energy and structure with a stem in the selected region. The two regions must be of equal length.

To *prohibit* two regions to form a stem, open the sequence and:

Select the two regions you want to prohibit by pressing Ctrl while selecting - (use ⌘ on Mac) | right-click the selection | Add Structure Prediction Constraints | Prohibit Stem Here

This will add an annotation labeled "Prohibited Stem" to the sequence (see figure 26.6).

Using this procedure to add base pairing constraints will force the algorithm to compute minimum free energy and structure without a stem in the selected region. Again, the two selected regions must be of equal length.

To prohibit a region to be part of *any* base pair, open the sequence and:

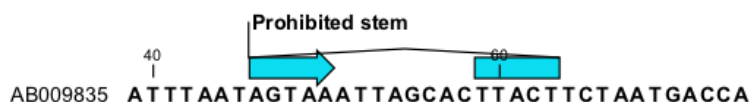


Figure 26.6: Prohibit the selected bases from forming a stem.

Select the bases you don't want to base pair | right-click the selection | Add Structure Prediction Constraints | Prohibit From Forming Base Pairs

This will add an annotation labeled "No base pairs" to the sequence, see 26.7.



Figure 26.7: Prohibiting any of the selected base from pairing with other bases.

Using this procedure to add base pairing constraints will force the algorithm to compute minimum free energy and structure without a base pair containing any residues in the selected region.

When you click **Predict secondary structure** (🔗) and click **Next**, check **Apply base pairing constraints** in order to force or prohibit stem regions or prohibit regions from forming base pairs.

You can add multiple base pairing constraints, e.g. simultaneously adding forced stem regions and prohibited stem regions and prohibit regions from forming base pairs.

26.1.3 Structure as annotation

You can choose to add the elements of the best structure as annotations (see figure 26.8).

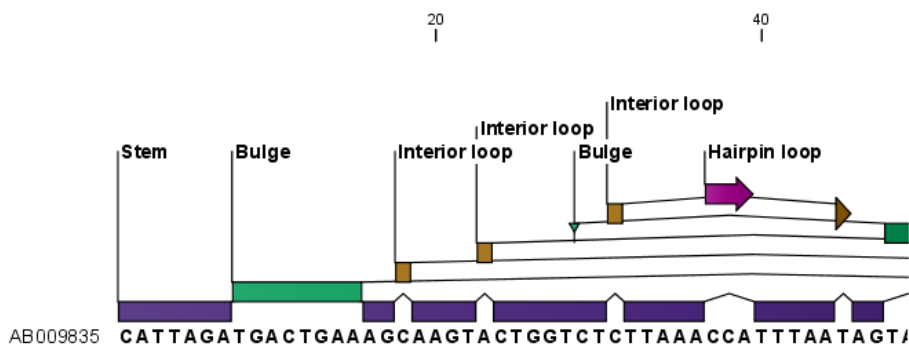


Figure 26.8: Annotations added for each structure element.

This makes it possible to use the structure information in other analysis in the *CLC Genomics Workbench*. You can e.g. align different sequences and compare their structure predictions.

Note that possibly existing structure annotation will be removed when a new structure is calculated and added as annotations.

If you generate multiple structures, only the best structure will be added as annotations. If you wish to add one of the sub-optimal structures as annotations, this can be done from the **Show Secondary Structure Table** (🔗) described in section 26.2.2.

26.2 View and edit secondary structures

When you predict RNA secondary structure (see section 26.1), the resulting predictions are attached to the sequence and can be shown as:

- Annotations in the ordinary sequence views (Linear sequence view (🔍), Annotation table (📄) etc. This is only possible if this has been chosen in the dialog in figure 26.2. See an example in figure 26.8.
- Symbolic representation below the sequence (see section 26.2.3).
- A graphical view of the secondary structure (see section 26.2.1).
- A tabular view of the energy contributions of the elements in the structure. If more than one structure have been predicted, the table is also used to switch between the structures shown in the graphical view. The table is described in section 26.2.2.

26.2.1 Graphical view and editing of secondary structure

To show the secondary view of an already open sequence, click the **Show Secondary Structure 2D View** (🔍) button at the bottom of the sequence view.

If the sequence is not open, click **Show** (📄) and select **Secondary Structure 2D View** (🔍).

This will open a view similar to the one shown in figure 26.9.

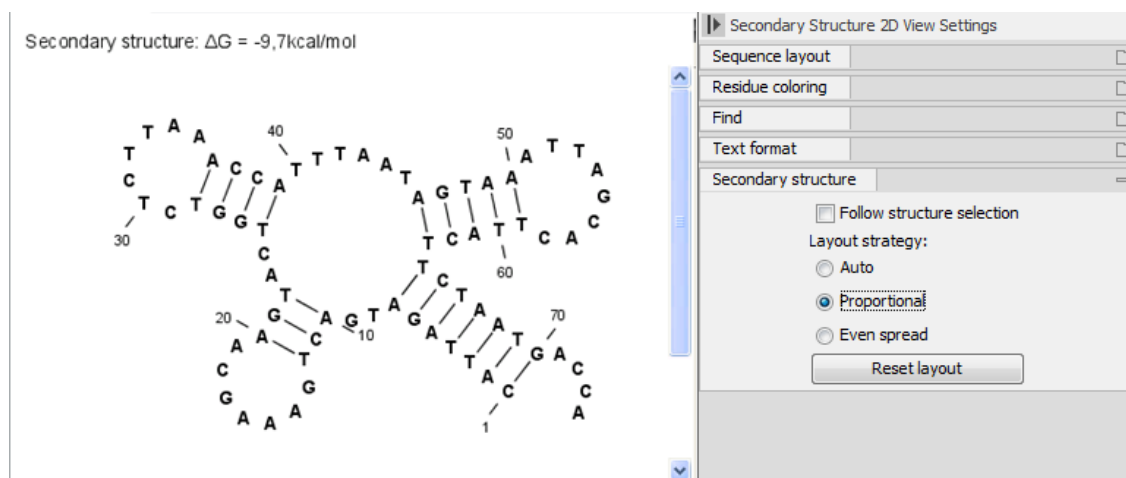


Figure 26.9: The secondary structure view of an RNA sequence zoomed in.

Like the normal sequence view, you can use **Zoom in** (🔍) and **Zoom out** (🔍). Zooming in will reveal the residues of the structure as shown in figure 26.9. For large structures, zooming out will give you an overview of the whole structure.

Side Panel settings

The settings in the **Side Panel** are a subset of the settings in the normal sequence view described in section 15.2.1. However, there are two additional groups of settings unique to the secondary structure 2D view: **Secondary structure**.

- **Follow structure selection.** This setting pertains to the connection between the structures in the secondary structure table (Figure 26.9). If this option is checked, the structure displayed in the secondary structure 2D view will follow the structure selections made in this table. See section 26.2.2 for more information.
- **Layout strategy.** Specify the strategy used for the layout of the structure. In addition to these strategies, you can also modify the layout manually as explained in the next section.
 - **Auto.** The layout is adjusted to minimize overlapping structure elements [Han et al., 1999]. This is the default setting (see figure 26.10).
 - **Proportional.** Arc lengths are proportional to the number of residues (see figure 26.11). Nothing is done to prevent overlap.
 - **Even spread.** Stems are spread evenly around loops as shown in figure 26.12.
- **Reset layout.** If you have manually modified the layout of the structure, clicking this button will reset the structure to the way it was laid out when it was created.

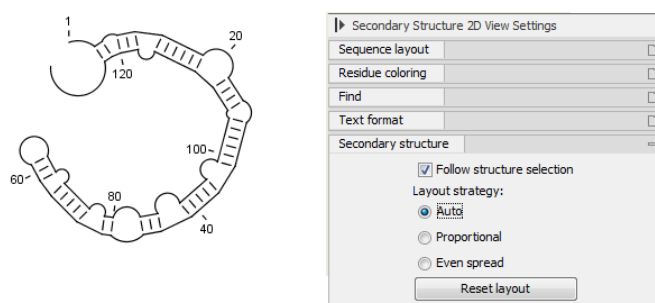


Figure 26.10: *Auto layout. Overlaps are minimized.*

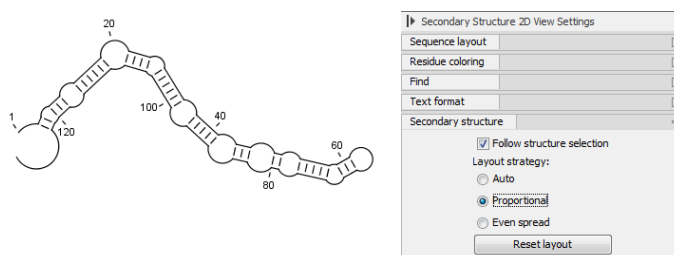


Figure 26.11: *Proportional layout. Length of the arc is proportional to the number of residues in the arc.*

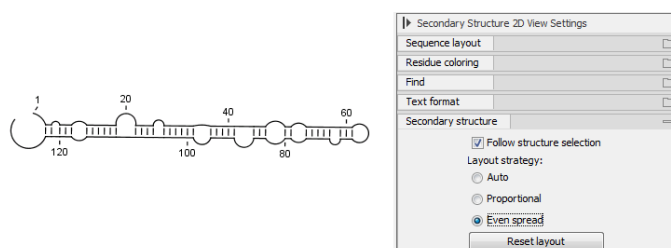


Figure 26.12: *Even spread. Stems are spread evenly around loops.*

Selecting and editing

When you are in **Selection mode** (Figure 26.13), you can select parts of the structure like in a normal sequence view:

Press down the mouse button where the selection should start | move the mouse cursor to where the selection should end | release the mouse button

One of the advantages of the secondary structure 2D view is that it is integrated with other views of the same sequence. This means that any selection made in this view will be reflected in other views (see figure 26.13).

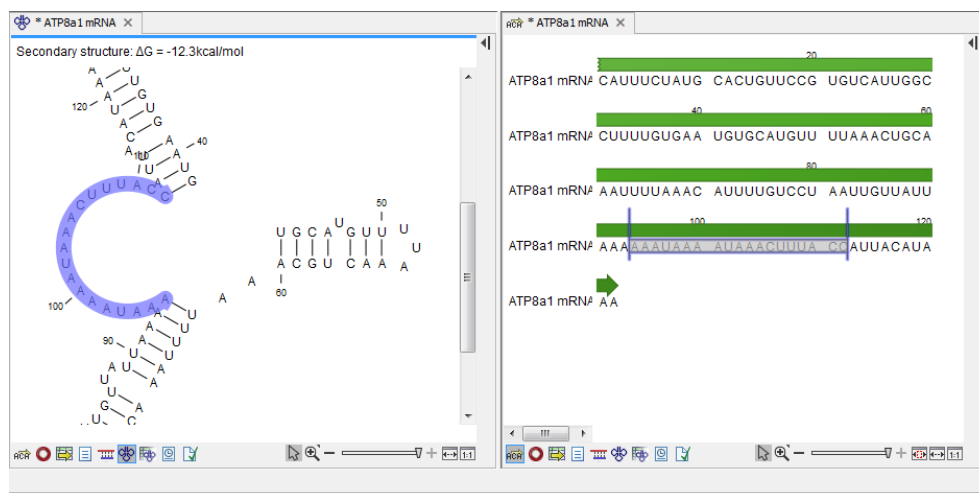



Figure 26.13: A *split view* of the secondary structure view and a linear sequence view.

If you make a selection in another sequence view, this will also be reflected in the secondary structure view.

The *CLC Genomics Workbench* seeks to produce a layout of the structure where none of the elements overlap. However, it may be desirable to manually edit the layout of a structure for ease of understanding or for the purpose of publication.

To edit a structure, first select the **Pan** () mode in the Tool bar (right-click on the zoom icon below the View Area). Now place the mouse cursor on the opening of a stem, and a visual indication of the anchor point for turning the substructure will be shown (see figure 26.14).

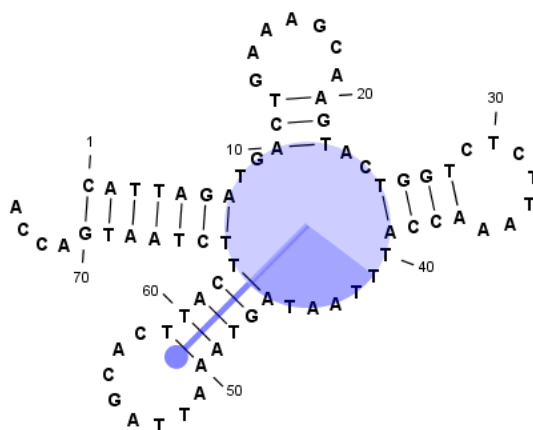


Figure 26.14: The blue circle represents the anchor point for rotating the substructure.

Click and drag to rotate the part of the structure represented by the line going from the anchor point. In order to keep the bases in a relatively sequential arrangement, there is a restriction

on how much the substructure can be rotated. The highlighted part of the circle represents the angle where rotating is allowed.

In figure 26.15, the structure shown in figure 26.14 has been modified by dragging with the mouse.

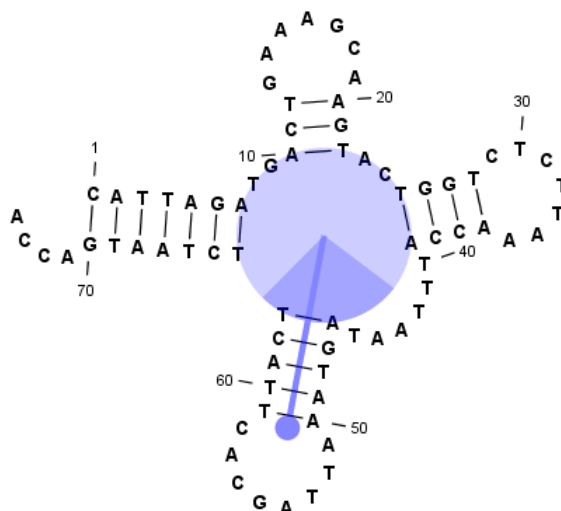


Figure 26.15: The structure has now been rotated.

Press **Reset layout** in the **Side Panel** to reset the layout to the way it looked when the structure was predicted.

26.2.2 Tabular view of structures and energy contributions

There are three main reasons to use the **Secondary structure table**:

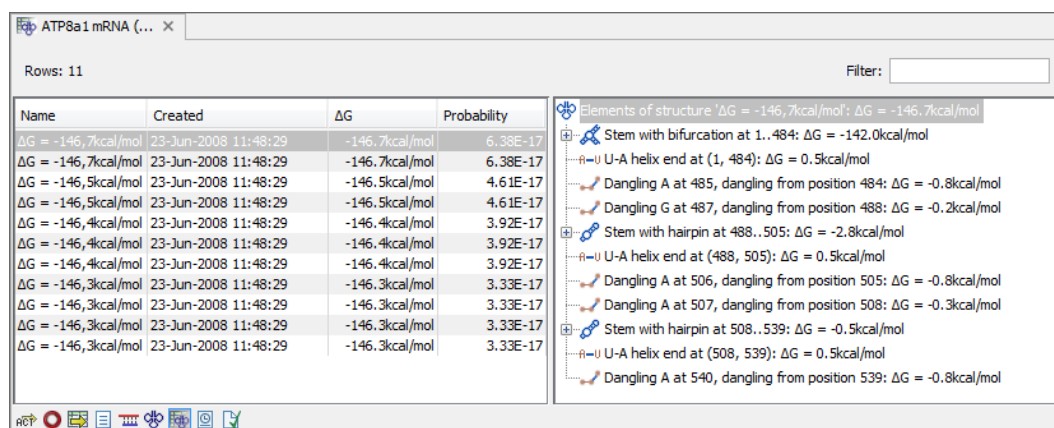
- If more than one structure is predicted (see section 26.1), the table provides an overview of all the structures which have been predicted.
- With multiple structures you can use the table to determine which structure should be displayed in the Secondary structure 2D view (see section 26.2.1).
- The table contains a hierarchical display of the elements in the structure with detailed information about each element's energy contribution.

To show the secondary structure table of an already open sequence, click the **Show Secondary Structure Table** (📄) button at the bottom of the sequence view.

If the sequence is not open, click **Show** (👉) and select **Secondary Structure Table** (📄).

This will open a view similar to the one shown in figure 26.16.

On the left side, all computed structures are listed with the information about structure name, when the structure was created, the free energy of the structure and the probability of the structure if the partition function was calculated. Selecting a row (equivalent: a structure) will display a tree of the contained substructures with their contributions to the total structure free energy. Each substructure contains a union of nested structure elements and other substructures (see a detailed description of the different structure elements in section 26.5.2). Each substructure



The screenshot shows a software window titled "ATP8a1 mRNA (... X)". It contains a table with 11 rows of secondary structure data and a list of substructures on the right.

Name	Created	ΔG	Probability
$\Delta G = -146.7 \text{ kcal/mol}$	23-Jun-2008 11:48:29	-146.7 kcal/mol	6.38E-17
$\Delta G = -146.7 \text{ kcal/mol}$	23-Jun-2008 11:48:29	-146.7 kcal/mol	6.38E-17
$\Delta G = -146.5 \text{ kcal/mol}$	23-Jun-2008 11:48:29	-146.5 kcal/mol	4.61E-17
$\Delta G = -146.5 \text{ kcal/mol}$	23-Jun-2008 11:48:29	-146.5 kcal/mol	4.61E-17
$\Delta G = -146.4 \text{ kcal/mol}$	23-Jun-2008 11:48:29	-146.4 kcal/mol	3.92E-17
$\Delta G = -146.4 \text{ kcal/mol}$	23-Jun-2008 11:48:29	-146.4 kcal/mol	3.92E-17
$\Delta G = -146.4 \text{ kcal/mol}$	23-Jun-2008 11:48:29	-146.4 kcal/mol	3.92E-17
$\Delta G = -146.3 \text{ kcal/mol}$	23-Jun-2008 11:48:29	-146.3 kcal/mol	3.33E-17
$\Delta G = -146.3 \text{ kcal/mol}$	23-Jun-2008 11:48:29	-146.3 kcal/mol	3.33E-17
$\Delta G = -146.3 \text{ kcal/mol}$	23-Jun-2008 11:48:29	-146.3 kcal/mol	3.33E-17
$\Delta G = -146.3 \text{ kcal/mol}$	23-Jun-2008 11:48:29	-146.3 kcal/mol	3.33E-17

Elements of structure ' $\Delta G = -146.7 \text{ kcal/mol}$ ': $\Delta G = -146.7 \text{ kcal/mol}$

- Stem with bifurcation at 1..484: $\Delta G = -142.0 \text{ kcal/mol}$
- U-A helix end at (1, 484): $\Delta G = 0.5 \text{ kcal/mol}$
- Dangling A at 485, dangling from position 484: $\Delta G = -0.8 \text{ kcal/mol}$
- Dangling G at 487, dangling from position 488: $\Delta G = -0.2 \text{ kcal/mol}$
- Stem with hairpin at 488..505: $\Delta G = -2.8 \text{ kcal/mol}$
- U-A helix end at (488, 505): $\Delta G = 0.5 \text{ kcal/mol}$
- Dangling A at 506, dangling from position 505: $\Delta G = -0.8 \text{ kcal/mol}$
- Dangling A at 507, dangling from position 508: $\Delta G = -0.3 \text{ kcal/mol}$
- Stem with hairpin at 508..539: $\Delta G = -0.5 \text{ kcal/mol}$
- U-A helix end at (508, 539): $\Delta G = 0.5 \text{ kcal/mol}$
- Dangling A at 540, dangling from position 539: $\Delta G = -0.8 \text{ kcal/mol}$

Figure 26.16: The secondary structure table with the list of structures to the left, and to the right the substructures of the selected structure.

contributes a free energy given by the sum of its nested substructure energies and energies of its nested structure elements.

The substructure elements to the right are ordered after their occurrence in the sequence; they are described by a region (the sequence positions covered by this substructure) and an energy contribution. Three examples of mixed substructure elements are "Stem base pairs", "Stem with bifurcation" and "Stem with hairpin".

The "Stem base pairs"-substructure is simply a union of stacking elements. It is given by a joined set of base pair positions and an energy contribution displaying the sum of all stacking element-energies.

The "Stem with bifurcation"-substructure defines a substructure enclosed by a specified base pair with and with energy contribution ΔG . The substructure contains a "Stem base pairs"-substructure and a nested bifurcated substructure (multi loop). Also bulge and interior loops can occur separating stem regions.

The "Stem with hairpin"-substructure defines a substructure starting at a specified base pair with an enclosed substructure-energy given by ΔG . The substructure contains a "Stem base pairs"-substructure and a hairpin loop. Also bulge and interior loops can occur, separating stem regions.

In order to describe the tree ordering of different substructures, we use an example as a starting point (see figure 26.17).

The structure is a (disjoint) nested union of a "Stem with bifurcation"-substructure and a dangling nucleotide. The nested substructure energies add up to the total energy. The "Stem with bifurcation"-substructure is again a (disjoint) union of a "Stem base pairs"-substructure joining position 1-7 with 64-70 and a multi loop structure element opened at base pair(7,64). To see these structure elements, simply expand the "Stem with bifurcation" node (see figure 26.18).

The multi loop structure element is a union of three "Stem with hairpin"-substructures and contributions to the multi loop opening considering multi loop base pairs and multi loop arcs.

Selecting an element in the table to the right will make a corresponding selection in the **Show Secondary Structure 2D View** (🔍) if this is also open and if the "Follow structure selection" has been set in the editors side panel. In figure 26.18 the "Stem with bifurcation" is selected in the table, and this part of the structure is high-lighted in the Secondary Structure 2D view.

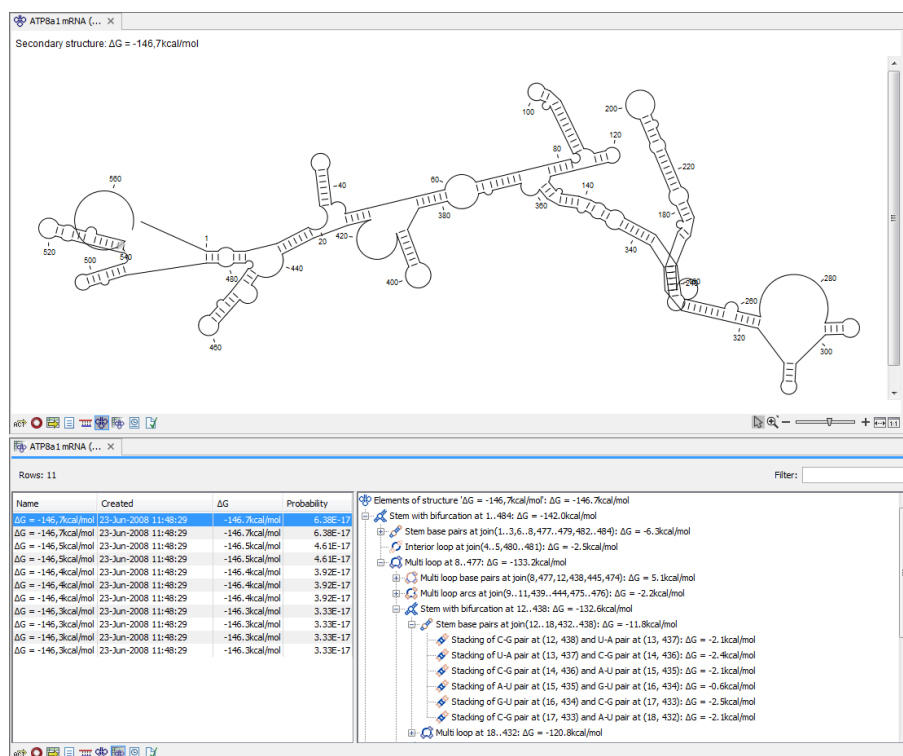


Figure 26.17: A split view showing a structure table to the right and the secondary structure 2D view to the left.

The correspondence between the table and the structure editor makes it easy to inspect the thermodynamic details of the structure while keeping a visual overview as shown in the above figures.

Handling multiple structures The table to the left offers a number of tools for working with structures. Select a structure, right-click, and the following menu items will be available:

- **Open Secondary Structure in 2D View** (🔍). This will open the selected structure in the Secondary structure 2D view.
- **Annotate Sequence with Secondary Structure**. This will add the structure elements as annotations to the sequence. Note that existing structure annotations will be removed.
- **Rename Secondary Structure**. This will allow you to specify a name for the structure to be displayed in the table.
- **Delete Secondary Structure**. This will delete the selected structure.
- **Delete All Secondary Structures**. This will delete all the selected structures. Note that once you save and close the view, this operation is irreversible. As long as the view is open, you can **Undo** (↶) the operation.

- **Show first.** If not all structures are shown, this can be used to determine the number of structures to be shown.
- **Sort by.** When you select to display e.g. four out of eight structures, this option determines which the "first four" should be.
 - Sort by ΔG .
 - Sort by name.
 - Sort by time of creation.

If these three options do not provide enough control, you can rename the structures in a meaningful alphabetical way so that you can use the "name" to display the desired ones.

- **Base pair symbol.** How a base pair should be represented (see figure 26.19).
- **Unpaired symbol.** How bases which are not part of a base pair should be represented (see figure 26.19).
- **Height.** When you zoom out, this option determines the height of the symbols as shown in figure 26.20 (when zoomed in, there is no need for specifying the height).
- **Base pair probability.** See section 26.2.4 below).

When you zoom in and out, the appearance of the symbols change. In figure 26.19, the view is zoomed in. In figure 26.20 you see the same sequence zoomed out to fit the width of the sequence.

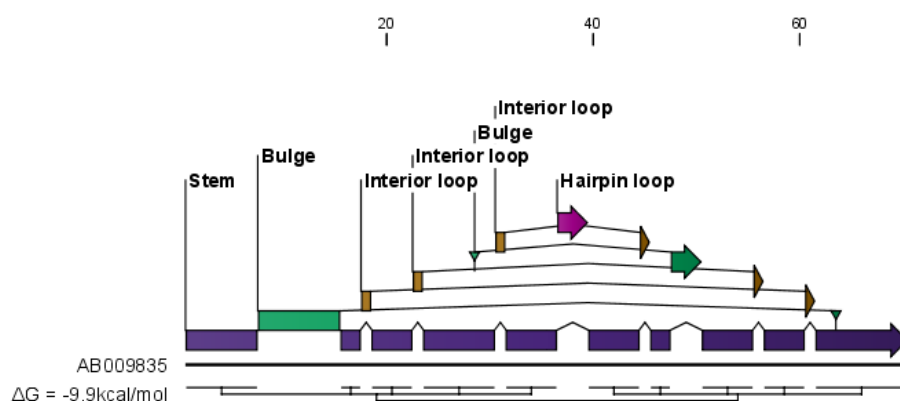


Figure 26.20: The secondary structure visualized below the sequence and with annotations shown above. The view is zoomed out to fit the width of the sequence.

26.2.4 Probability-based coloring

In the **Side Panel** of both linear and secondary structure 2D views, you can choose to color structure symbols and sequence residues according to the probability of base pairing / not base pairing, as shown in figure 26.4.

In the linear sequence view (ACT), this is found in **Nucleotide info** under **Secondary structure**, and in the secondary structure 2D view (ACT), it is found under **Residue coloring**.

For both paired and unpaired bases, you can set the foreground color and the background color to a gradient with the color at the left side indicating a probability of 0, and the color at the right side indicating a probability of 1.

Note that you have to **Zoom to 100%** () in order to see the coloring.

26.3 Evaluate structure hypothesis

Hypotheses about an RNA structure can be tested using *CLC Genomics Workbench*. A structure hypothesis H is formulated using the structural constraint annotations described in section 26.1.2. By adding several annotations complex structural hypotheses can be formulated (see 26.21).

Given the set S of all possible structures, only a subset of these S_H will comply with the formulated hypotheses. We can now find the probability of H as:

$$P(H) = \frac{\sum_{s_H \in S_H} P(s_H)}{\sum_{s \in S} P(s)} = \frac{PF_H}{PF_{\text{full}}},$$

where PF_H is the partition function calculated for all structures permissible by H (S_H) and PF_{full} is the full partition function. Calculating the probability can thus be done with two passes of the partition function calculation, one with structural constraints, and one without. 26.21.

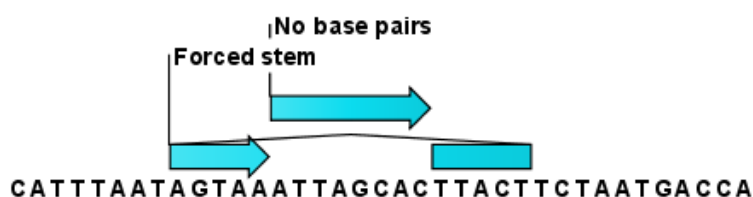


Figure 26.21: Two constraints defining a structural hypothesis.

26.3.1 Selecting sequences for evaluation

Start the **Evaluate Structure Hypothesis** tool by going to:

Tools | Classical Sequence Analysis () | **RNA Structure** () | **Evaluate Structure Hypothesis** ()

This opens the dialog shown in figure 26.22.

If you had selected sequences before running the tool, those sequences will be listed in the **Selected Elements** pane of the dialog. Use the arrows to add or remove sequences or sequence lists from the selected elements. Note that the selected sequences must contain a structure hypothesis in the form of manually added constraint annotations.

Click **Next** to adjust evaluation parameters (see figure 26.23).

The partition function algorithm includes a number of advanced options:

- **Avoid isolated base pairs.** The algorithm filters out isolated base pairs (i.e. stems of length 1).

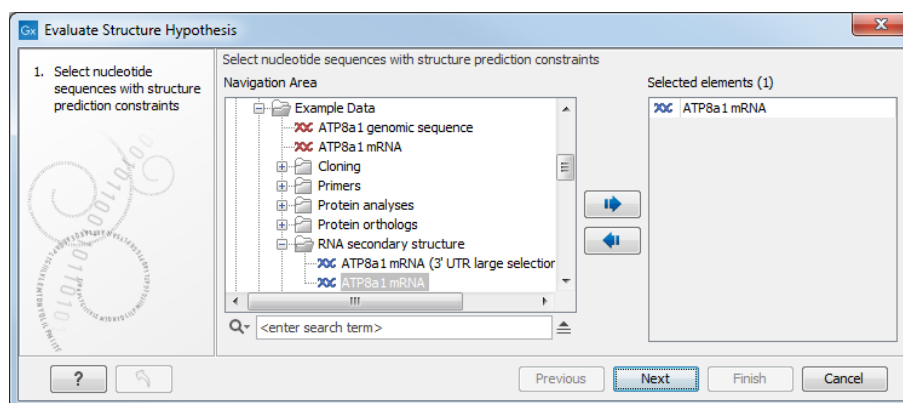


Figure 26.22: Selecting RNA or DNA sequences for evaluating structure hypothesis.

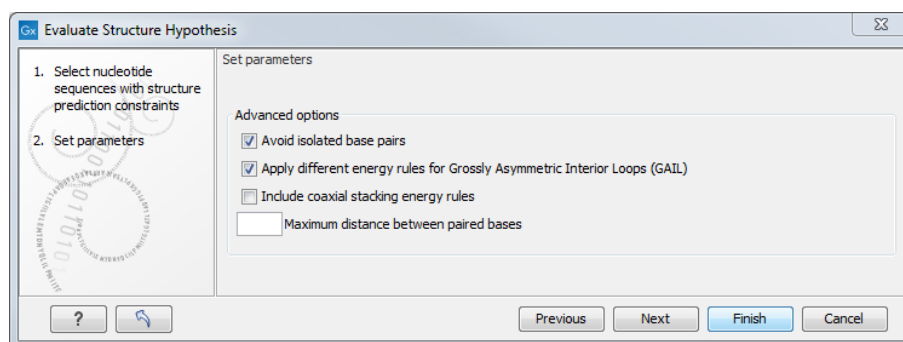


Figure 26.23: Adjusting parameters for hypothesis evaluation.

- **Apply different energy rules for Grossly Asymmetric Interior Loops (GAIL).** Compute the minimum free energy applying different rules for Grossly Asymmetry Interior Loops (GAIL). A Grossly Asymmetry Interior Loop (GAIL) is an interior loop that is $1 \times n$ or $n \times 1$ where $n > 2$ (see <http://mfold.rna.albany.edu/doc/mfold-manual/node5.php>)
- **Include coaxial stacking energy rules.** Include free energy increments of coaxial stacking for adjacent helices [Mathews et al., 2004].

26.3.2 Probabilities

After evaluation of the structure hypothesis an annotation is added to the input sequence. This annotation covers the same region as the annotations that constituted the hypothesis and contains information about the probability of the evaluated hypothesis (see figure 26.24).

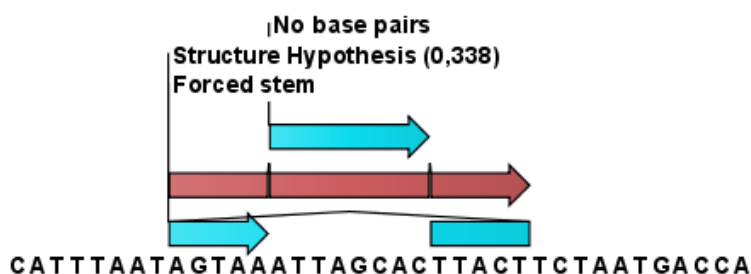


Figure 26.24: This hypothesis has a probability of 0.338 as shown in the annotation.

26.4 Structure scanning plot

In *CLC Genomics Workbench* it is possible to scan larger sequences for the existence of local conserved RNA structures. The structure scanning approach is similar in spirit to the works of [Workman and Krogh, 1999] and [Clote et al., 2005]. The idea is that if natural selection is operating to maintain a stable local structure in a given region, then the minimum free energy of the region will be markedly lower than the minimum free energy found when the nucleotides of the subsequence are distributed in random order.

The algorithm works by sliding a window along the sequence. Within the window, the minimum free energy of the subsequence is calculated. To evaluate the significance of the local structure signal its minimum free energy is compared to a background distribution of minimum free energies obtained from shuffled sequences, using *Z*-scores [Rivas and Eddy, 2000]. The *Z*-score statistics corresponds to the number of standard deviations by which the minimum free energy of the original sequence deviates from the average energy of the shuffled sequences. For a given *Z*-score, the statistical significance is evaluated as the probability of observing a more extreme *Z*-score under the assumption that *Z*-scores are normally distributed [Rivas and Eddy, 2000].

26.4.1 Selecting sequences for scanning

To run the **Evaluate Structure Hypothesis** tool, go to:

Tools | Classical Sequence Analysis (🔍) | RNA Structure (📄) | Evaluate Structure Hypothesis (📊)

This opens the dialog shown in figure 26.25.

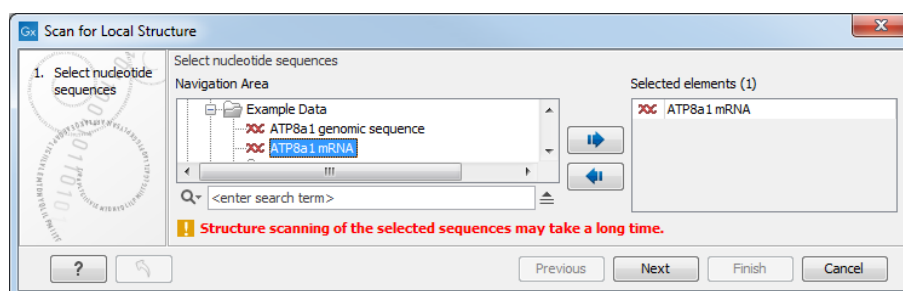


Figure 26.25: Selecting RNA or DNA sequences for structure scanning.

If you have selected sequences before running the tool, those sequences will be listed in the **Selected Elements** pane of the dialog. Use the arrows to add or remove sequences or sequence lists from the selected elements.

Click **Next** to adjust scanning parameters (see figure 26.26).

The first group of parameters pertain to the methods of sequence resampling. There are four ways of resampling, all described in detail in [Clote et al., 2005]:

- **Mononucleotide shuffling.** Shuffle method generating a sequence of the exact same mononucleotide frequency
- **Dinucleotide shuffling.** Shuffle method generating a sequence of the exact same dinucleotide frequency

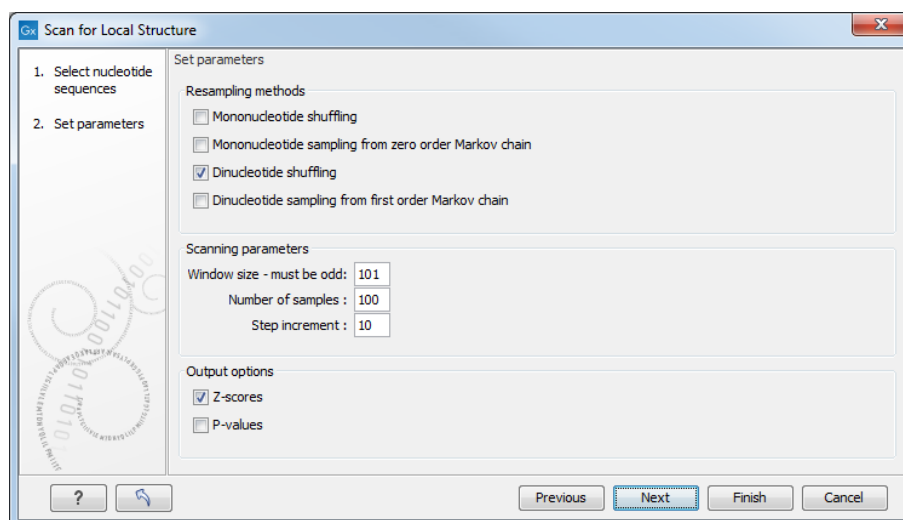


Figure 26.26: Adjusting parameters for structure scanning.

- **Mononucleotide sampling from zero order Markov chain.** Resampling method generating a sequence of the same expected mononucleotide frequency.
- **Dinucleotide sampling from first order Markov chain.** Resampling method generating a sequence of the same expected dinucleotide frequency.

The second group of parameters pertain to the scanning settings and include:

- **Window size.** The width of the sliding window.
- **Number of samples.** The number of times the sequence is resampled to produce the background distribution.
- **Step increment.** Step increment when plotting sequence positions against scoring values.

The third parameter group contains the output options:

- **Z-scores.** Create a plot of Z-scores as a function of sequence position.
- **P-values.** Create a plot of the statistical significance of the structure signal as a function of sequence position.

26.4.2 The structure scanning result

The output of the analysis are plots of Z -scores and probabilities as a function of sequence position. A strong propensity for local structure can be seen as spikes in the graphs (see figure 26.27).

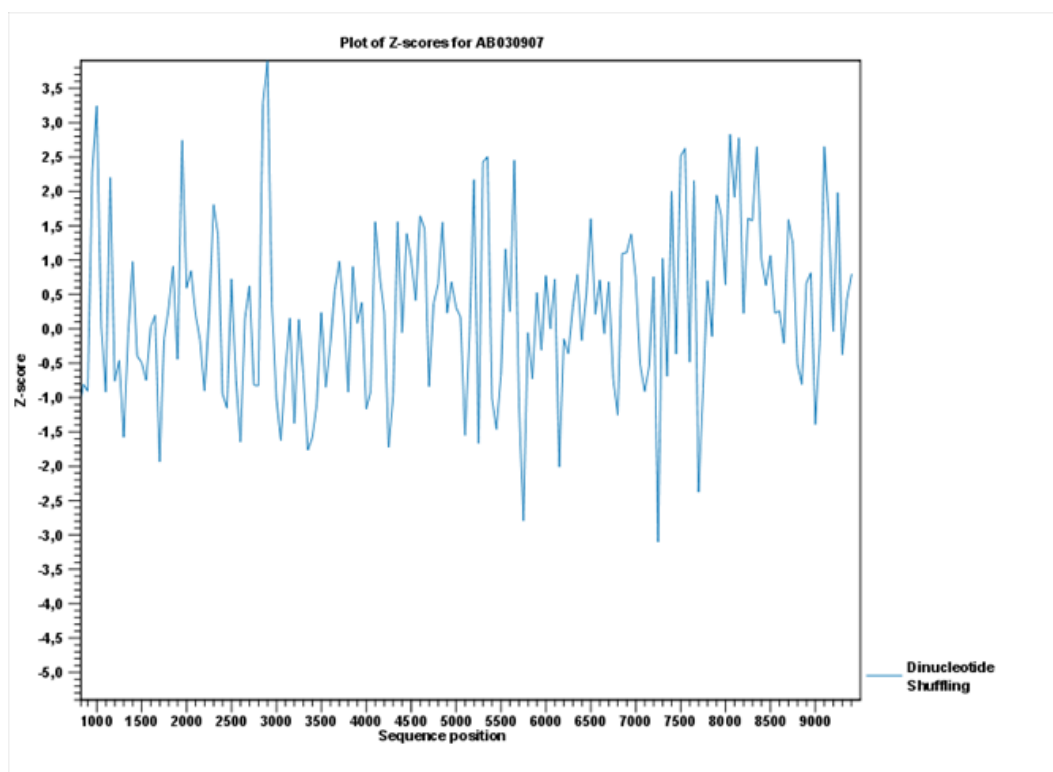


Figure 26.27: A plot of the Z-scores produced by sliding a window along a sequence.

26.5 Bioinformatics explained: RNA structure prediction by minimum free energy minimization

RNA molecules are hugely important in the biology of the cell. Besides their rather simple role as an intermediate messenger between DNA and protein, RNA molecules can have a plethora of biologic functions. Well known examples of this are the infrastructural RNAs such as tRNAs, rRNAs and snRNAs, but the existence and functionality of several other groups of non-coding RNAs are currently being discovered. These include micro- (miRNA), small interfering- (siRNA), Piwi interacting- (piRNA) and small modulatory RNAs (smRNA) [Costa, 2007].

A common feature of many of these non-coding RNAs is that the molecular structure is important for the biological function of the molecule.

Ideally, biological function is best interpreted against a 3D structure of an RNA molecule. However, 3D structure determination of RNA molecules is time-consuming, expensive, and difficult [Shapiro et al., 2007] and there is therefore a great disparity between the number of known RNA sequences and the number of known RNA 3D structures.

However, as it is the case for proteins, RNA tertiary structures can be characterized by secondary structural elements. These are defined by hydrogen bonds within the molecule that form several recognizable "domains" of secondary structure like stems, hairpin loops, bulges and internal loops (see below). Furthermore, the high degree of base-pair conservation observed in the evolution of RNA molecules shows that a large part of the functional information is actually contained in the secondary structure of the RNA molecule.

Fortunately, RNA secondary structure can be computationally predicted from sequence data allowing researchers to map sequence information to functional information. The subject of this

paper is to describe a very popular way of doing this, namely free energy minimization. For an in-depth review of algorithmic details, we refer the reader to [Mathews and Turner, 2006](#).

26.5.1 The algorithm

Consider an RNA molecule and one of its possible structures S_1 . In a stable solution there will be an equilibrium between unstructured RNA strands and RNA strands folded into S_1 . The propensity of a strand to leave a structure such as S_1 (the stability of S_1), is determined by the free energy change involved in its formation. The structure with the lowest free energy (S_{min}) is the most stable and will also be the most represented structure at equilibrium. The objective of minimum free energy (MFE) folding is therefore to identify S_{min} amongst all possible structures.

In the following, we only consider structures without pseudoknots, i.e. structures that do not contain any non-nested base pairs.

Under this assumption, a sequence can be folded into a single coherent structure or several sequential structures that are joined by unstructured regions. Each of these structures is a union of well described structure elements (see below for a description of these). The free energy for a given structure is calculated by an additive nearest neighbor model. Additive, means that the total free energy of a secondary structure is the sum of the free energies of its individual structural elements. Nearest neighbor, means that the free energy of each structure element depends only on the residues it contains and on the most adjacent Watson-Crick base pairs.

The simplest method to identify S_{min} would be to explicitly generate all possible structures, but it can be shown that the number of possible structures for a sequence grows exponentially with the sequence length [[Zuker and Sankoff, 1984](#)] leaving this approach unfeasible. Fortunately, a two step algorithm can be constructed which implicitly surveys all possible structures without explicitly generating the structures [[Zuker and Stiegler, 1981](#)]: The first step determines the free energy for each possible sequence fragment starting with the shortest fragments. Here, the lowest free energy for longer fragments can be expediently calculated from the free energies of the smaller sub-sequences they contain. When this process reaches the longest fragment, i.e., the complete sequence, the MFE of the entire molecule is known. The second step is called traceback, and uses all the free energies computed in the first step to determine S_{min} - the exact structure associated with the MFE. Acceptable calculation speed is achieved by using *dynamic programming* where sub-sequence results are saved to avoid recalculation. However, this comes at the price of a higher requirement for computer memory.

The structure element energies that are used in the recursions of these two steps, are derived from empirical calorimetric experiments performed on small molecules see e.g. [[Mathews et al., 1999](#)].

Suboptimal structures determination A number of known factors violate the assumptions that are implicit in MFE structure prediction. [[Schroeder et al., 1999](#)] and [[Chen et al., 2004](#)] have shown experimental indications that the thermodynamic parameters are sequence dependent. Moreover, [[Longfellow et al., 1990](#)] and [[Kierzek et al., 1999](#)], have demonstrated that some structural elements show non-nearest neighbor effects. Finally, single stranded nucleotides in multi loops are known to influence stability [[Mathews and Turner, 2002](#)].

These phenomena can be expected to limit the accuracy of RNA secondary structure prediction by free energy minimization and it should be clear that the predicted MFE structure may deviate

somewhat from the actual preferred structure of the molecule. This means that it may be informative to inspect the landscape of suboptimal structures which surround the MFE structure to look for general structural properties which seem to be robust to minor variations in the total free energy of the structure.

An effective procedure for generating a sample of suboptimal structures is given in [Zuker, 1989a]. This algorithm works by going through all possible Watson-Crick base pair in the molecule. For each of these base pairs, the algorithm computes the most optimal structure among all the structures that contain this pair, see figure 26.28.

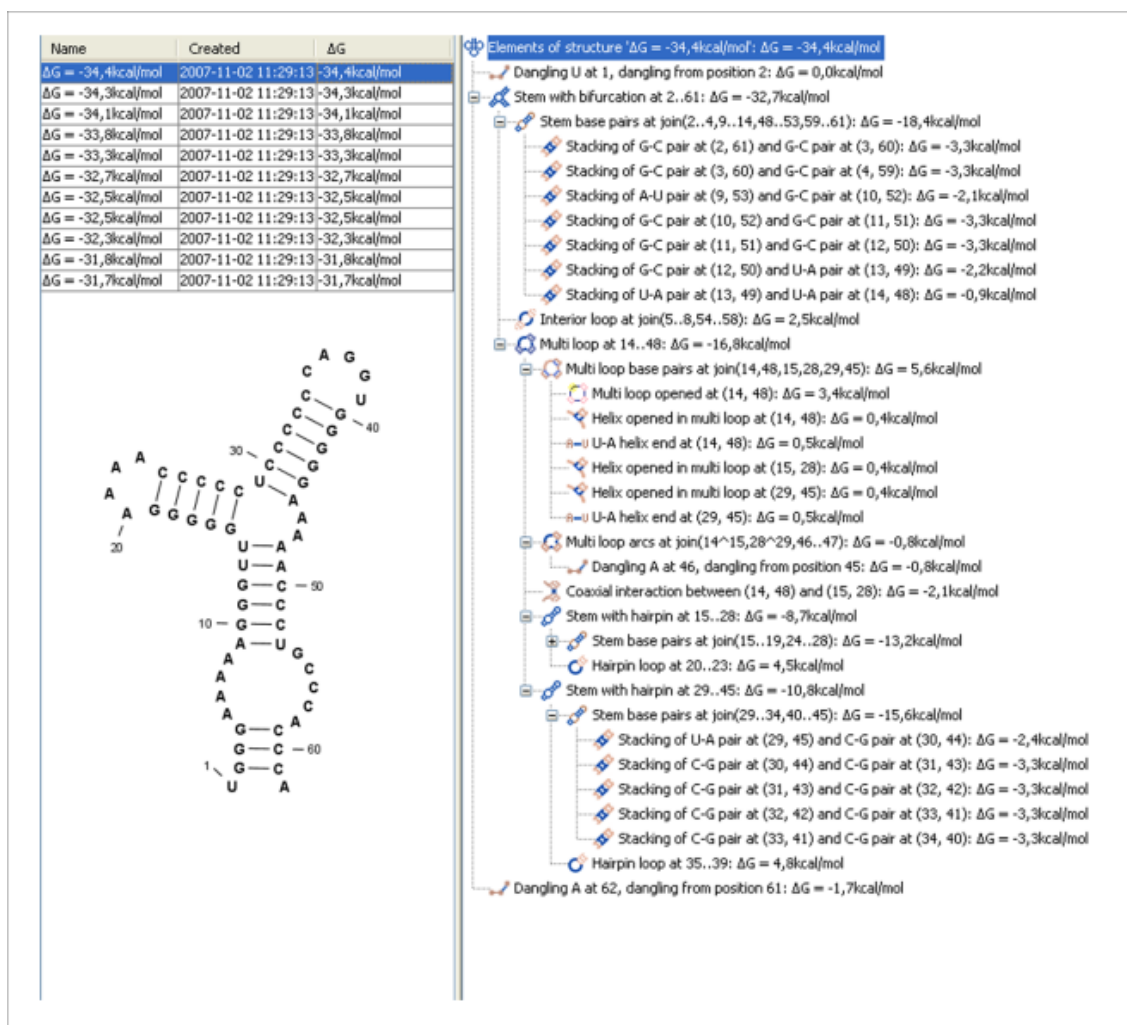


Figure 26.28: A number of suboptimal structures have been predicted using **CLC Genomics Workbench** and are listed at the top left. At the right hand side, the structural components of the selected structure are listed in a hierarchical structure and on the left hand side the structure is displayed.

26.5.2 Structure elements and their energy contribution

In this section, we classify the structure elements defining a secondary structure and describe their energy contribution.

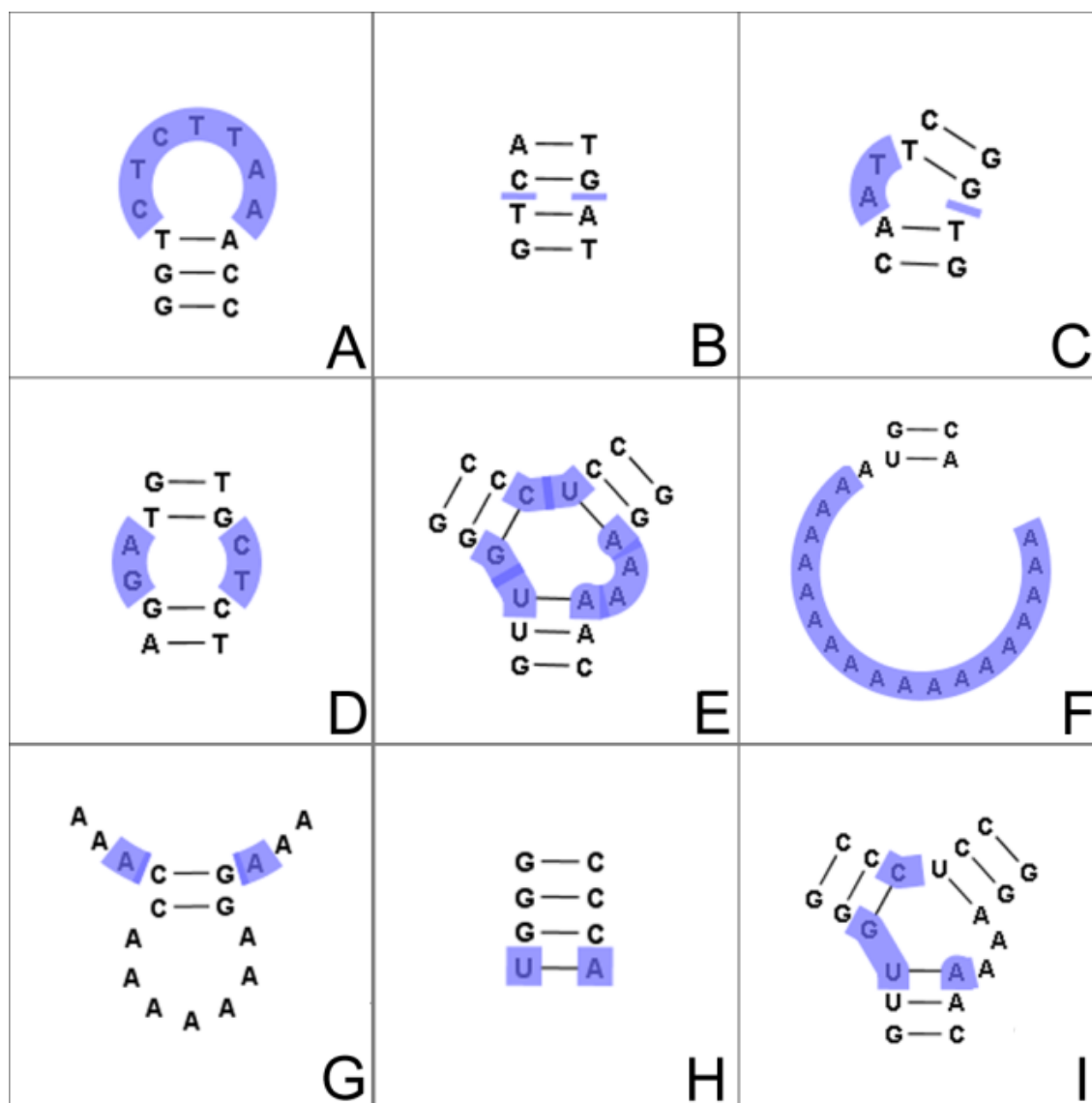


Figure 26.29: The different structure elements of RNA secondary structures predicted with the free energy minimization algorithm in **CLC Genomics Workbench**. See text for a detailed description.

Nested structure elements The structure elements involving nested base pairs can be classified by a given base pair and the other base pairs that are nested and *accessible* from this pair. For a more elaborate description we refer the reader to [Sankoff et al., 1983] and [Zuker and Sankoff, 1984].




If the nucleotides with position number (i, j) form a base pair and $i < k, l < j$, then we say that the base pair (k, l) is **accessible** from (i, j) if there is no intermediate base pair (i', j') such that $i < i' < k, l < j' < j$. This means that (k, l) is nested within the pair i, j and there is no other base pair in between.



Using the number of accessible base pairs, we can define the following distinct structure elements:

1. **Hairpin loop** (🌀). A base pair with 0 other accessible base pairs forms a *hairpin loop*. The energy contribution of a hairpin is determined by the length of the unpaired (loop) region




and the two bases adjacent to the closing base pair which is termed a terminal mismatch (see figure 26.29A).

2. A base pair with 1 accessible base pair can give rise to three distinct structure elements:

- **Stacking of base pairs** (). A *stacking* of two consecutive pairs occur if $i' - i = 1 = j - j'$. Only canonical base pairs ($A - U$ or $G - C$ or $G - U$) are allowed (see figure 26.29B). The energy contribution is determined by the type and order of the two base pairs.
- **Bulge** (). A *bulge loop* occurs if $i' - i > 1$ or $j - j' > 1$, but not both. This means that the two base pairs enclose an unpaired region of length 0 on one side and an unpaired region of length ≥ 1 on the other side (see figure 26.29C). The energy contribution of a bulge is determined by the length of the unpaired (loop) region and the two closing base pairs.
- **Interior loop** (). An *interior loop* occurs if both $i' - i > 1$ and $j - j' > 1$. This means that the two base pairs enclose an unpaired region of length ≥ 1 on both sides (see figure 26.29D). The energy contribution of an interior loop is determined by the length of the unpaired (loop) region and the four unpaired bases adjacent to the opening- and the closing base pair.

3. **Multi loop opened** (). A base pair with more than two accessible base pairs gives rise to a *multi loop*, a loop from which three or more stems are opened (see figure 26.29E). The energy contribution of a multi loop depends on the number of **Stems opened in multi-loop** () that protrude from the loop.

Other structure elements

- A collection of single stranded bases not accessible from any base pair is called an *exterior* (or *external*) *loop* (see figure 26.29F). These regions do not contribute to the total free energy.
- **Dangling nucleotide** (). A *dangling nucleotide* is a single stranded nucleotide that forms a stacking interaction with an adjacent base pair. A dangling nucleotide can be a 3' or 5'-dangling nucleotide depending on the orientation (see figure 26.29G). The energy contribution is determined by the single stranded nucleotide, its orientation and on the adjacent base pair.
- **Non-GC terminating stem** (). If a base pair other than a G-C pair is found at the end of a stem, an energy penalty is assigned (see figure 26.29H).
- **Coaxial interaction** (). Coaxial stacking is a favorable interaction of two stems where the base pairs at the ends can form a stacking interaction. This can occur between stems in a multi loop and between the stems of two different sequential structures. Coaxial stacking can occur between stems with no intervening nucleotides (adjacent stems) and between stems with one intervening nucleotide from each strand (see figure 26.29I). The energy contribution is determined by the adjacent base pairs and the intervening nucleotides.

Experimental constraints A number of techniques are available for probing RNA structures. These techniques can determine individual components of an existing structure such as the existence of a given base pair. It is possible to add such experimental constraints to the secondary structure prediction based on free energy minimization (see figure 26.30) and it has been shown that this can dramatically increase the fidelity of the secondary structure prediction [Mathews and Turner, 2006].



Figure 26.30: Known structural features can be added as constraints to the secondary structure prediction algorithm in **CLC Genomics Workbench**.

Part IV

High-throughput sequencing

Chapter 27

Tracks

Contents

27.1 Track types	703
27.2 Track lists	704
27.3 Working with tracks	708
27.3.1 Visualizing, zooming and navigating tracks	708
27.3.2 The Table view	710
27.3.3 The Chromosome Table view	711
27.3.4 Finding information in tracks	712
27.3.5 Extracting sequences from tracks	713
27.4 Reference data as tracks	716
27.5 Modify tracks	716
27.5.1 Annotate with Exon Numbers	717
27.5.2 Annotate with Nearby Information	718
27.5.3 Annotate with Overlap Information	719
27.5.4 Collapse Overlapping Annotations	719
27.5.5 Remove Information from Track	721
27.5.6 Resize Annotations	721
27.6 Filter tracks	724
27.6.1 Filter Annotations on Name	724
27.6.2 Filter Based on Overlap	724
27.7 Merge tracks	726
27.7.1 Merge Annotation Tracks	726
27.7.2 Merge Variant Tracks	727
27.8 Convert tracks	728
27.8.1 Convert to Tracks	728
27.8.2 Convert from Tracks	729
27.9 Graph tracks	731
27.9.1 Create GC Content Graph	731
27.9.2 Create Mapping Graph	733
27.9.3 Identify Graph Threshold Areas	735

Tracks provide a unified framework for the visualization, comparison and analysis of genome-scale studies such as whole-genome sequencing or exome resequencing projects. Information in tracks is tied to a genomic position. Each track type supports a particular type of data, with functionality relevant to the type of data available. Details about the different types of tracks are provided in section 27.1.

When the mouse cursor is hovered over items in the graphical view of a track, information about that item and about the location the cursor is hovered over is displayed in several places (figure 27.1):

- Just below the coordinates near the top. Where relevant, the following are also displayed:
 - The item name.
 - The strand the item is located on.
 - For items comprised of multiple parts, e.g. mRNAs containing multiple exons, the number of the part that the cursor is hovered over.
 - The position of the cursor relative to the start of the item.
- In the View Area: A tooltip containing information about the item.
- In the lower right corner of the Workbench frame, as relevant for the item type, including:
 - The item type.
 - The item name.
 - The position of the cursor relative to the start of the item.
 - For items comprised of multiple parts, e.g. mRNAs containing multiple exons, the number of the part that the cursor is hovered over.
 - The strand the item is located on.
 - The position of the cursor relative to the reference.
 - Other information relevant to the item. For example, for variants, the variant type and its size.

Compatible tracks can be stacked in a *Track List*, supporting comparative analysis (figure 27.1). See section 27.2 for details about creating and working with Track Lists.

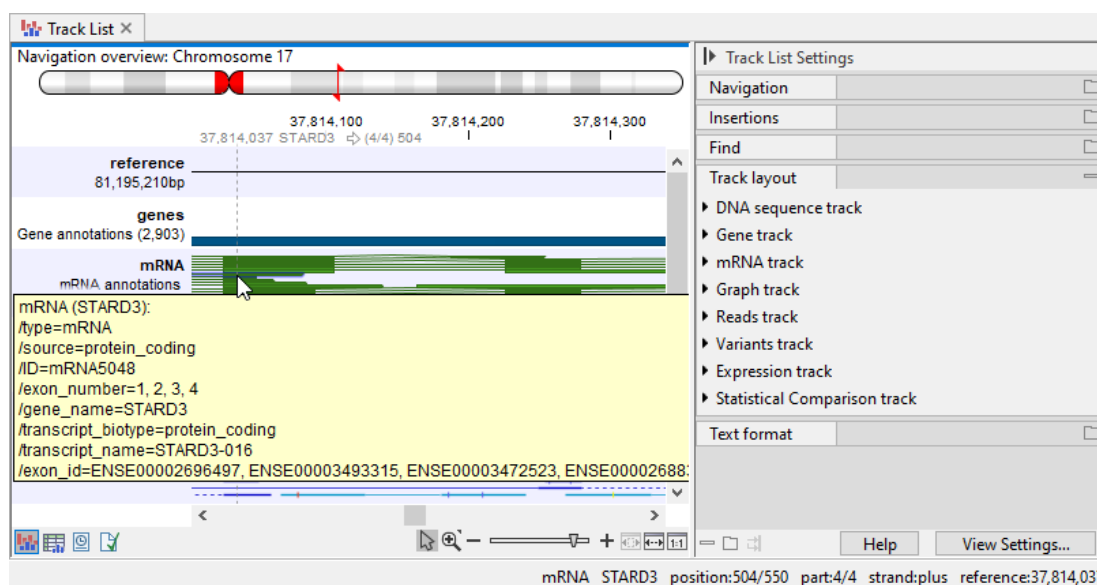


Figure 27.1: Compatible tracks have been added to a Track List and the cursor is hovered over an item in one of the tracks, revealing information about that item in three places: just under the coordinates near the top, in a tooltip near the item itself, and in the lower right corner of the Workbench frame.

27.1 Track types

Different track types support different types of data (figure 27.2). Key examples are:

Sequence Track (📄) A Sequence Track contains one or more sequences. It is usually used for the reference sequences of a genome or for sets of contigs.

Reads Track (📄) A Reads Track contains a read mapping, i.e. reads aligned against a set of reference sequences.

Variant Track (📄) A Variant Track contains information about various types of variants. These can include SNVs, MNVs, replacements, insertions and deletions, along with details about each variant, for example the allele, its length and frequency.

Annotation Track (📄) An Annotation Track contains information about a certain type of annotation, for example gene annotations, mRNA annotations, or a set of target regions. Some analyses require specific types of Annotation Tracks as input. For example, gene tracks and mRNA tracks are commonly used when running **RNA-Seq Analysis**, see section 33.4.1.

Coverage Graph Track (📄) A Coverage Graph Track contains a graphical display of the coverage at each position of a reads track. It can be produced by **Graph** tools, see section 27.9.

Expression Track (📄) An Expression Track contains expression values for genes or for transcripts. See section 33.4.1 for further details.

Statistical Comparison Track (📄) A Statistical Comparison Track contains results from a differential expression analysis. See section 33.6.5 for further details.

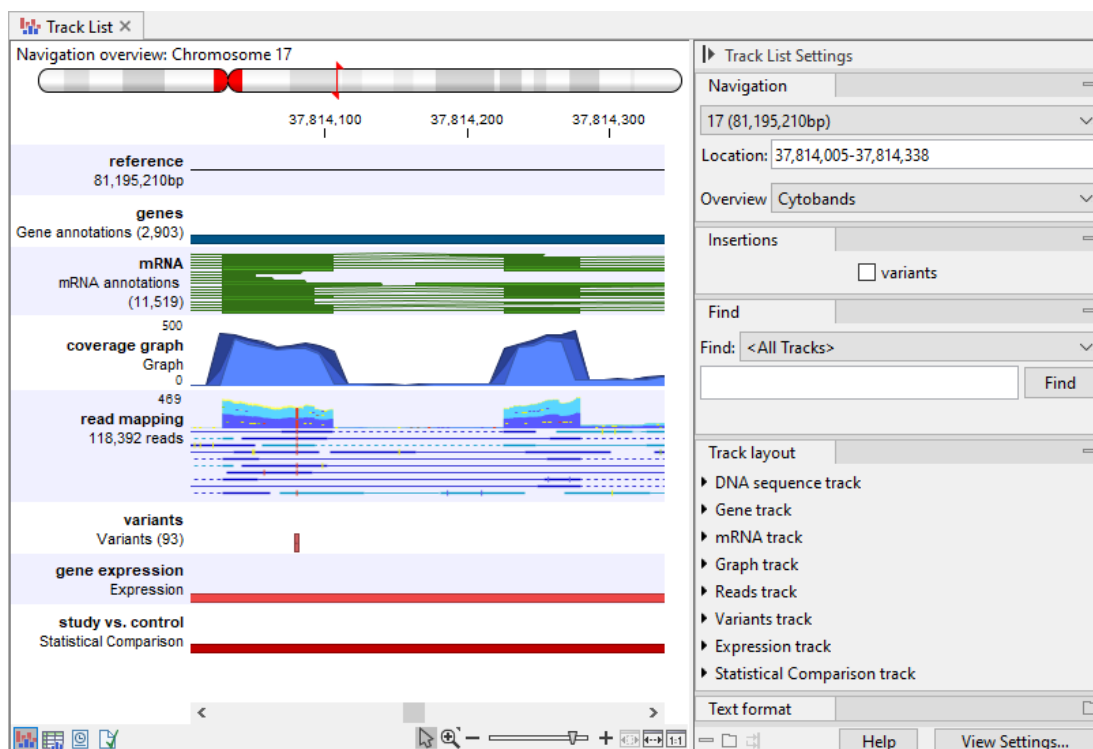


Figure 27.2: A track list containing different types of tracks. From the top: a Sequence Track, two Annotation Tracks containing genes and mRNAs, respectively, a Coverage Graph Track, a Reads Track, a Variant Track, an Expression Track, and a Statistical Comparison Track.

27.2 Track lists

Compatible tracks can be added to a **Track List** (📊), allowing multiple data types at the same genomic position to be viewed simultaneously (figure 27.3). Opening linked views of particular tracks makes navigating to positions of interest simple.

Different views are available using the icons at the bottom of the Track List. Of note is the **Chromosome Table** view (📊), which contains a row for each chromosome and a column for each track containing the number of items present in that track for that chromosome.

A Track List is made up of references to the tracks it contains, rather than containing the track data directly itself. When a track referred to by a Track List cannot be found, the text "Unresolved track" is displayed where the track would otherwise have been.

Track compatibility and copying Track Lists are described later in this section.

Creating Track Lists

Track Lists can be created in several ways:

- Run the **Create Track List** tool.

This tool can be launched from:

Tools | **Utility Tools** (🔧) | **Tracks** (📊) | **Create Track List** (📊)

or by clicking on the **Create Track List** button at the bottom of graphical track views.

- Use the menu item available from the top Toolbar:

New (📄) | **Track List** (📊)

This is also available from under the **File** menu.

- Select one or more tracks in the Navigation Area and drag them onto a track already open in the viewing area.
The selected tracks and the open track must be compatible. Track compatibility is described further below.
- Use the **Create Track List** element in a workflow (see section 14.2.6).

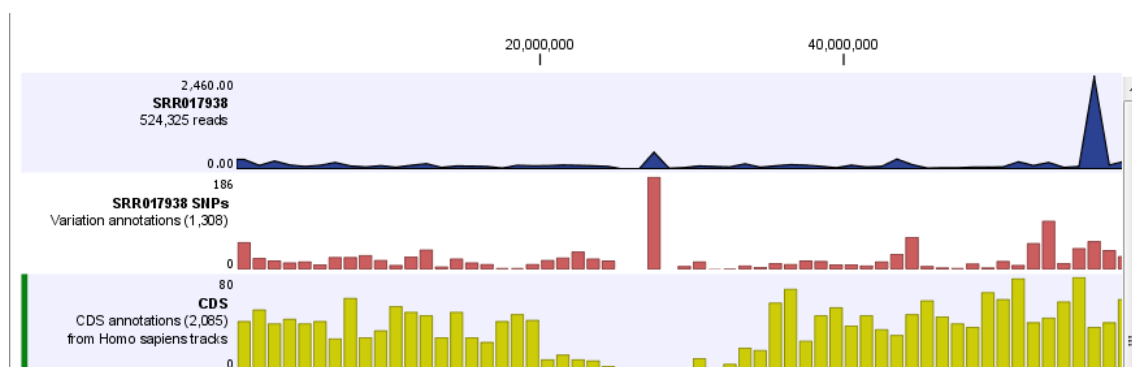


Figure 27.3: A track list referring to 3 tracks. By zooming into a given location, the details of the read mapping (top track), variants (middle track) and overlapping CDS annotations (bottom track) can be investigated.

Compatible tracks

Tracks must be compatible to be added to the same Track List. To be compatible, they must have the same positional coordinates, i.e. they contain the same number of chromosomes, and the chromosome lengths are the same in each track. Tracks generated using the same reference data will thus always be compatible, and can be added to a single Track List.

If two or more chromosomes have the same length, tracks can still be compatible as long as the names of the same-length chromosomes are unique within each track, and are the same between the tracks.

Working with tracks in Track Lists

The same functionality available when working with tracks can be used to work with tracks in Track Lists, (see section 27.3). This section focuses on functionality specific to Track Lists.

Common activities when working with Track Lists include:

- **Re-ordering tracks in a Track List** To move a track to a different location in the Track List, click on it, and, keeping the mouse button depressed, drag the track up or down to the desired position. Release the mouse button to drop the track at that position.

- **Opening tracks in a linked view** A track can be opened in linked view, often a table view, by:

- Double-clicking on a track name.
- Pressing the Ctrl key (⌘ on macs) and clicking on the name of the track of interest.
- Right-clicking on the track and choosing the option **Open This Track** from the context menu that appears.

See figure 27.4 for an example of a track opened in a linked view.

When working with linked views, clicking on an item or row in one view brings the focus in other views to the corresponding location or information. See section 27.3.2 and section 2.1 for information about the track table view and linked views, respectively.

- **Adding more tracks to a Track List** Any track based on a compatible genome can be added to an existing Track List by:
 - Dragging them from the **Navigation Area** into the open Track List.
 - Right-clicking anywhere in the Track List and choosing **Include More Tracks** from the context menu that appears.
- **Remove tracks from a Track List** To remove a track, right-click on a track and choose **Remove Track** from the context menu that appears.

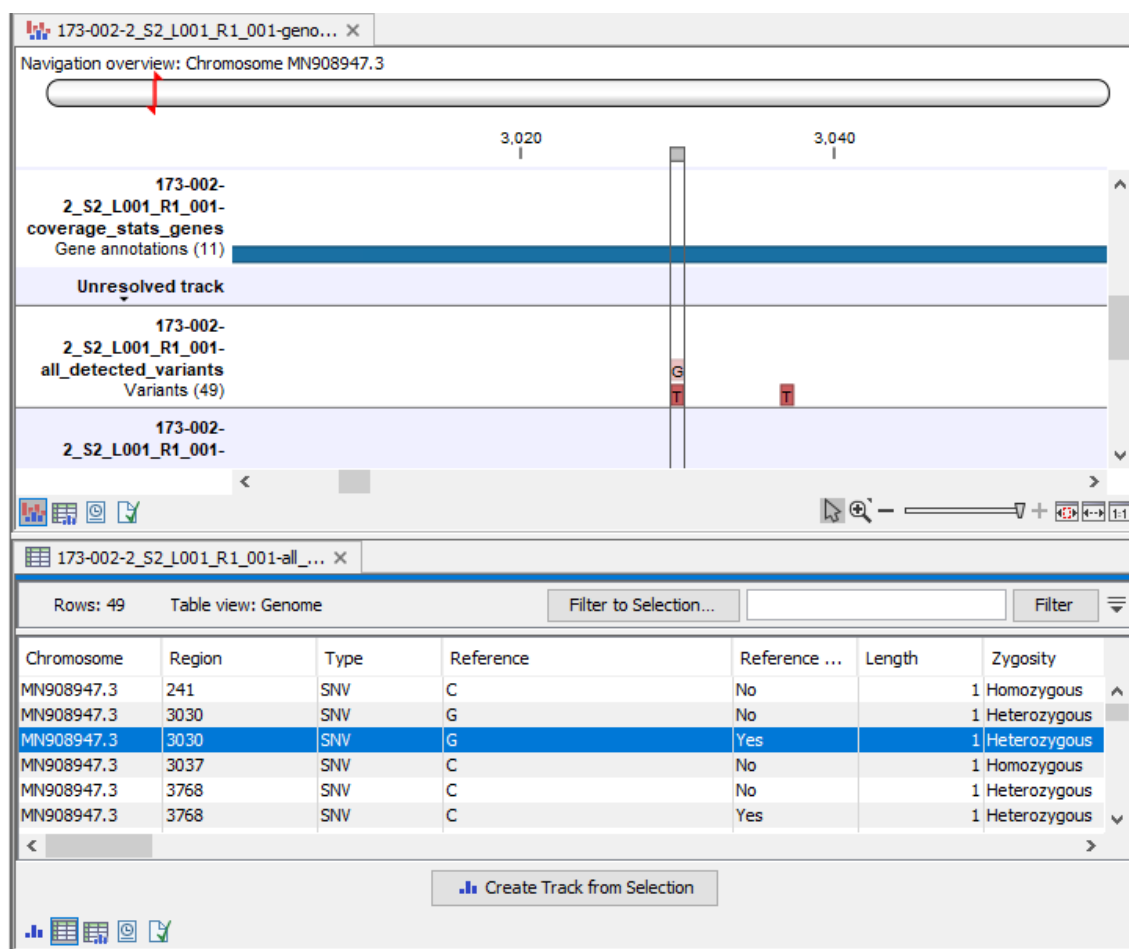


Figure 27.4: A variant track has been opened in a linked table view by double-clicking on the name of the track in the Track List. A row was selected in the table, moving the focus in the graphical view to the location referred to in that row. Conversely, double-clicking on a variant in the graphical view would select the relevant row(s) in the table view.

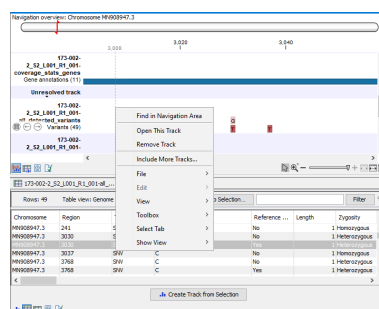


Figure 27.5: Right-click on a track to reveal options relevant for working with the Track List.

Copying Track Lists

When a Track List and all the tracks it refers to are copied in a single action, the new Track List will refer to the new copies of the tracks. If only the Track List is copied, the new Track List will refer to the same track elements that the original Track List.

If a subset of the tracks referred to are copied, then references to the copied tracks are updated in the new Track List, while references to the original tracks are maintained for those not copied.

General information about creating copies of elements is described in section 3.1.6.

27.3 Working with tracks

27.3.1 Visualizing, zooming and navigating tracks

Side Panel

The Side Panel is shown to the right of a track opened in the View Area (figure 27.6). The settings generally allow users to navigate the track using a specific position on a specific chromosome, to find a particular nucleotide sequence or annotation, and to change the text format.

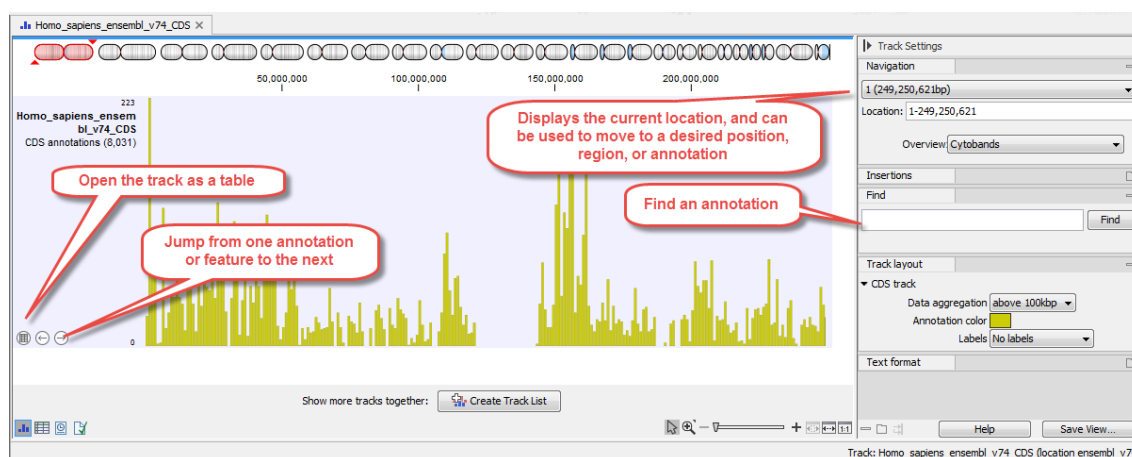


Figure 27.6: A CDS track with its Side Panel visible to the right.

In the Navigation section of the Track Side Panel, the upper field gives information about which chromosome is currently shown. The drop-down list can be used to jump to a different chromosome. The Location field displays the start and end positions of the shown region of the chromosome, but can also be used to navigate the track: enter a range or a single location point to get the visualization to zoom in the region of interest. It is also possible to enter the name of a chromosome (MT: or 5:), the name of a gene or transcript (BRCA" or DHFR-001), or even the range on a particular gene or transcript (BRCA2:122-124) Finally, the Overview drop-down menu defines what is shown above the track: cytobands, or cytobands with aggregated data. It can also be hidden all together.

Additional settings specific to the type of track being open can be available. For example, you can change a Reads track layout as explained in section 30.3.2. Similarly, when working with annotation tracks, the settings **Track Layout | Labels** controls where labels will be displayed in relation to the annotations (figure 27.7).

Once you have changed the visualization of the track to your liking, it is possible to save the settings as described in section 4.6.

Additional tools

In a reads track, a vertical scroll bar will appear to the right of the reads when hovering on them to navigate through high coverage regions.

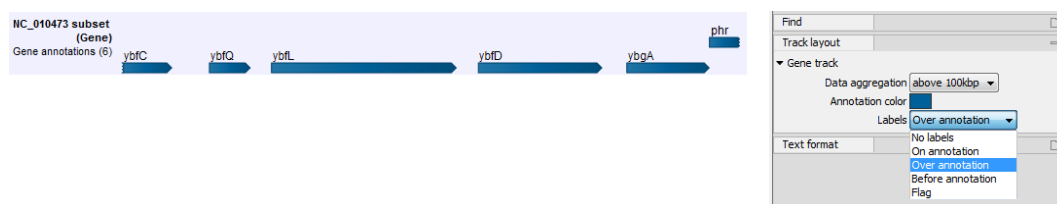


Figure 27.7: The Side Panel for annotation tracks showing the Labels drop down menu.

Specified tracks have buttons that will appear under the track name on the left side of the View Area (highlighted in figure 27.8): these buttons can be used to open the track as a table, or jump to the previous and next element.



Figure 27.8: Hovering on a track will show additional buttons under the track name.

Zooming

Clicking on icons in the bottom, right corner of the View area allows you to zoom in and zoom out.

- To zoom in to 100 % to see the data at base level, click the **Zoom to base level** (1:1) icon.
- To zoom out to see all the data, click the **Zoom to Fit** (←→) icon.


When zooming out you will see that the data is visualized in an aggregated format using a density bar plot or a graph.

Navigation and zooming shortcuts

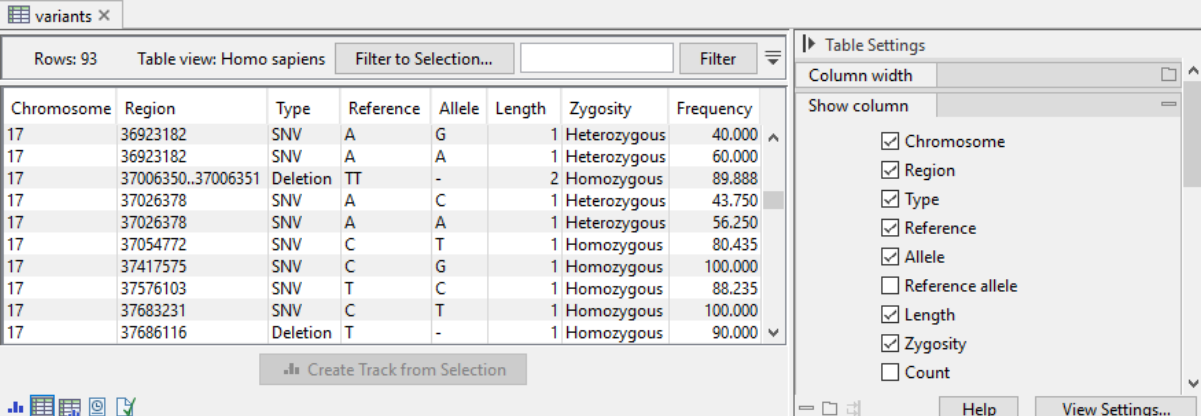
You can also use the zoom and scroll shortcuts described in the table below:

Action	Windows/Linux	macOS
Vertical scroll in reads tracks	Alt + Scroll wheel	Alt + Scroll wheel
Vertical scroll in reads tracks, fast	Shift+Alt+Scroll wheel	Shift+Alt+Scroll wheel
Vertical zoom in graph tracks	Ctrl + Scroll wheel	⌘ + Scroll wheel
Zoom	Ctrl + Scroll wheel	⌘ + Scroll wheel
Zoom In Mode	Ctrl + 2	⌘ + 2
Zoom In (without clicking)	'+' (plus)	'+' (plus)
Zoom Out Mode	Ctrl + 3	⌘ + 3
Zoom Out (without clicking)	'-' (minus)	'-' (minus)
Zoom to base level	Ctrl + 0	⌘ + 0
Zoom to fit screen	Ctrl + 6	⌘ + 6
Zoom to selection	Ctrl + 5	⌘ + 5
Reverse zoom mode	press and hold Shift	press and hold Shift

27.3.2 The Table view

The information stored by many track types (section 27.1) can be viewed in a table. To open the table, click on  at the bottom of the track.

The table contains a row for each item in the track. Positional information (e.g. Chromosome, Region) is always included. The other information present depends on the type of data in the track (figure 27.9).



Chromosome	Region	Type	Reference	Allele	Length	Zygosity	Frequency
17	36923182	SNV	A	G	1	Heterozygous	40.000
17	36923182	SNV	A	A	1	Heterozygous	60.000
17	37006350..37006351	Deletion	TT	-	2	Homozygous	89.888
17	37026378	SNV	A	C	1	Heterozygous	43.750
17	37026378	SNV	A	A	1	Heterozygous	56.250
17	37054772	SNV	C	T	1	Homozygous	80.435
17	37417575	SNV	C	G	1	Homozygous	100.000
17	37576103	SNV	T	C	1	Homozygous	88.235
17	37683231	SNV	C	T	1	Homozygous	100.000
17	37686116	Deletion	T	-	1	Homozygous	90.000


Figure 27.9: Table view of a variant track. Positional information is provided in the first two columns, with the remaining columns containing information relevant to variant data. The full list of columns available for this track is provided in the Side Panel. Only the columns selected there are displayed in the table.

See chapter 9 for further information about working with tables.

Linked views

Opening the graphical and table views of a track in linked views makes it easy to navigate to positions of interest. Selecting rows in the table selects the corresponding items in the graphical view and move the focus in the graphical view to those items where possible. Conversely, selecting items in the graphical view selects the corresponding rows in the table.

From the graphical view of a track, the table can be opened in a linked view using one of the following methods:

- Press Ctrl (⌘ on Mac) while clicking .
- Right-click on the track tab, and select "Show View | Table".


For further details about working with different views for the same data element, see section [2.1](#).

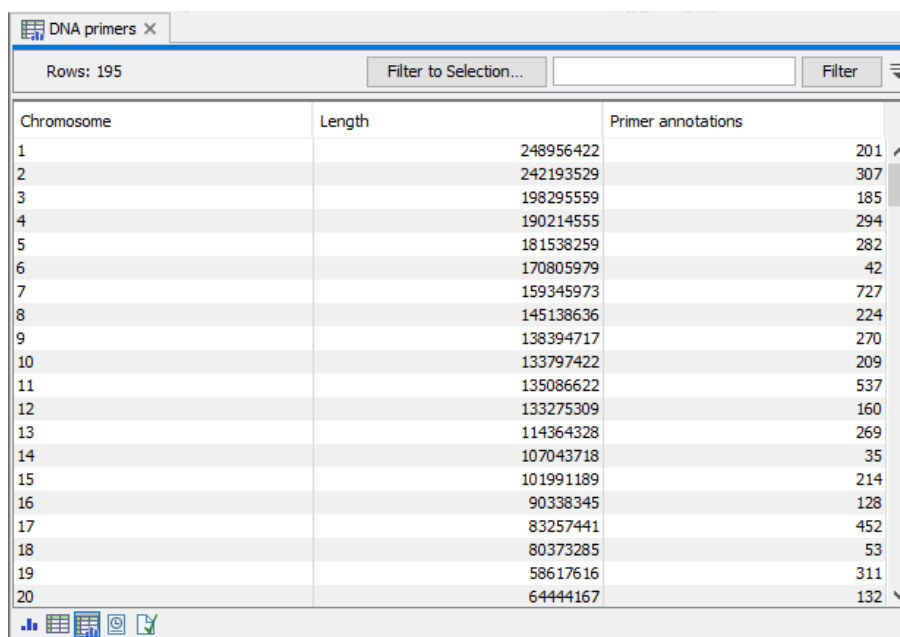
Note: Filtering the table in a linked view, so that only a subset of rows is visible, does not affect the contents of the graphical view. To manually create a track containing a subset of the available items, select the rows of interest and click on the Create Track from Selection button at the bottom of the table view. Several tools for creating tracks with a subset of items are also available, including:

- Filter Annotations on Name, section [27.6.1](#)
- Filter Based on Overlap, section [27.6.2](#)
- Filter on Custom Criteria, section [37.3](#)

27.3.3 The Chromosome Table view

The Chromosome Table view provides an overview of the data contained in tracks or track lists, summarized by chromosome (figure [27.10](#)). Clicking on a row in this view causes the focus to jump to the selected chromosome in other open views of the same data element. See section [2.1.4](#) for information about working with split views.

To open the Chromosome Table view, click on the  icon, present at the bottom of supported track types and track lists.



Chromosome	Length	Primer annotations
1	248956422	201
2	242193529	307
3	198295559	185
4	190214555	294
5	181538259	282
6	170805979	42
7	159345973	727
8	145138636	224
9	138394717	270
10	133797422	209
11	135086622	537
12	133275309	160
13	114364328	269
14	107043718	35
15	101991189	214
16	90338345	128
17	83257441	452
18	80373285	53
19	58617616	311
20	64444167	132

Figure 27.10: The Chromosome Table view gives an overview of data contained in a track or a track list.

27.3.4 Finding information in tracks

Basic search functionality is provided in the **Find** palette of the Side Panel of many track types and track lists (figure 27.11). Using the Find functionality, you can search exact terms, or partial terms (see below). When searching in a track, the search starts at the selected chromosome and returns the first match found. Clicking on the Find again returns the next match, and so on. When searching in a track list, all tracks are searched by default. However, the Find palette in a track list includes a drop down menu of the tracks, so the search can be limited to just a single, specified track.

To find all items matching a certain search term, or to search with a combination of terms, open the table view and use table filtering functionality (see chapter 9.2) or use dedicated tools for filtering the track such as Filter Annotations on Name, section 27.6.1 or Filter on Custom Criteria, section 37.3.

Enter a word in the Find field and press on the Find button to start the search for exactly that term. Add an asterisk to the start, end or to both ends of the word to make the search more general. For example:

- **BRCA*** would find terms starting with BRCA, for example, BRCA2.
- ***RCA1** would find terms ending in RCA1, for example BRCA1 and SMARCA1.
- ***RCA*** would find terms containing RCA, for example BRCA1, BRCA2 and SMARCA1.

Under the Find field, the progress of the search is reported, and then the first result, when a hit has been found. The tooltip provides details about the hit (figure 27.12).

More advanced searches can be performed by filtering the table view, see section 27.3.2 (figure 27.13).

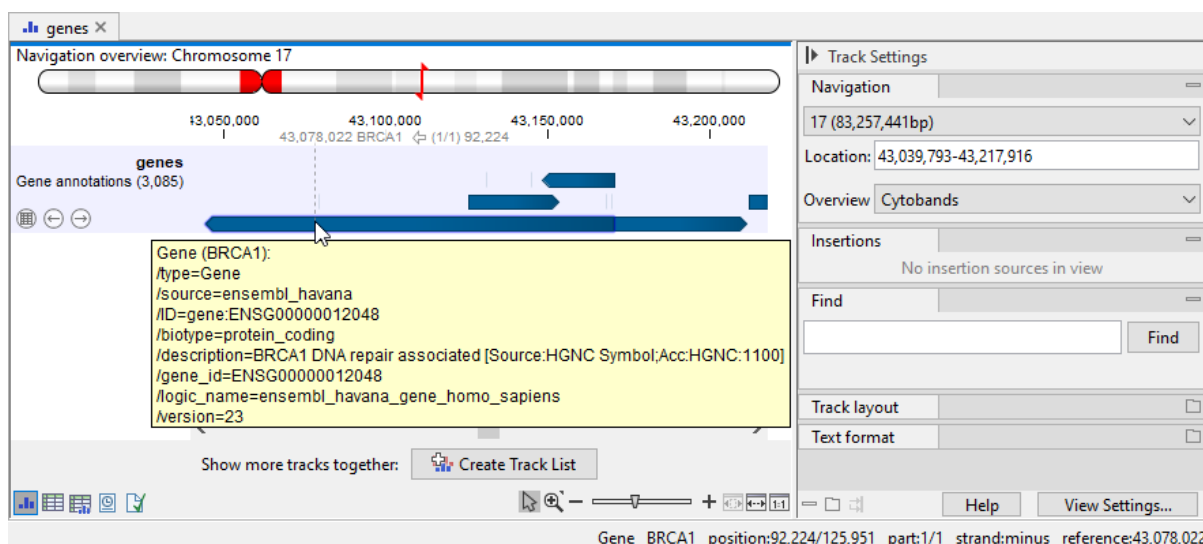


Figure 27.11: The *BRCA1* gene and the information stored for it, revealed as a tooltip.

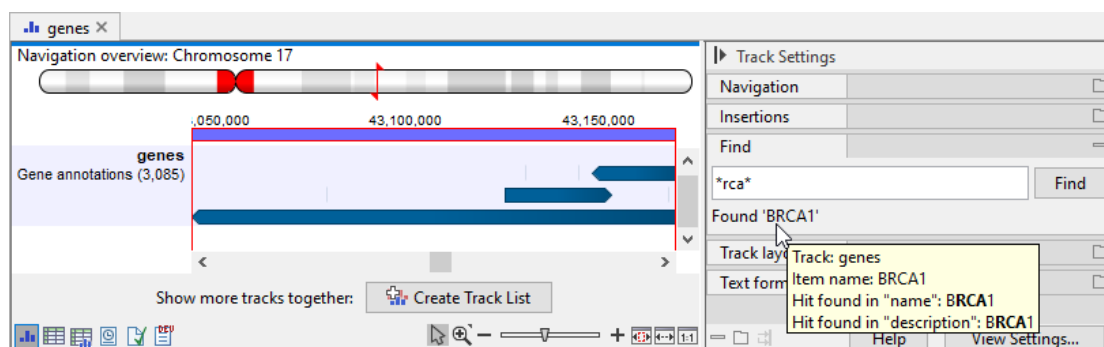


Figure 27.12: The *BRCA1* gene was found by this search. The tooltip contains details about where the term that was searched for was found.

27.3.5 Extracting sequences from tracks

This section describes functionality for extracting reads from reads tracks and a (portion of a) sequence from a sequence track. Extracting reads from stand-alone read mappings is described in section 22.7.6 and extracting a (portion of a) sequence from sequence lists is described in section 15.2.2.

Extracting reads from reads tracks

Reads can be extracted from reads tracks in multiple ways:

- **Create Reads Track from Selection.** Available from the right-click menu of the reads track (figure 27.14). A new reads track consisting of just the reads overlapping the selected region will be created. Options are available to specify the nature of the reads to be included (e.g. match specificity, paired status, etc.). These options are the same as those provided in the **Extract Reads** tool.
- **Extract Reads.** Available from the Tools menu. It extracts all reads or a subset of

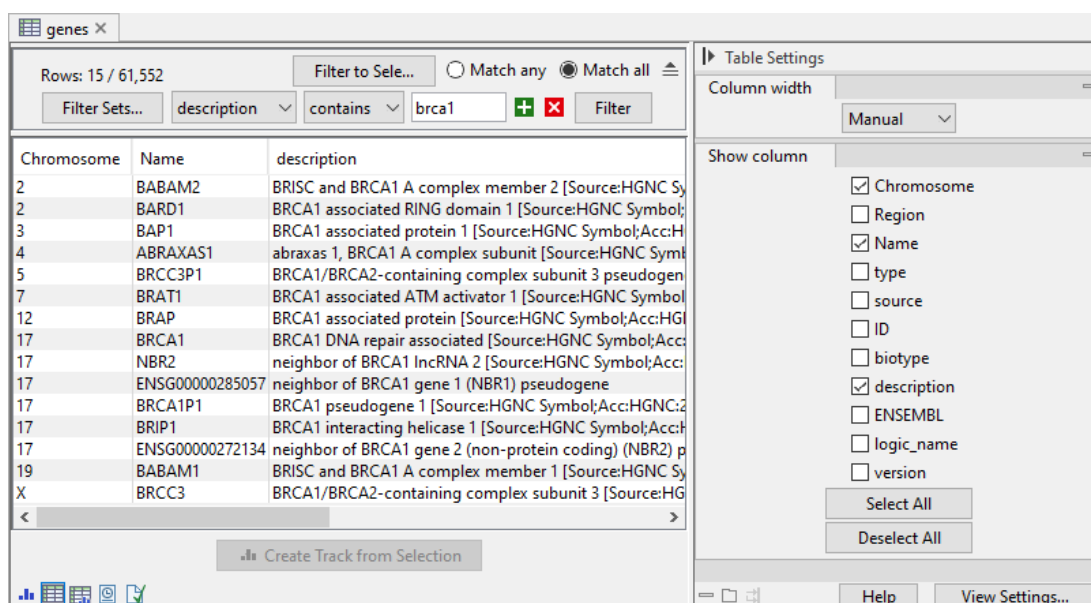


Figure 27.13: The 15 rows where "brca1" was found in the "description" column are visible in the table after filtering.

reads, specified based on location relative to a set of regions and/or based on specified characteristics of the reads. Reads can be extracted to a reads track or sequence list. See section 37.2.

- **Extract Sequences.** Available from the Tools menu. It extracts all reads to a sequence list or individual sequences. See section 18.2.
- **Open Selection in New View.** Available from the right-click menu of the reads in the reads tracks (figure 27.15). The selected read is opened in a separate view.

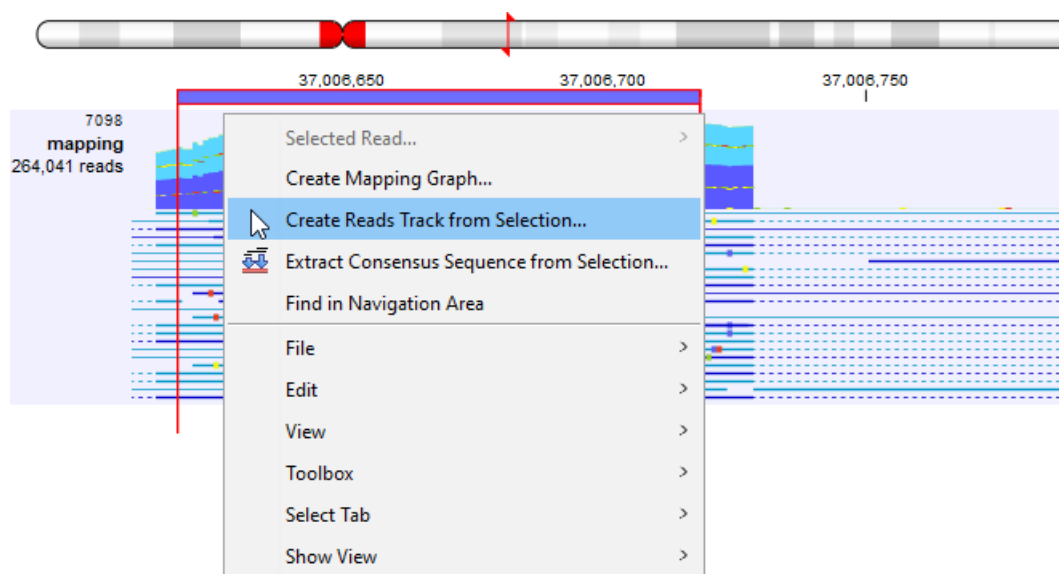


Figure 27.14: Right-click on the selected region in a reads track for revealing the available options.

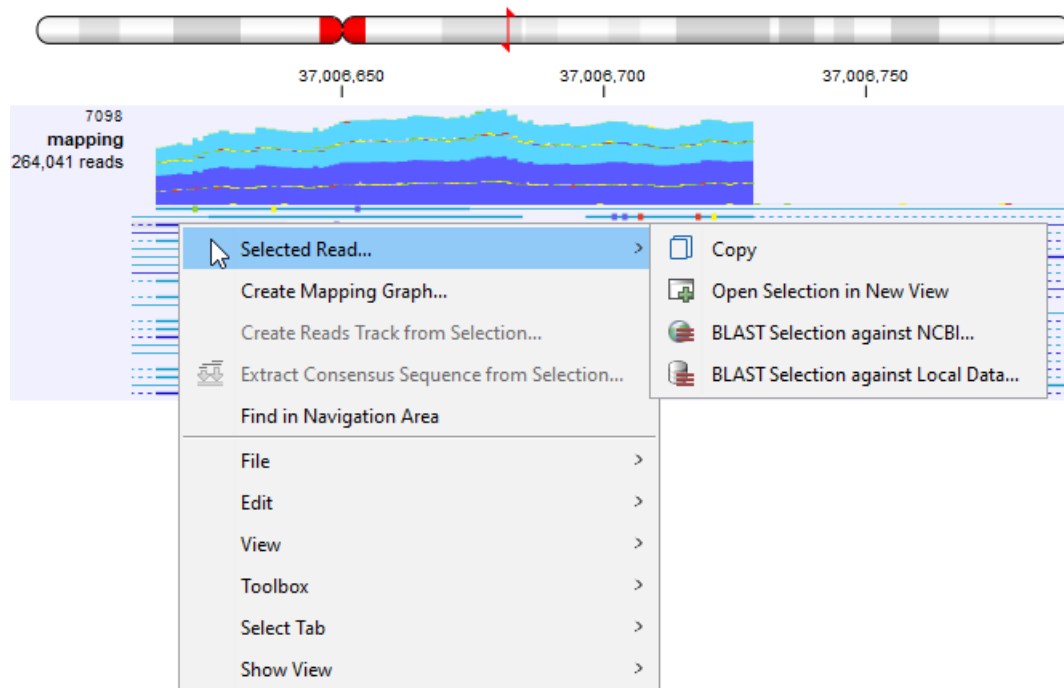


Figure 27.15: Right-click on a read in a reads track for revealing the available options.

Extracting sequences from sequence tracks

Sequences can be extracted from sequence tracks in multiple ways:

- **Extract Sequence.** Available from the right-click menu of the sequence track (figure 27.16). A new sequence element containing the sequence is created. If a region is selected, only the sequence for the selection is created. When the sequence track is part of a Track list that also contains annotation tracks, an option will be available for also extracting the annotations.
- **Extract Sequences.** Available from the Tools menu. It extracts all sequences from the sequence track to a sequence list or individual sequences. See section 18.2.

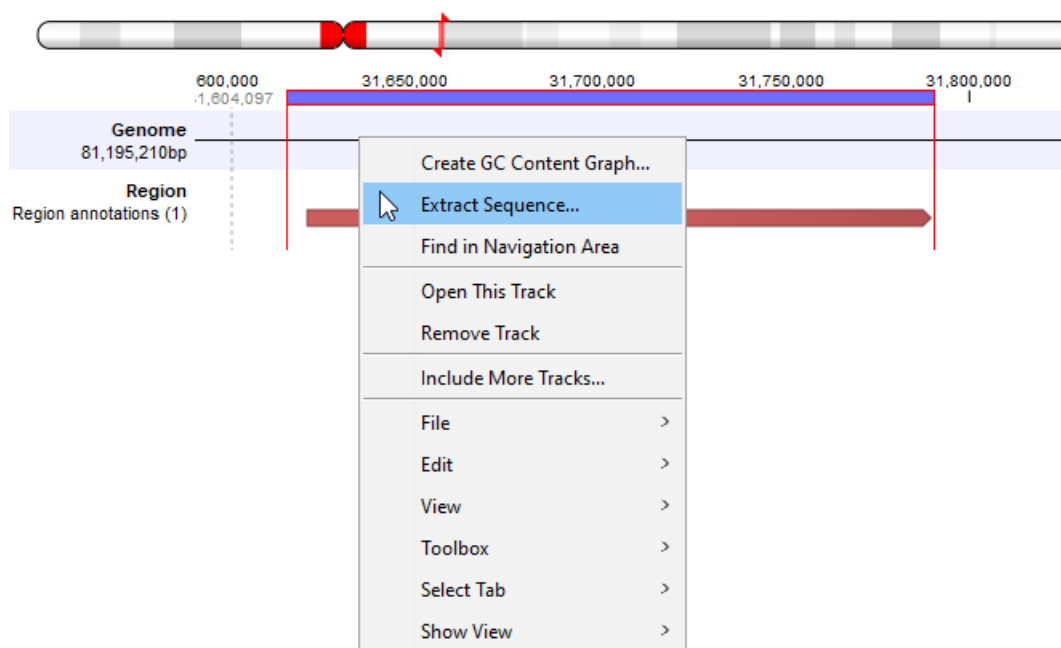


Figure 27.16: Right-click on a sequence from a sequence track for revealing the available options.

27.4 Reference data as tracks

Reference data, such as genome sequences, annotations, known variants, etc, is needed for many types of analyses. When reference data and associated analysis results are in track-based formats, they can be viewed together, allowing investigation and comparison based on shared genomic coordinates (see section 27.2).

Common ways of obtaining reference data in a track format include:

1. **Downloading reference data using the Reference Data Manager** Common genomes, along with annotations, variants, and other available resources, can be downloaded using the Reference Data Manager. See section 11.1 for details about downloading reference data from various public resources, and section 11.2 for details about downloading human, mouse and rat reference data provided by QIAGEN. The latter is particularly relevant when running template workflows provided by plugins and modules.
2. **Importing reference data** Import data in standard formats into the *CLC Genomics Workbench* as tracks using the **Tracks** importer (see section 7.2).
3. **Creating tracks from existing CLC data elements** Create track elements from existing CLC data elements using **Convert to Tracks** (see section 27.8.1).

27.5 Modify tracks

This section describes tools for modifying and annotating tracks. For annotating variant tracks, see also chapter 32.2.

27.5.1 Annotate with Exon Numbers

To run **Annotate with Exon Numbers**, go to:

Tools | **Utility Tools** (🔧) | **Tracks** (📊) | **Modify Tracks** (🔧) | **Annotate with Exon Numbers** (📄➡️)

Given a track with mRNA or CDS annotations, a new track will be created in which variants or other track items are annotated with the numbering of the corresponding exon with numbered exons based on the transcript annotations and gene names in the input track (see an example of a result in figure 27.17).

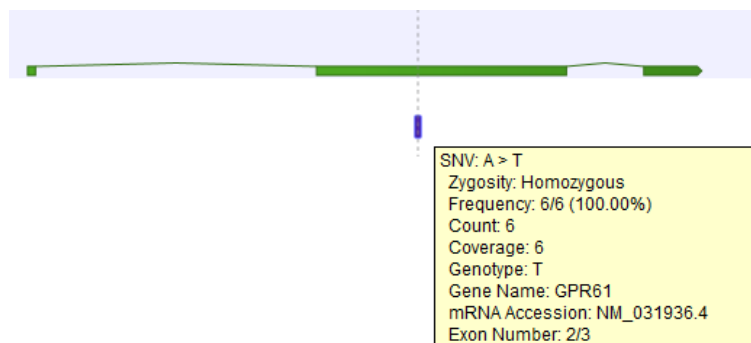


Figure 27.17: A variant found in the second exon out of three in total.

When there are multiple isoforms, a comma-separated list of the exon numbers is given. If an annotation overlaps an intron and has no partial overlap with an exon, this is indicated with a dash. Examples of exon annotations:

- **[3/7]** The annotation overlaps exon 3 in a gene that has 7 exons.
- **[-/4]** The annotation overlaps an intron in a gene that has 4 exons.
- **[-/4, 38/38]** The annotation overlaps an intron in a gene that has 4 exons as well as exon 38 in a gene with 38 exons.
- **[12..17/29]** The annotation overlaps exons 12 to 17 in a gene that has 29 exons.
- **[12..9/18]** The annotation overlaps exons 9 to 12 in a gene that has 18 exons. Exon 12 is written before exon 9, because the gene is located on the reverse strand.

The option **One transcript per gene** chooses a single transcript per gene. So if a variant overlaps multiple transcripts of the same gene, the tool chooses one of these transcripts and only shows overlapping exon numbers for one transcript per gene. The single transcript per gene that is used is chosen based on multiple criteria: First choose the transcript with best priority (lowest priority number). If none of the transcripts has a priority, or multiple transcripts have the same priority, then choose the transcript with the most exons overlapping the input item. Then choose the longest transcript, meaning the one with the highest sum of exon lengths. And finally choose the lexicographically first transcript id.

27.5.2 Annotate with Nearby Information

The tool **Annotate with Nearby Information** creates a copy of the input annotation track (📁) and adds information about nearby annotations present on a separate annotation track.

To start the tool, go to:

Tools | Utility Tools (🔧) | **Tracks** (📁) | **Modify Tracks** (📁) | **Annotate with Nearby Information** (📁)

The tool outputs a track containing four additional columns in the table view compared to the input track:

- **5'/3' annotation** The name of the nearest upstream/downstream annotation, if any.
- **5'/3' distance** The distance to the nearest upstream/downstream annotation or 0 if the input annotation overlaps the nearest upstream/downstream annotation. The distance is given by the shortest distance between the start or end of the annotation and the start or end of the input annotation, regardless of the orientations of the annotations (figure 27.18).

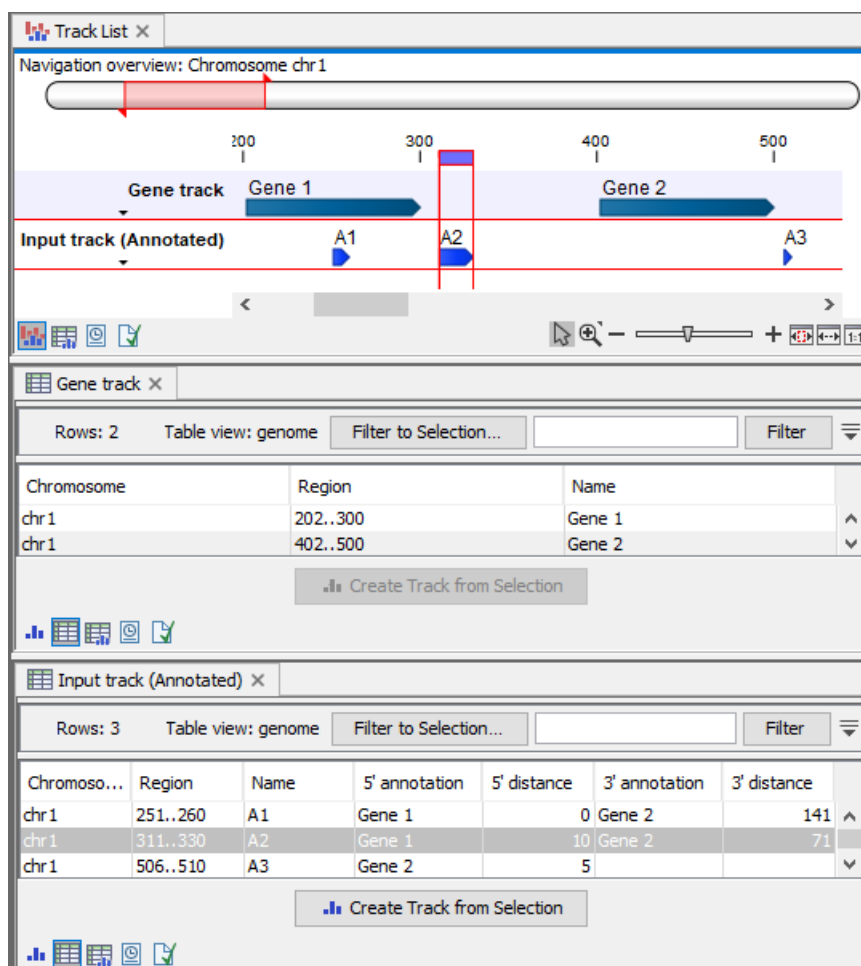


Figure 27.18: Top: Track list containing a gene track that was used to annotate the input track. Middle: Table view of the gene track. Bottom: Table view of the annotated input track.

27.5.3 Annotate with Overlap Information

The tool Annotate with Overlap Information will create a copy of the track used as input and add information from overlapping annotations.

To start the tool, go to:

Tools | Utility Tools (🔧) | **Tracks** (📊) | **Modify Tracks** (📁) | **Annotate with Overlap Information** (🔗)

First, select the track you wish to annotate and click **Next**. You can choose variant tracks, expression tracks or statistical comparison tracks as input. Next, select the annotation track to be used for overlap comparison. The requirement for being registered as an overlap is that parts of the annotations are overlapping, regardless of the strandedness of the annotations: this makes it unsuitable for comparing two gene tracks, but great for annotating variants with overlapping genes or regulatory regions.

It is possible to "Keep only one copy of duplicate annotations" by leaving the option checked as it is by default (see figure 27.19).

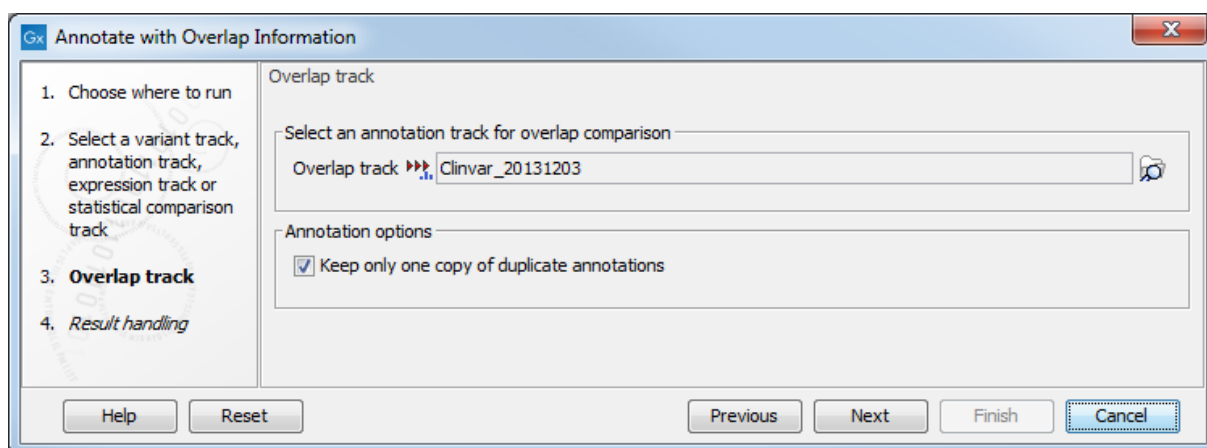


Figure 27.19: Choose an overlap track with which you wish to annotate your input file.

The result of this tool is a new track with all the annotations from the input track and with additional information from the annotations that overlap from the other track. Annotations are visible in the tooltips that appears when hovering the mouse on a variant in the Track view, or as additional columns in the Table view of the track.

27.5.4 Collapse Overlapping Annotations

Collapse Overlapping Annotations takes an annotation track as input and outputs an annotation track where any overlapping annotations are collapsed into a single annotation (figure 27.20).

To run **Collapse Overlapping Annotations**, go to:

Tools | Utility Tools (🔧) | **Tracks** (📊) | **Modify Tracks** (📁) | **Collapse Overlapping Annotations** (🔗)

A collapsed annotation in the output track, resulting from overlapping input annotations (figure 27.20):

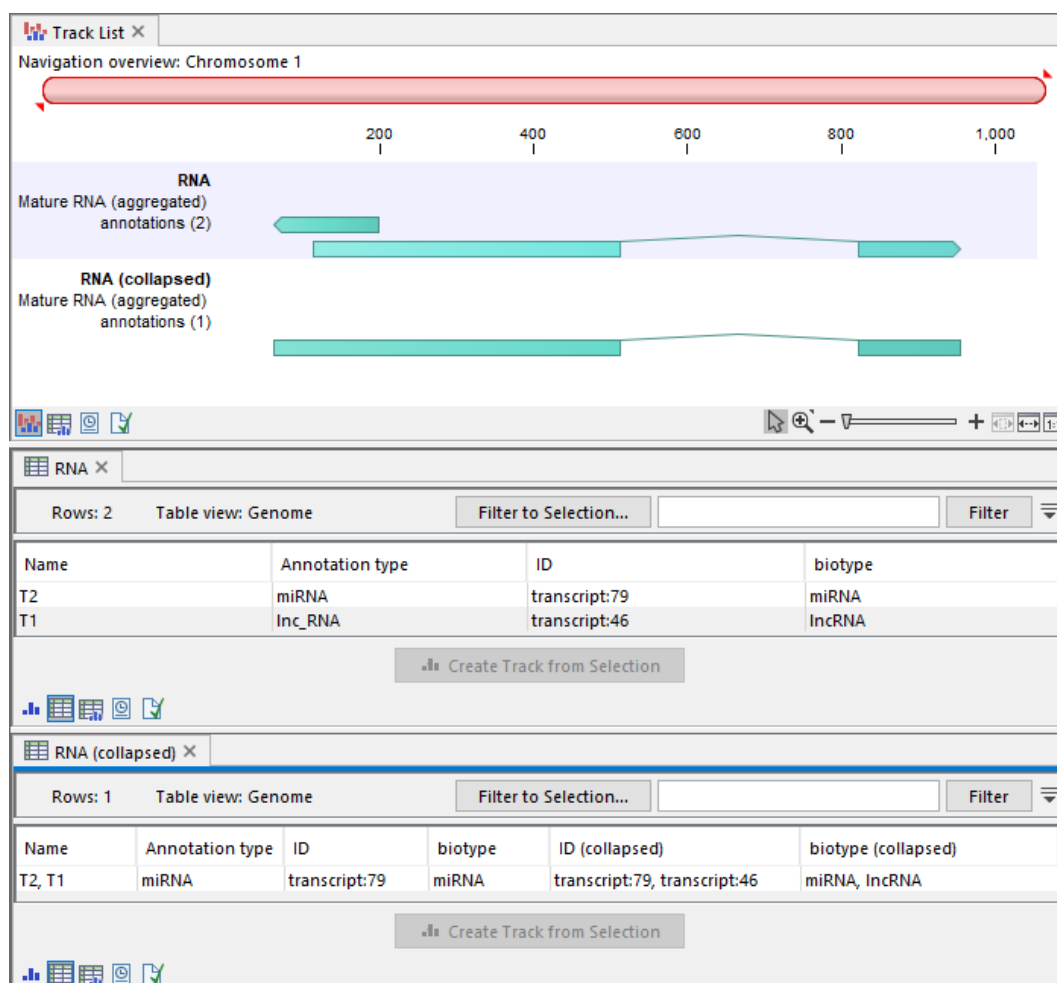


Figure 27.20: A split view displaying a track list at the top with two annotation tracks and their corresponding table views below. The track list includes an input track with two overlapping RNA annotations, and the output track from Collapse Overlapping Annotations where the two annotations are collapsed.

- Has the name of the input annotations, separated by commas.
- Has the same strand as the input annotations if they are all on the same strand, and has no strand if the input annotations are on different strands.
- Has the same annotation type and attributes as the left-most input annotation.
- Has additional collapsed attributes with all values from the input annotations, separated by commas.

Unwanted attributes can be removed using **Remove Information from Track**, see section 27.5.5.

If annotations with different annotation types or strands should not be collapsed, create separate annotation tracks containing annotations with the same type/strand. In this case:

- Open the table view (📊) of the annotation track.
- Filter to the desired annotation type/strand. See chapter 9.2 for details.


- Select the filtered rows (shortcut Ctrl + A, or ⌘ + A on Mac).
- Create a new track using the **Create Track from Selection** button.
- Repeat for each annotation type/strand.

The filtered tracks can be collapsed separately and then merged using **Merge Annotation Tracks**, see section 27.7.1.





27.5.5 Remove Information from Track

Remove Information from Track takes a track as input and outputs another track containing only the selected attributes.



The following track types are accepted as input:


- Variant track ()
- Annotation track ()
- Expression track ()
- Statistical comparison track ()

To run the tool, go to:

Tools | **Utility Tools** () | **Tracks** () | **Modify Tracks** () | **Remove Information from Track** ()

The attributes can be entered by typing directly in the text field (figure 27.21). Each attribute should be on a separate line.

To make attribute selection easier, the **Load Attributes** button can be used to populate the attributes available in the element selected in the **Template element** field (figure 27.22). The arrows () and () move attributes between the **Available** and **Selected** lists.

By default, the tool's input is preselected. Use the browse () button to select a different element. We recommend selecting as template a small element containing all necessary attributes, as loading attributes from large elements can take a long time.

Below the attributes, there is an option to choose whether the selected attributes should be kept (**Keep**) or removed (**Remove**).

27.5.6 Resize Annotations

Resize Annotations takes an annotation track as input and outputs an annotation track with adjusted annotation lengths.

To run the tool, go to:

Tools | **Utility Tools** () | **Tracks** () | **Modify Tracks** () | **Resize Annotations** ()

The following options can be configured (figure 27.23):

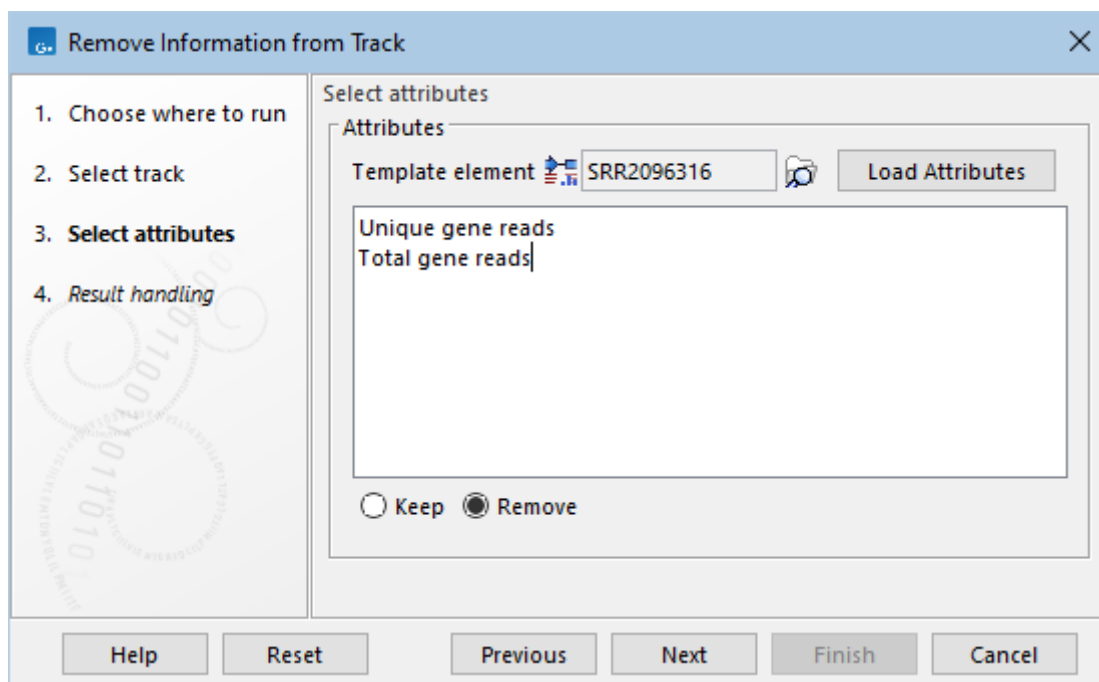


Figure 27.21: Attributes can be entered by typing directly in the text field.

- **Lengthen at 5' end** Extend the size of the annotations at the 5' end.
- **Shorten at 5' end** Reduce the size of the annotations at the 5' end.
- **Change 5' position by (bp)** The number of basepairs by which the annotations should be lengthened or shortened.
- **Lengthen at 3' end** Extend the size of the annotations at the 3' end.
- **Shorten at 3' end** Reduce the size of the annotations at the 3' end.
- **Change 3' position by (bp)** The number of basepairs by which the annotations should be lengthened or shortened.

The 5' and 3' ends are relative to the strand of the annotation. For annotations on the positive strand, the 5' end is the first position and 3' end is the last position, and vice versa for annotations on the negative strand. Annotations with no strand are treated as being on the positive strand.

An annotation corresponds to a genomic region, which can consist of a single interval, such as a gene, or multiple intervals, such as the exons of an mRNA, where each exon is depicted as a separate interval (see "mRNA" at the top of figure 27.24).

If an annotation consists of a single interval and is shortened by at least the length of that interval, it will be removed from the track (see "mRNA (shorter)" in the middle of figure 27.24).

If an annotation consists of multiple intervals, each interval is resized separately. Intervals will be removed from the annotation when the size is reduced by at least the length of the interval (see "mRNA (shorter)" in the middle of figure 27.24), and two or more intervals will be merged into a

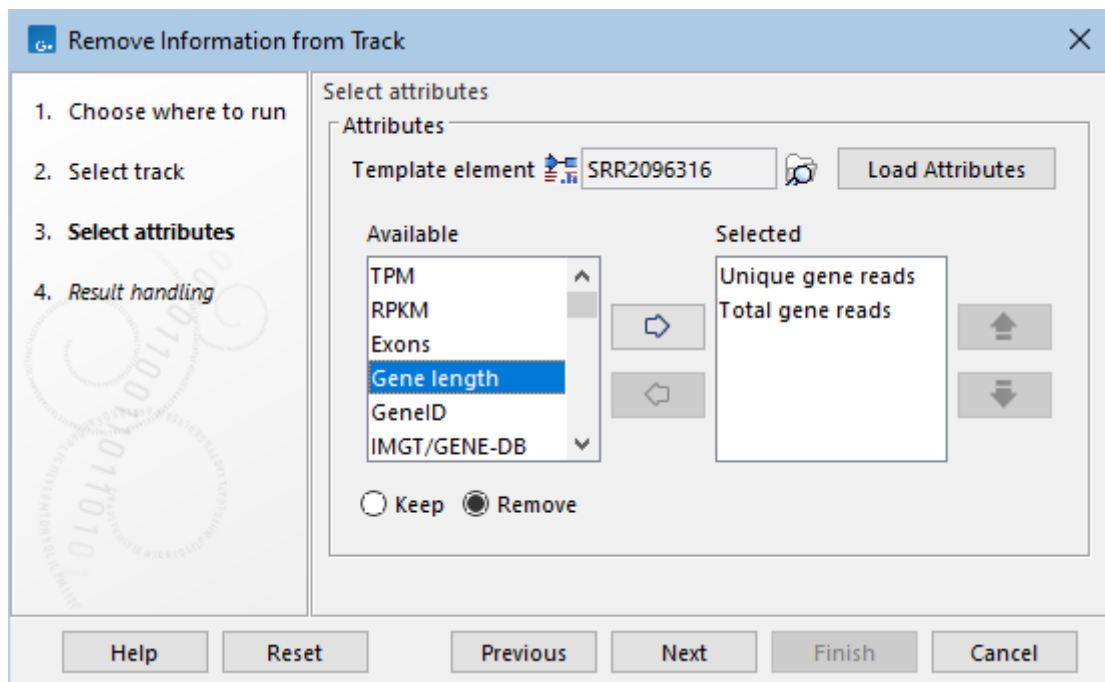


Figure 27.22: After loading attributes from a template element, attributes can be moved between the available and selected lists using the arrows.

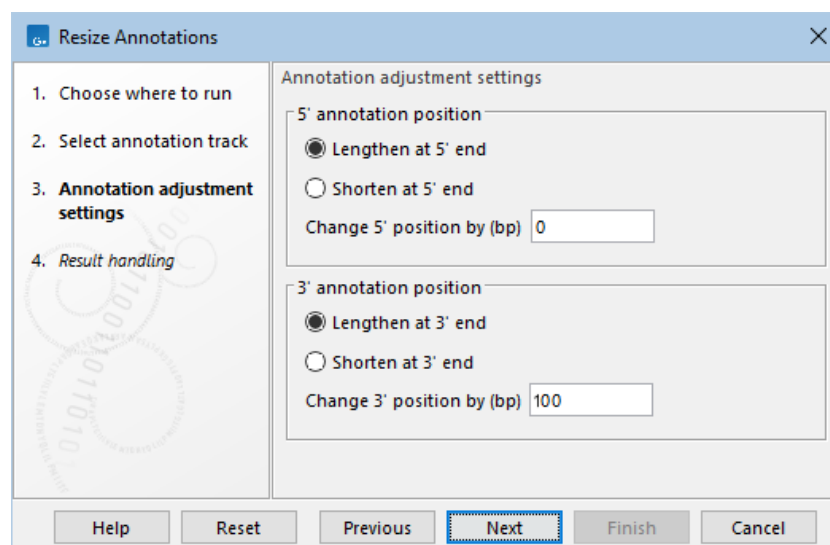


Figure 27.23: Extend all annotations by 100 bp at the 3' end.

single interval if the length is increased to the point where they overlap (see "mRNA (longer)" at the bottom of figure 27.24).

Extending annotations is limited to the chromosome boundaries for non-circular chromosomes. For circular chromosomes, annotations can be extended to wrap around the start or end of the chromosome.

Resized annotations that overlap (see "mRNA (longer)" at the bottom of figure 27.24) can be merged using **Collapse Overlapping Annotations**, see section 27.5.4.

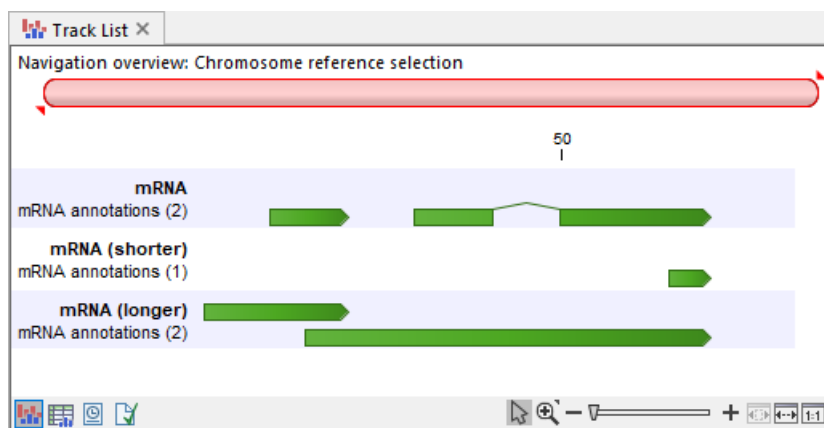


Figure 27.24: A track list with three tracks. Top: an input track with two mRNA annotations - one with a single interval, the other with two intervals. Middle and bottom: the output tracks from *Resize Annotations*, where annotations were shortened and lengthened at the 5' end, respectively.

27.6 Filter tracks

This section describes tools for filtering tracks of different types. For filtering variants, see also chapter 32.1.

27.6.1 Filter Annotations on Name

The name filter allows you to use a list of names as input to create a new track only with these names. This is useful if you wish to filter your variants so that only those within certain genes are reported.

A flow of work could be to first create a new gene track containing only the genes of interest using this tool. Then use the **Filter Based on Overlap** tool (see section 27.6.2) to filter the variants based on the track containing the genes of interest.

Tools | Utility Tools (📁) | **Tracks** (📁) | **Filter Tracks** (📁) | **Filter Annotations on Name** (🔍)

Select the track you wish to filter and click **Next**.

As shown in figure 27.25, you can specify a list of annotation names. Each name should be on a separate line.

In the bottom part of the wizard you can choose whether you wish to keep the annotations that are found, or whether you wish to exclude them. In the use case described above a track was created with only those annotations being kept that matched the specified names. Sometimes the other option may be useful, for example if you wish to screen certain categories of genes from the analysis (for example excluding all cancer genes).

27.6.2 Filter Based on Overlap

Filter Based on Overlap takes as input a track that contains region-specific information, hereby referred to as annotations. The following track types are supported:

- Variant track (📊)

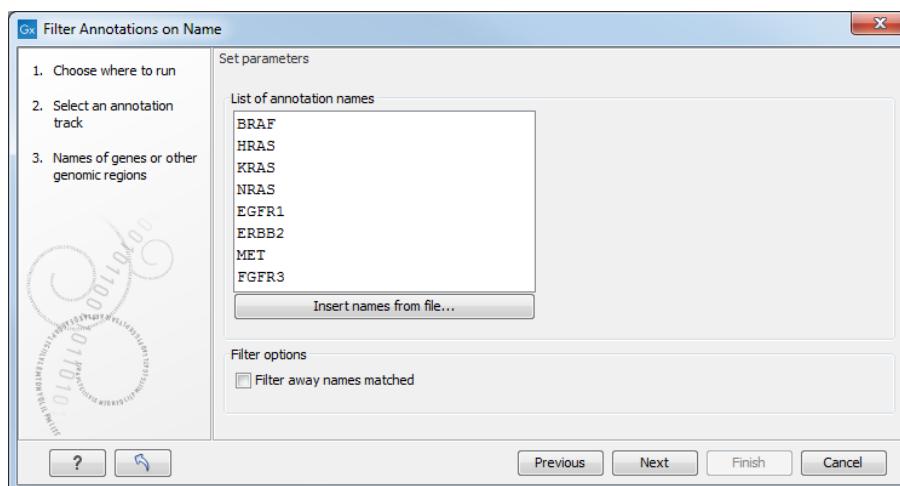


Figure 27.25: Specify names for filtering.

- Annotation track (📊)
- Expression track (📈)
- Statistical comparison track (📊)

The output is a track of the same type as the input, where the annotations are filtered based on overlaps with the annotations found in one or more other tracks.

The tool can be used to e.g. restrict to variants that fall within genes or regulatory regions.

To run the tool, go to:

Tools | **Utility Tools** (🔧) | **Tracks** (📊) | **Filter Tracks** (📊) | **Filter Based on Overlap** (📊)

The following options can be configured (figure 27.26):

- **Track(s)** At least one track, used for calculating the overlaps.
- **Overlapping annotations options** The content of the output track (figure 27.27). Hover the mouse cursor over the option to see an example on input and output.
 - **Keep overlapping annotations** The annotations in the input track that overlap with any annotations in the other track(s).
 - **Keep overlapping annotation parts** The annotations in the input track that overlap with any annotations in the other track(s), shortened to the region of the overlap. This option can only be used for annotation tracks.
 - **Remove overlapping annotations** The annotations in the input track that do not overlap with any annotations in the other track(s).
 - **Remove overlapping annotation parts** The annotations in the input track that do not overlap with any annotations in the other track(s), shortened to the region of no overlap. This option can only be used for annotation tracks.

Two annotations are considered to be overlapping if they share at least one position.

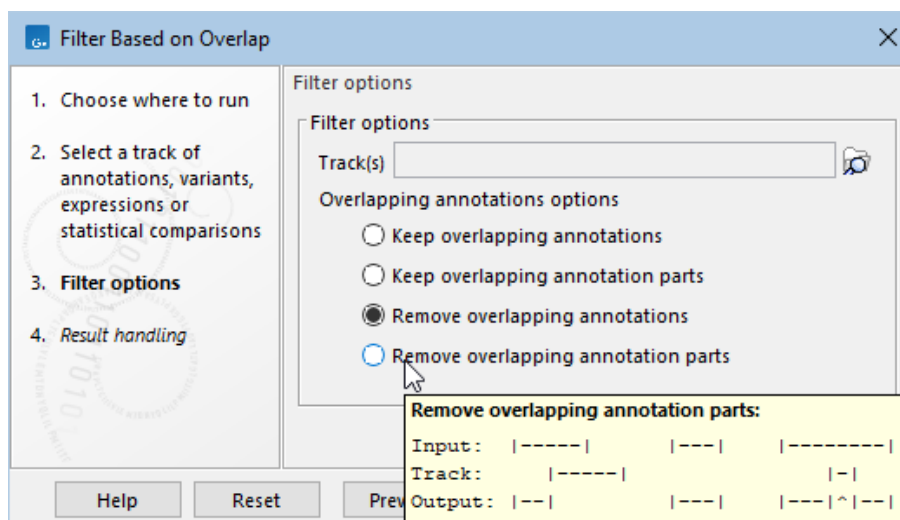


Figure 27.26: The available options in *Filter Based on Overlap*. Hovering the mouse cursor over an overlapping annotation option reveals a tooltip with an example of input and output.

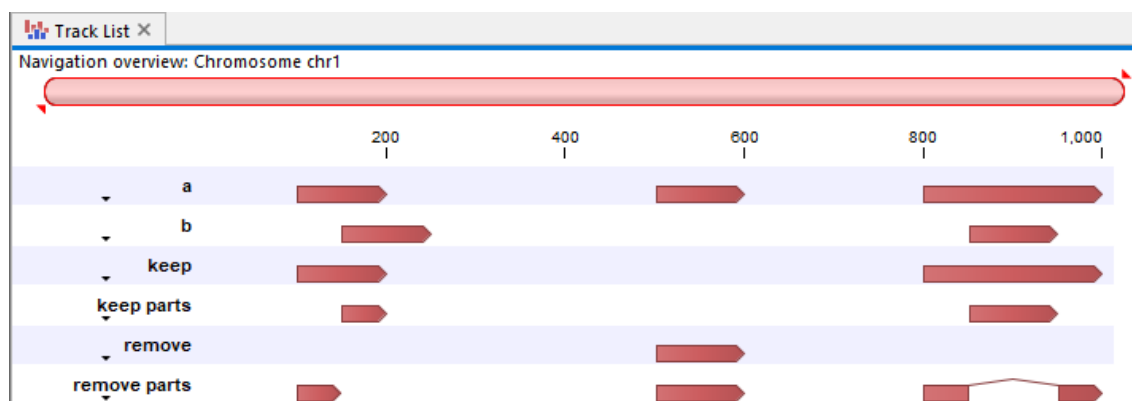


Figure 27.27: A track list with an input annotation track *a* and another annotation track *b*, as well as the output tracks from *Filter Based on Overlap*, using each overlapping annotations option. "Keep overlapping annotations": keep. "Keep overlapping annotation parts": keep parts. "Remove overlapping annotations": remove. "Remove overlapping annotation parts": remove parts.

27.7 Merge tracks

This section describes tools for merging annotation and variant tracks.

27.7.1 Merge Annotation Tracks

Merge Annotation Tracks merges annotations from two or more annotation tracks into a single annotation track. Only tracks based on compatible genomes can be merged. Merging annotation tracks can be particularly useful with gene tracks from different sources that use different naming conventions.

This tool is **not** intended for comparison of variant tracks. That is described in section [32.3](#).

To run the tool, go to:

Tools | **Utility Tools**  | **Tracks**  | **Merge Tracks**  | **Merge Annotation Tracks** 

Duplicate annotation handling

Annotations of the same type, with the same start and end coordinates, and the same number of intervals, are considered duplicate annotations.

Duplicate annotations are merged to a single annotation in the output track. The merged annotation's name is based on the annotation name in the top-most track selected in the wizard (that contained that annotation). Information for most other columns, including the region and strand information, is also populated using the information from that input track.

Columns named "Origin names" and "Origin tracks" are added to the merged annotation track. These contain the names of the annotations and the names of the tracks that contributed to the merged annotation.

27.7.2 Merge Variant Tracks

Merge Variant Tracks merges variants from two or more variant tracks into a single variant track. Only tracks based on compatible genomes can be merged.

To run the tool, go to:

Tools | **Utility Tools**  | **Tracks**  | **Merge Tracks**  | **Merge Variant Tracks** 

Select the variant tracks to merge.

In the next wizard step, the following options can be configured (figure 27.28:

- **Don't merge variants** Duplicated variants are not merged and all variants from the inputs will be included
- **Merge duplicated variants** Variants defining the same mutation are merged.
- **Merge duplicated variants (zygosity-aware)** If the duplicated variant is homozygous, all variants from all other inputs with identical regions will be discarded.
- **Annotate variants** A column called **Origin tracks** is added. The name of the input track the variant came from is recorded in it. Note that standard variant annotations are retained, whether or not this option is selected.

Extra columns are created in the output track to contain the annotations of any duplicates of a variant found. The names of these extra columns include the name of the type of information contained followed by the originating track name. Such columns are made for all but the first of the input tracks. The names of all the input tracks where that variant was found are entered into the **Origin tracks** column.

Please also see section 32.3 for information about tools designed to support variant comparison.

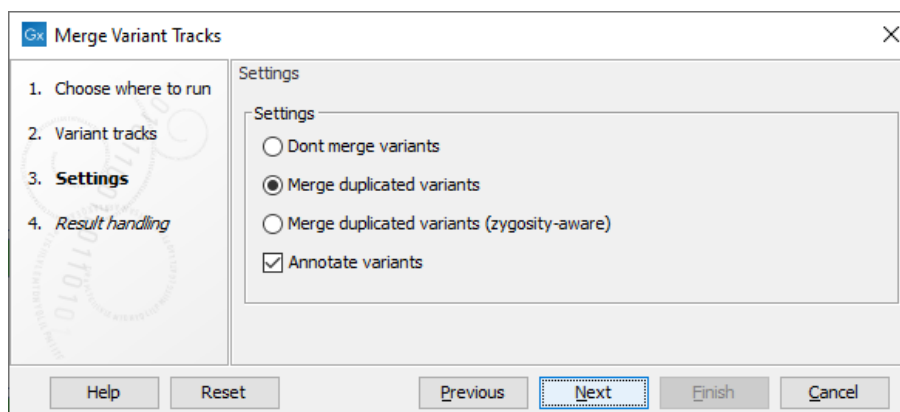


Figure 27.28: Merge Variant Tracks settings.

27.8 Convert tracks

The *CLC Genomics Workbench* provides tools for converting data to tracks, for extracting sequences and annotations from tracks, and for creating standard annotated sequences and mappings.

27.8.1 Convert to Tracks

When working with tracks, information from standard sequences and mappings are split into specialized tracks with sequence, annotations and reads. This tool creates a number of tracks based on the input sequences:

Tools | Utility Tools (📁) | **Tracks** (📊) | **Convert Tracks** (📁) | **Convert to Tracks** (📁)

The following kinds of data can be converted to tracks: nucleotide sequences (📄), sequence lists (📄) and read mappings (📄)/ (📄). Select the input and click **Next** to specify which tracks should be created (see figure 27.29).

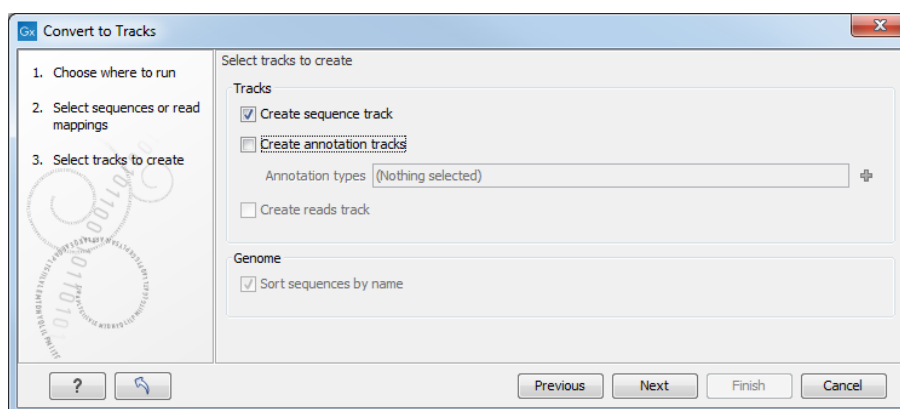


Figure 27.29: Converting data to tracks.

For sequences and sequence lists, you can **Create a sequence track** (for mappings, this will be the reference sequence) and a number of **Annotation tracks**. For each annotation type selected, a track will be created. For mappings, a **Reads track** can be created as well.

At the bottom of the dialog, there is an option to sort sequences by name. This is useful for

example to order chromosomes in the menus etc (chr1, chr2, etc). Alphanumerical sorting is used to ensure that the part of the name consisting of numbers is sorted numerically (to avoid e.g. chr10 getting in front of chr2). When working with de novo assemblies with huge numbers of contigs, this option will require additional memory and computation time.

27.8.2 Convert from Tracks

Tracks are useful for comparative analysis and visualization, but sometimes it is necessary to convert a track to a normal sequence or mapping. This can be done with the **Convert from Tracks** tool that can be found here:

Tools | **Utility Tools**  | **Tracks**  | **Convert Tracks**  | **Convert from Tracks** 

One or more tracks can be used as input. In the example given in figure 27.30 a reads track and two annotation tracks are converted simultaneously to an annotated read mapping (figure 27.31).

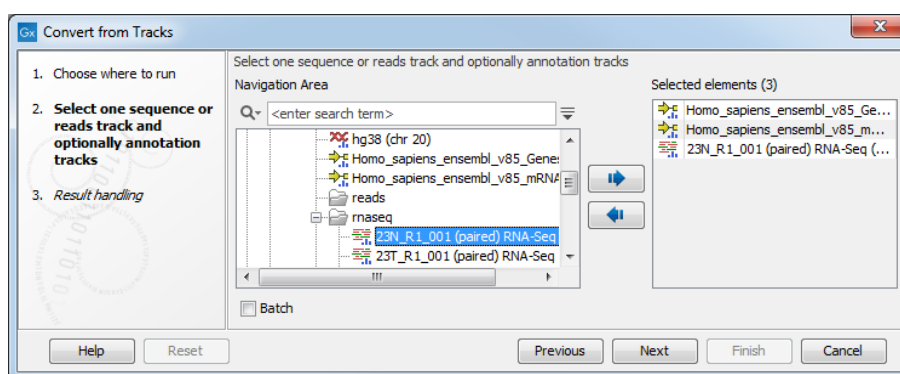






Figure 27.30: A reads track and two annotation tracks are converted from track format to stand-alone format.



Likewise it is possible to create an annotated, stand-alone reference from a reference track and the desired number of annotation tracks. This is shown in figure 27.32 where one reference and two annotation tracks are used as input.

The output is shown in figure 27.33. The reference sequence has been transformed to stand-alone format with the two annotations "CDS" and "Gene".

Depending on the input provided, the tool will create one of the following types of output:

Sequence  Will be created when a sequence track  with a genome with only one sequence (one chromosome) is provided as input

Sequence list  Will be created when a sequence track  with a genome with several sequences (several chromosomes) is provided as input

Mapping  Will be created when a reads track  with a genome with only one sequence (one chromosome) is provided as input.



Mapping table  Will be created when a reads track  with a genome with several sequences (several chromosomes) is provided as input.



Figure 27.31: The upper part of the figure shows the three individual input tracks, arranged for simplicity in a track list. The lower part of the figure shows the resulting stand-alone annotated read mapping.

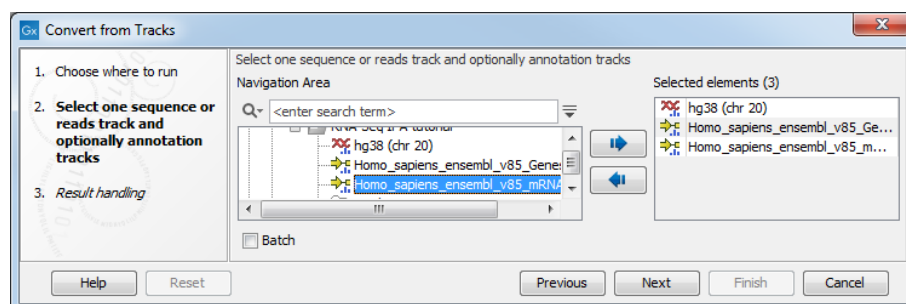


Figure 27.32: A reference track and two annotation tracks are converted from track format to stand-alone format.



In all cases, any number of annotation tracks (/ ) can be provided, and the annotations will be added to the sequences (reference sequence for mappings) as shown in figure 27.31.



Figure 27.33: The upper part of the figure shows the three input tracks, shown for simplicity in a track list. The lower part of the figure shows the resulting stand-alone annotated reference sequence.

27.9 Graph tracks

Graph tracks contain numerical information pertaining to sequence tracks, (see section 27.9.1) or read mapping tracks (aka "reads tracks") , (see section 27.9.2).

Regions in graph tracks that meet specified thresholds can be identified using the **Identify Graph Threshold Areas**, described in section 27.9.3.

27.9.1 Create GC Content Graph

Create GC Content Graph takes a sequence track as input and creates a graph track of the GC content of that sequence.

To run the tool, go to:

Tools | Utility Tools | **Tracks** | **Graph Tracks** | **Create GC Content Graph**

After selecting a sequence track as input (figure 27.34) the window size is specified (figure 27.35). This is the size of the region over which the GC content should be calculated. For example, with a window size of 25, the GC content would be calculated for a region comprised of a central base plus the 12 bases upstream and the 12 bases downstream of it. The value calculated for a given window is assigned to the central base of that window. As there is a single central base, the window size must be an odd number.

The output is a graph track containing the GC content values calculated for each base. When zoomed fully in, hovering the mouse cursor over a position reveals a tooltip with the GC content for that position. This information is also displayed in the bottom right corner of the Workbench frame (figure 27.36).

When zoomed out fully, three overlapping plots in blue are visible (figure 27.37). These display

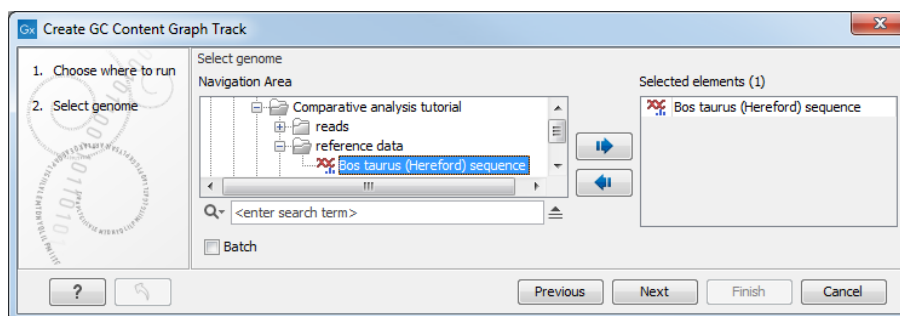


Figure 27.34: Select a sequence track as input to Create GC Content Graph.

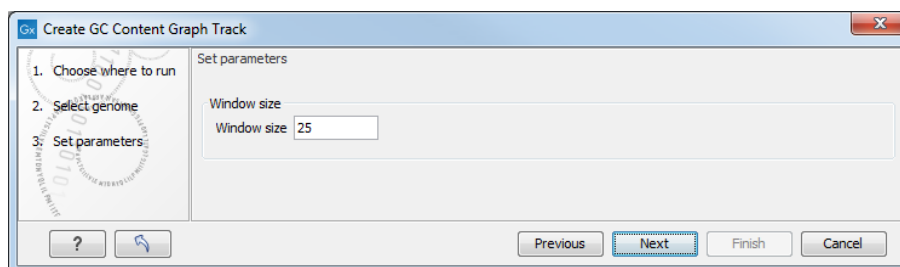


Figure 27.35: Specify the window size. The window size is the region around each individual base that should be used to calculate the GC content in a given region.

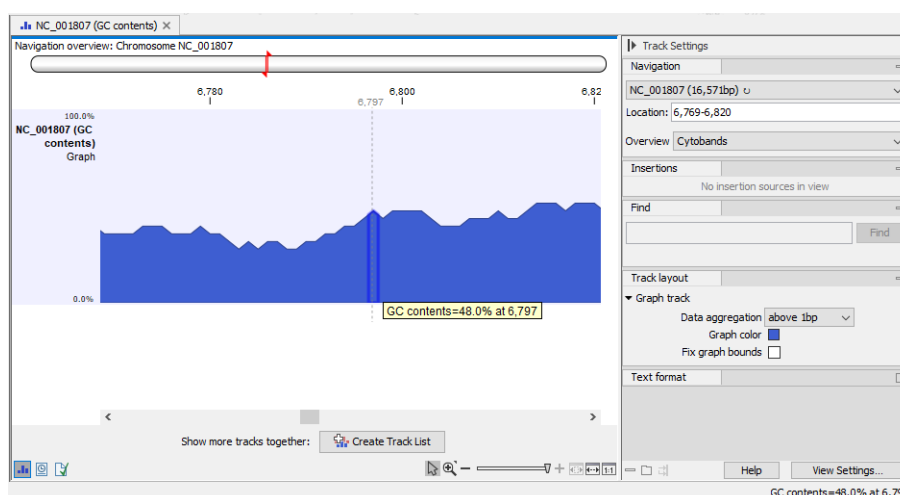


Figure 27.36: A GC content graph track, zoomed in to the single nucleotide level. Hovering the mouse cursor over the track reveals information about the GC content in the window centered on that position.

a summary of the GC content values along the sequence. The highest of the three (darkest shade) shows the maximum values, the middle one (medium shade) shows the mean values, and the bottom one (lightest shade) shows the minimum values. Hovering the mouse cursor over a position reveals a tooltip showing information about the range of values and the size of region those values pertain to. This information is also displayed in the bottom right corner of the Workbench frame.

The y-axis value range to display can be specified in the **Track layout** palette of the **Side Panel**.

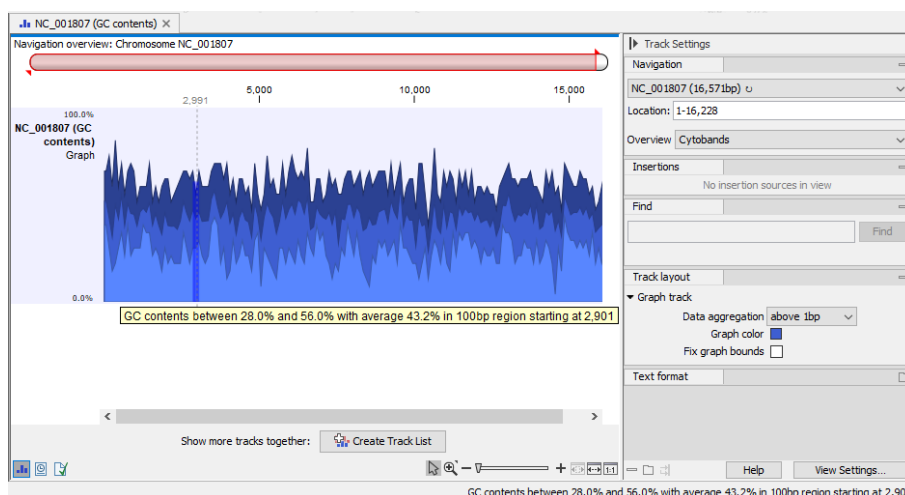


Figure 27.37: The GC content graph track when zoomed out fully contains three plots. These show, respectively, the highest, mean and lowest GC content values along the sequence. Hovering the mouse cursor over the track reveals information about the values in that region.

By default, the range displayed is 0 to 100%. Check the "Fix graph bounds" to reveal fields where the value range can be entered. In the context of a Track List, this setting affects the y-axis values for all graph tracks in the list.

27.9.2 Create Mapping Graph

Create Mapping Graph creates graph tracks containing coverage or distance values derived from a read mapping.

To run the tool, go to:

Tools | Utility Tools () | Tracks () | Graph Tracks () | Create Mapping Graph ()

After specifying the read mapping track (aka "reads track") to use as input (figure 27.38), the values to generate graph tracks for are specified (figure 27.39). One graph track is created for each type selected.

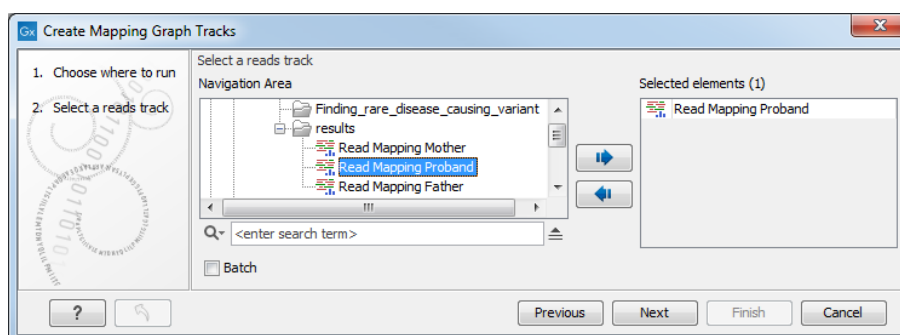


Figure 27.38: A read mapping track is selected as input to Create Mapping Graph.

The value types are:

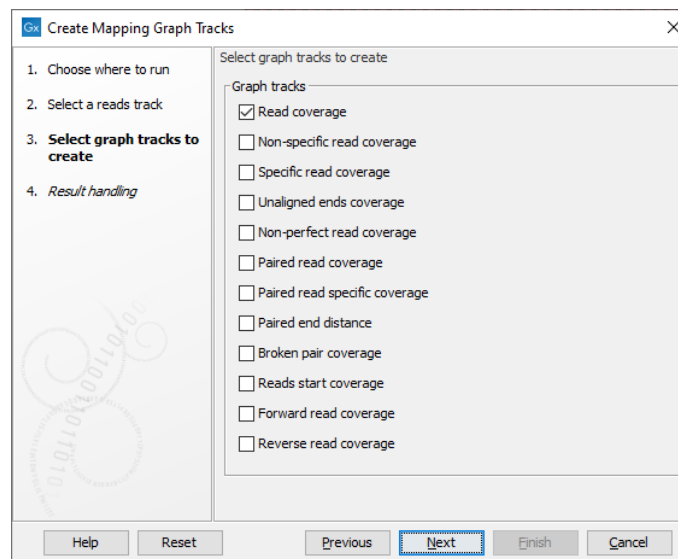


Figure 27.39: Choose the types of graph tracks to be created.

- **Read coverage** The number of reads contributing to the mapping at each position. See section 29.3.1 for further details.
- **Non-specific read coverage** The number of reads mapped at the position that would map equally well to other places in the reference sequence.
- **Specific read coverage** The number of reads that map uniquely at the position, i.e. they do not map equally well to other places in the reference sequence.
- **Unaligned ends coverage** The number of reads with unaligned ends at the position. Unaligned ends arise when a read has been locally aligned and there are mismatches or gaps relative to the reference sequence at the end of the read. Unaligned regions do not contribute to coverage in other graph track types.
- **Non-perfect read coverage** The number of reads at the position with one or more mismatches or gaps relative to the reference sequence.
- **Paired read coverage** The number of intact read pairs mapped to the position. Coverage is counted as one in positions where the reads of a pair overlap.
- **Paired read specific coverage** The number of intact paired reads that map uniquely at the position, i.e. they do not map equally well to other places in the reference sequence.
- **Paired end distance** The average distance between the forward and reverse reads of pairs mapped to the position.
- **Broken pair coverage** The number of broken paired reads mapped to the position. A pair is marked as broken if only one read in the pair matches the reference, if the reads map to different chromosomes, or if the distance or relative orientation between the reads is not within the expected values.
- **Reads start coverage** The number of reads with their start mapped to the position.

- **Forward read coverage** The number of reads mapping in forward direction. First and second read of a pair will be counted separately.
- **Reverse read coverage** The number of reads mapping in reverse direction. First and second read of a pair will be counted separately.

A Track List with three graph tracks containing different types of information from a single mapping is shown in figure 27.40. The view in that figure is zoomed out fully, so three overlapping plots are visible in each track. The highest of the three plots (darkest shade) shows the maximum values, the middle one (medium shade) shows the mean values, and the bottom one (lightest shade) shows the minimum values observed along the mapping.

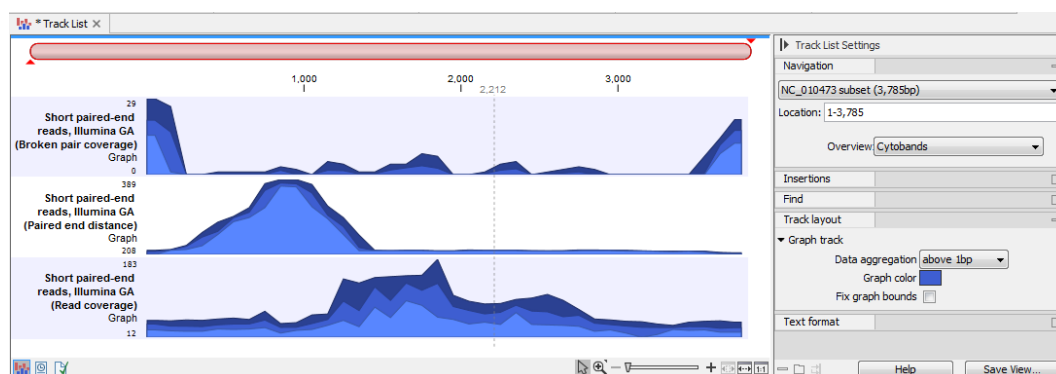


Figure 27.40: A Track List with three graph tracks containing different values from data in a single read mapping: broken pair coverage (top), paired end distance (middle) and read coverage (bottom). The view is fully zoomed out, so each track shows 3 overlapping plots. These show respectively, the highest, mean and lowest values along the mapping.

Hovering the mouse over a position reveals a tooltip showing the value at that position (figure 27.41).

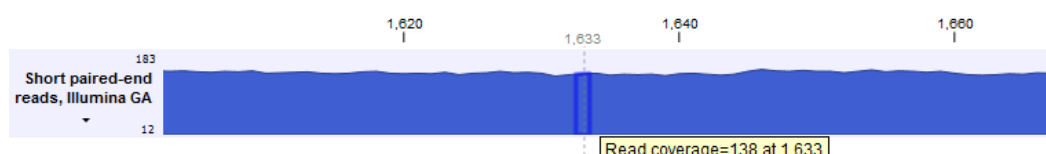


Figure 27.41: A graph track zoomed in to the single nucleotide level. The tooltip contains the value in that track at the position the mouse cursor is hovered over.

The y-axis values to display can be specified in the **Track layout** palette of the **Side Panel**. Check the "Fix graph bounds" to reveal fields where the value range can be entered. By default, the range displayed is 0 to the maximum value within that track. In the context of a Track List, this setting affects the y-axis values for all graph tracks in the list.

27.9.3 Identify Graph Threshold Areas

Identify Graph Threshold Areas identifies regions in a graph track that fall within specified thresholds. The output is a new track containing an annotation for each region that met the specified criteria (figure 27.42).

To run the tool go to:

Tools | Utility Tools (📁) | Tracks (📁) | Graph Tracks (📁) | Identify Graph Threshold Areas (📁)

After providing a graph track as input, a lower threshold and an upper threshold can be specified, along with the window size to use.

The window size defines the size of the region over which an average value for each position should be calculated. It is these calculated values that are compared with the threshold(s) set. For example, with a window size of 100, each position is assigned the average of the values for the surrounding 100 positions.

A window size of 1 results in the original value for every position being used for the comparison. However, a very small window size can lead to many small, fragmented regions being annotated. Larger window sizes can provide output that is more readily interpreted, and can help detect regions that may only be noticed when looking at the larger context with lower resolution. See figure 27.43.

Detection can be restricted to include or exclude certain regions. For this, a track with these regions must be provided.

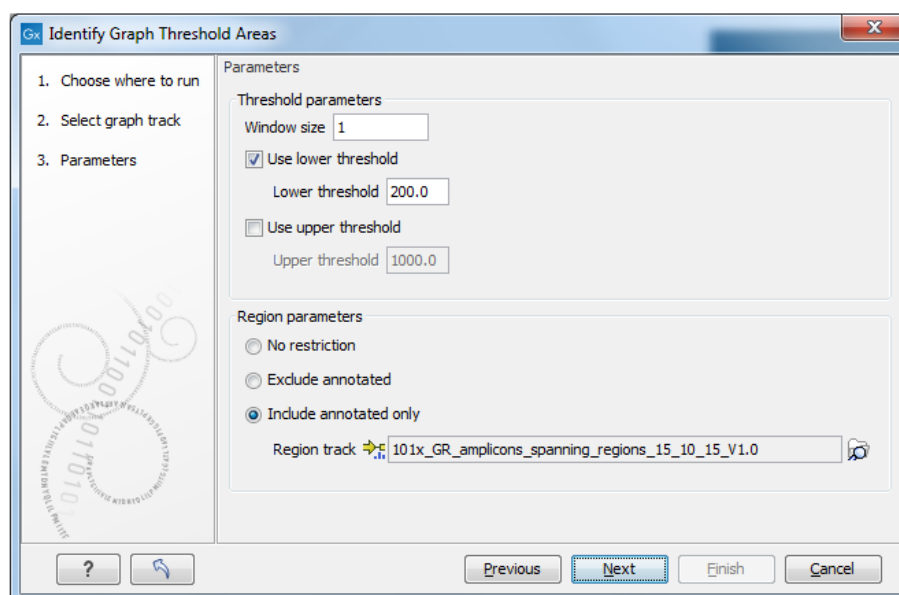


Figure 27.42: Thresholds to meet and the window size to use, are specified when launching the tool. Optionally, the analysis can be restricted to include or exclude particular regions.

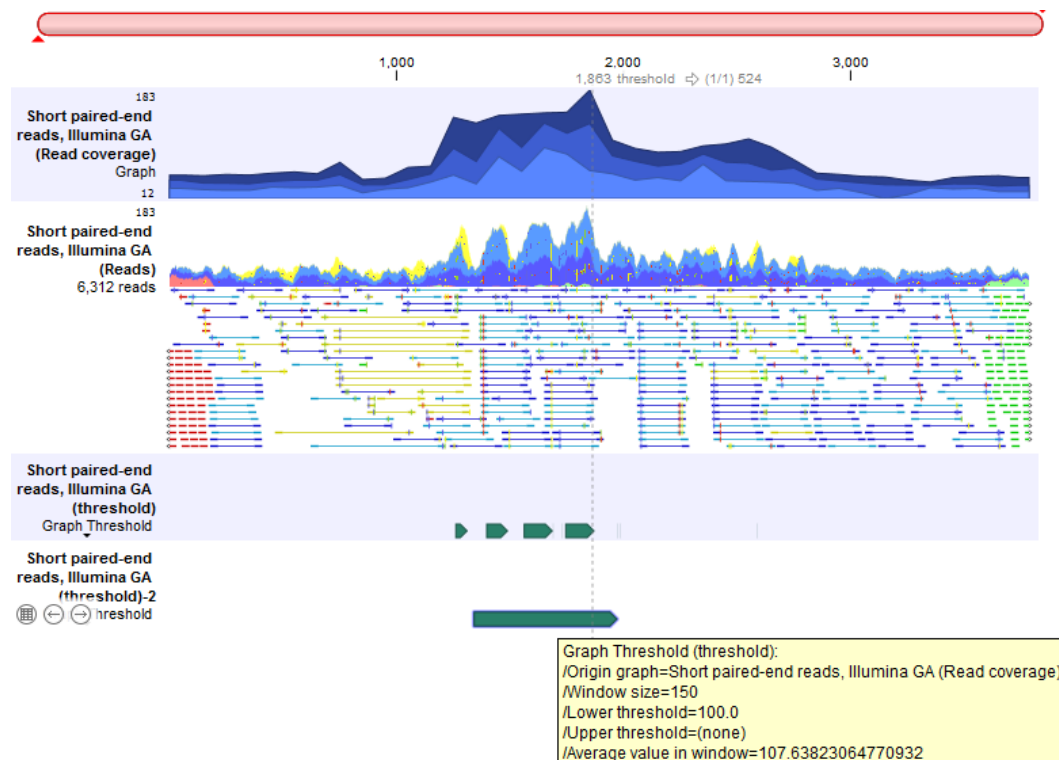


Figure 27.43: A Track List containing a read coverage graph, a reads track, and two threshold graphs containing annotations of regions where coverage in the mapping graph track was above 100. The upper threshold track was generated with a window size of 1 and contains several small annotations, revealing local minima in the coverage. The lower threshold track was generated using a window size of 150, resulting in a single, unbroken annotation covering the high coverage region.

Chapter 28

Prepare sequencing data

Contents

28.1 QC for Sequencing Reads	738
28.1.1 Per-sequence analysis	740
28.1.2 Per-base analysis	741
28.1.3 Over-representation analyses	741
28.2 Trim Reads	743
28.2.1 Quality trimming	743
28.2.2 Adapter trimming	744
28.2.3 Trim adapter list	746
28.2.4 Homopolymer trimming	753
28.2.5 Sequence trimming	754
28.2.6 Sequence filtering	754
28.2.7 Trim output	755
28.3 Demultiplex Reads	758
28.3.1 Running Demultiplex Reads	758
28.3.2 Output from Demultiplex Reads	762
28.3.3 Running Demultiplex Reads in workflows	762

28.1 QC for Sequencing Reads

Quality assurance as well as concerns regarding sample authenticity in biotechnology and bioengineering have always been serious topics in both production and research. While next generation sequencing techniques greatly enhance in-depth analyses of DNA-samples, they introduce additional error-sources. Resulting error-signatures can neither be easily removed from resulting sequencing data nor necessarily recognized, mainly due to the massive amount of data. Biologists and sequencing facility technicians face not only issues of minor relevance, e.g. suboptimal library preparation, but also serious incidents, including sample-contamination or even mix-ups, ultimately threatening the accuracy of biological conclusions.

While many problems cannot be addressed entirely, **QC for Sequencing Reads** assists in the quality control process by assessing and visualizing statistics relating to:

- Sequence read lengths and base-coverage
- Nucleotide contributions and base ambiguities
- Quality scores
- Over-represented sequences and hints suggesting contamination events

The inspiration for this tool came from the FastQC-project (<https://www.bioinformatics.babraham.ac.uk/projects/fastqc/>).

Note that currently, adapter contamination, i.e., adapter sequences in the reads, cannot be detected in a reliable way with this tool. In some cases, adapter contamination will show up as enriched 5-mers near the end of sequences, but only if the contamination is severe.

The tool supports long reads but will ignore any bases beyond the first 100kb.

To run **QC for Sequencing Reads**, go to:

Tools | Prepare Sequencing Data  | **QC for Sequencing Reads** 

Select one or more sequence lists as input. When multiple sequence lists are selected, they are analyzed together, as a single sample, by default. To generate separate reports for different inputs, check the **Batch** box below the selection area. More information about running tools in batch mode can be found in section [12.3](#).

In the "Result handling" wizard step, you can select the reports to generate, and whether you want a sequence list containing potential duplicate sequences to be created.

Two reports can be generated:

- **A graphical report** This contains plots of the various QC metrics. An example plot is shown in figure [28.1](#). To support the visualization, end positions with a coverage below 0.005% across the reads are not included. This is because the number of the longest reads in a set may be small, which can result in high variance at the end positions. If such positions are included in the plots, it can make other points hard to see.
- **A summary report** This contains tables of values for the various QC metrics, as well as general information such as the creation date, the author, the software used, the number of data sets the report is based upon, the data set names, total read number and total number of nucleotides. The maximum number of rows in each table is 500. If there are more than 500 data points, then tables include each read position or length for the first 100 bases, after which a bin range or *n*th position is used for successive rows.

Each report is divided into sections reporting per-sequence, per-base and over-representation analyses. In the per-sequence analyses, some characteristic (a single value) is assessed for each sequence and then contributes to the overall assessment. In per-base assessments each base position is examined and counted independently. In both these sections, the first items assess the most simple characteristics that are supported by all sequencing technologies while the quality analyses examine quality scores reported from technology-dependent base callers. Please note that the NGS import tools of the *CLC Genomics Workbench* and *CLC Genomics Server* convert quality scores to PHRED-scale, regardless of the data source.

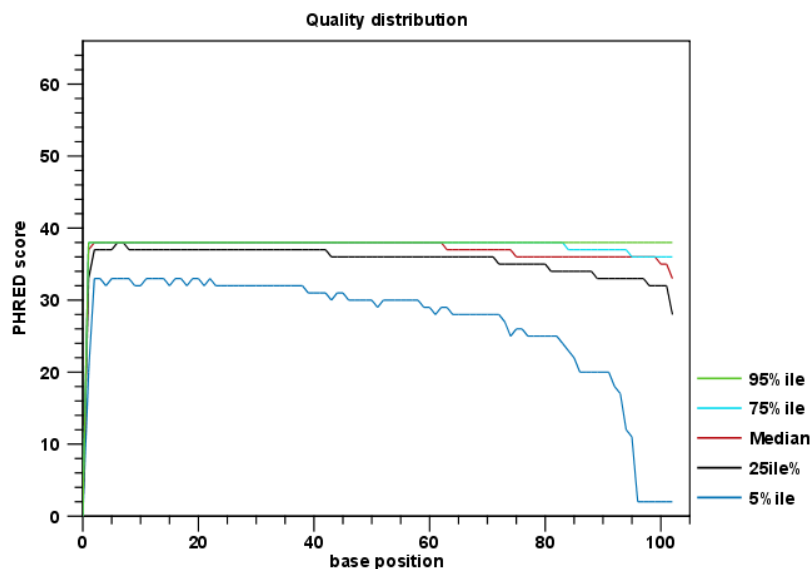


Figure 28.1: An example of a plot from the graphical report, showing the quality values per base position.

28.1.1 Per-sequence analysis

Lengths distribution Counts the number of sequences that have been observed for individual sequence lengths. The resulting table correlates sequence-lengths in base-pairs with numbers of sequences observed with that number of base-pairs. The length distribution depends on your library preparation and sequencing protocol. If you observe secondary peaks at unexpected lengths you may want to consider removing these. Using the Workbench Trim tool you can trim away reads above and/or below a certain length.

GC-content distribution Counts the number of sequences that feature individual %GC-contents in 101 bins ranging from 0 to 100%. The %GC-content of a sequence is calculated by dividing the absolute number of G/C-nucleotides by the length of that sequence, and should look like a normal distribution in the range of what is expected for the genome you are working with. If the GC-content is substantially lower (the normal distribution is shifted to the left), it may be that GC-rich areas have not been properly covered. You can check this by mapping the reads to your reference. A non-normal distribution, or one that has several peaks indicates the presence of contaminants in the reads.

Ambiguous base content Counts the number of sequences that feature individual %N-contents in 101 bins ranging from 0 to 100%, where N refers to all ambiguous base-codes as specified by IUPAC. The %N-content of a sequence is calculated by dividing the absolute number of ambiguous nucleotides through the length of that sequence. This distribution should be as close to 0 as possible.

Quality distribution Calculates the amount of sequences that feature individual PHRED-scores in 64 bins from 0 to 63. The quality score of a sequence as calculated as arithmetic mean of its base qualities. PHRED-scores of 30 and above are considered high quality. If you have many reads with low quality you may want to discuss this with your sequencing provider. Low quality bases/reads can also be trimmed off with the Trim Reads tool.

Reads passing average quality thresholds Calculates the percentage of sequences with average

quality higher than 20, 25, 30 and 35.

28.1.2 Per-base analysis

Please note that if the coverage is below 0.005% across the end positions of the reads, then these positions will not be shown in the plots described below (see section 28.1).

Coverage Calculates absolute coverages for individual base positions. The resulting graph correlates base-positions with the number of sequences that supported (covered) that position.

Nucleotide contributions Calculates absolute coverages for the four DNA nucleotides (A, C, G or T) for each base position in the sequences. In a random library you would expect little or no difference between the bases, thus the lines in this plot should be parallel to each other. The relative amounts of each base should reflect the overall amount of the bases in your genome. A strong bias along the read length where the lines fluctuate a lot for certain positions may indicate that an over-represented sequence is contaminating your sequences. However, if this is at the 5' or 3' ends, it will likely be adapters that you can remove using the Trim Reads tool.

GC-content Calculates absolute coverages of C's + G's for each base position in the sequences. If you see a GC bias with changes at specific base positions along the read length this could indicate that an over-represented sequence is contaminating your library.

Ambiguous base-content Calculates absolute coverages of N's, for each base position in the sequences, where N refers to all ambiguous base-codes as specified by IUPAC.

Quality distribution Calculates the amount of bases that feature individual PHRED-scores in 64 bins from 0 to 63. This results in a three-dimensional table, where dimension 1 refers to the base-position, dimension 2 refers to the quality-score and dimension 3 to amounts of bases observed at that position with that quality score. PHRED-scores above 20 are considered good quality. It is normal to see the quality dropping off near the end of reads. Such low-quality ends can be trimmed off using the Trim Reads tool.

28.1.3 Over-representation analyses

Please note that if the coverage is below 0.005% across the end positions of the reads, then these positions will not be shown in the enriched 5-mer distribution plot described below (see section 28.1).

Enriched 5-mer distribution The 5-mer analysis examines the enrichment of penta-nucleotides. The enrichment of 5-mers is calculated as the ratio of observed and expected 5-mer frequencies. The expected frequency is calculated as product of the empirical nucleotide probabilities that make up the 5-mer. (Example: given the 5-mer = CCCCC and cytosines have been observed to 20% in the examined sequences, the 5-mer expectation is 0.2^5). Note that 5-mers that contain ambiguous bases (anything different from A/T/C/G) are ignored. This analysis calculates the absolute coverage and enrichment for each 5-mer (observed/expected based on background distribution of nucleotides) for each base position, and plots position vs enrichment data for the top five enriched 5-mers (or fewer

if less than five enriched 5-mers are present). It will reveal if there is a bias at certain positions along the read length. This may originate from non-trimmed adapter sequences, poly A tails and more.

Sequence duplication levels The duplicated sequences analysis identifies sequence reads that have been sequenced multiple times. A high level of duplication may indicate an enrichment bias, as for instance introduced by PCR amplification. Please note that multiple input sequence lists will be considered as one federated data set for this analysis. Batch mode can be used to generate separate reports for individual sequence lists.

In order to identify duplicate reads the tool examines all reads in the input and uses a clone dictionary containing per clone the read representing the clone and a counter representing the size of the clone. For each input read these steps are followed: (1) check whether the read is already in the dictionary. (2a) if yes, increment the according counter and continue with next read. (2b) if not, put the read in the dictionary and set its counter to 1.

To achieve reasonable performance, the dictionary has a maximum capacity of 250,000 clones. To this end, step 2a involves a random decision as to whether a read is granted entry into the clone dictionary. Every read that is not already in the dictionary has the same chance T of entering the clone dictionary with $T = 250,000 / \text{total amount of input reads}$. This design has the following properties:

- The clone dictionary will ultimately contain at most 250,000 entries.
- The sum of all clone sizes in the dictionary amounts at most to the total number of input reads.
- Because of T being constant for all input reads, even a cluster of reads belonging to the same clone and first occurring towards the end of the input can be detected.
- Because of the random sampling, the tool might underestimate the size of a read clone, specifically if its first read representative does not make it into the dictionary. The ratio is that a larger clone has a higher cumulative chance of being eventually represented in the dictionary than a smaller clone.

Because all current sequencing techniques tend to report decreasing quality scores for the 3' ends of sequences, there is a risk that duplicates are NOT detected, merely because of sequencing errors towards their 3' ends. The identity of two sequence reads is therefore determined based on the identity of the first 50nt from the 5' end.

The results of this analysis are presented in a plot and a corresponding table correlating the clone size (duplication count) with the number of clones of that size. For example, if the input contains 10 sequences and each sequence was seen exactly once, then the table will contain only one row with duplication-count=1 and sequence-count=10. Note: due to space restrictions the corresponding bar-plot shows only bars for duplication-counts of $x=[0-100]$. Bar-heights of duplication-counts >100 are accumulated at $x=100$. Please refer to the table-report for a full list of individual duplication-counts.

Duplicated sequences This results in a list of actual sequences most prevalently observed. The list contains a maximum of 25 (most frequently observed) sequences and is only present in the supplementary report.

28.2 Trim Reads

CLC Genomics Workbench offers a number of ways to trim your sequence reads prior to assembly and mapping, including adapter trimming, quality trimming and length trimming. For each original read, the regions of the sequence to be removed for each type of trimming operation are determined independently according to choices made in the trim dialogs. The types of trim operations that can be performed are:

1. Quality trimming based on quality scores
2. Adapter trimming (automatic, or also with a Trim Adapter List, see section 28.2.2)
3. Homopolymer trimming
4. Sequence trimming to remove a specified number of bases at either 3' or 5' end of the reads
5. Sequence filtering to remove reads shorter or longer than a specified threshold

The trim operation that removes the largest region of the original read from either end is performed while other trim operations are ignored as they would just remove part of the same region.

Note that this may occasionally expose an internal region in a read that has now become subject to trimming. In such cases, trimming may have to be done more than once.

The result of the trim is a list of sequences that have passed the trim (referred to as the trimmed list below) and optionally a list of the sequences that have been discarded and a summary report (list of discarded sequences). The original data will not be changed.

To start trimming:

Tools | Prepare Sequencing Data  **| Trim Reads** 

This opens a dialog where you can add sequences or sequence lists. If you add several sequence lists, each list will be processed separately and you will get a list of trimmed sequences for each input sequence list.

When the sequences are selected, click **Next**.

28.2.1 Quality trimming

This opens the dialog displayed in figure 28.2 where you can specify parameters for quality trimming.

The following parameters can be adjusted in the dialog:

- **Trim using quality scores.** If the sequence files contain quality scores from a base caller algorithm this information can be used for trimming sequence ends. The program uses the modified-Mott trimming algorithm for this purpose (Richard Mott, personal communication): Quality scores in the Workbench are on a Phred scale, and formats using other scales will be converted during import. The Phred quality scores (Q), defined as: $Q = -10\log_{10}(P)$, where

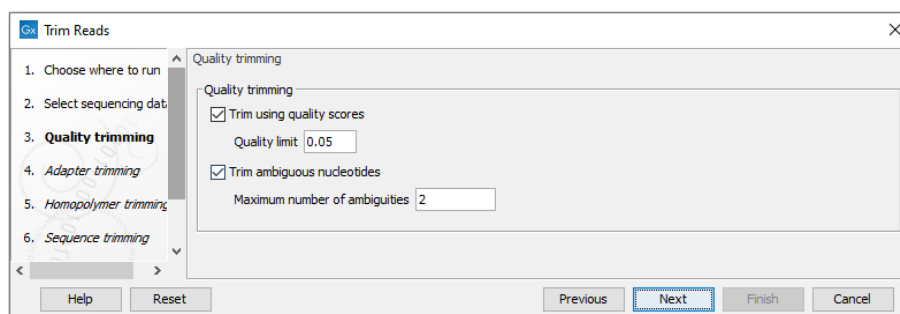


Figure 28.2: Specifying quality trimming.

P is the base-calling error probability, can then be used to calculate the error probabilities, which in turn can be used to set the limit for, which bases should be trimmed.

Hence, the first step in the trim process is to convert the quality score (Q) to an error probability: $p_{error} = 10^{-\frac{Q}{10}}$. (This now means that low values are high quality bases.)

Next, for every base a new value is calculated: $Limit - p_{error}$. This value will be negative for low quality bases, where the error probability is high.

For every base, the Workbench calculates the running sum of this value. If the sum drops below zero, it is set to zero. The part of the sequence not trimmed will be the region ending at the highest value of the running sum and starting at the last zero value before this highest score. Everything before and after this region will be trimmed. A read will be completely removed if the score never makes it above zero.

At <https://resources.qiagenbioinformatics.com/testdata/trim.zip> you find an example sequence and an Excel sheet showing the calculations done for this particular sequence to illustrate the procedure described above.

- **Trim ambiguous nucleotides.** This option trims the sequence ends based on the presence of ambiguous nucleotides (typically N). Note that the automated sequencer generating the data must be set to output ambiguous nucleotides in order for this option to apply. The algorithm takes as input the *maximal number of ambiguous nucleotides allowed in the sequence after trimming*. If this maximum is set to e.g. 3, the algorithm finds the maximum length region containing 3 or fewer ambiguities and then trims away the ends not included in this region. The "Trim ambiguous nucleotides" option trims all types of ambiguous nucleotides (see Appendix H).

28.2.2 Adapter trimming

Clicking **Next** will allow you to specify parameters for adapter trimming.

When you are analyzing sequencing data, the adapters must be trimmed off before you proceed with further analysis. The removal of adapters is often done directly on the sequencing machine, but in some cases, some adapters remain on the sequenced reads. The presence of remaining adapters can lead to misleading results, so we recommend to trim them off the reads (figure 28.3).

The default option for this trimming step is to use the "Automatic read-through adapter trimming", which will detect read-through adapter sequence on paired-end reads automatically. Read-through means that the sample DNA fragment being sequenced is shorter than the read length, such that the 3' end of one read includes the reverse-complement of the adapter from the start of the other

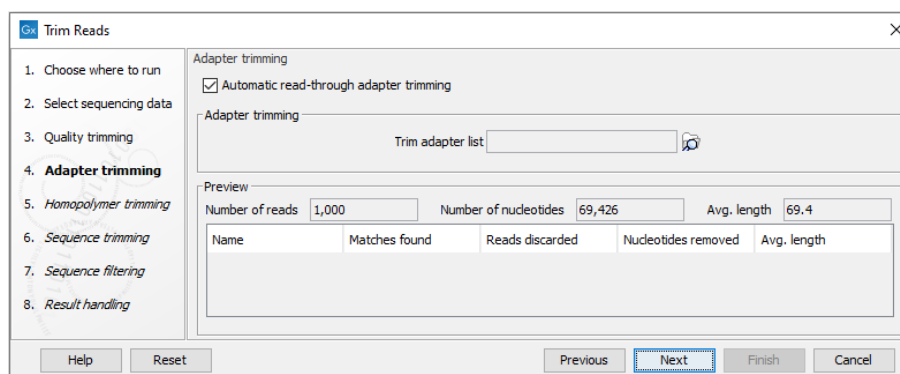


Figure 28.3: *Trimming your sequencing data for adapter sequences.*

read. Leaving this option enabled is always recommended: the trimming performed automatically can detect read-through of even a single nucleotide, which is not the case when trimming using a trim adapter list. When analyzing multiple sequence lists, automatic read-through adapter trimming analyzes and trims each list individually. The detected adapters for the first and second read for each sequence list can be found in the Trim Reads report.

There are a couple of limitations on the "Automatic read-through adapter trimming" option: this option detects overlap in paired reads containing standard nucleotides (A, T, C, and G). If the read contains ambiguous symbols, such as N, these will not match the standard nucleotides.

Also, the first and second read should be of equal (or near-equal) length - some sequencing protocols use asymmetric read lengths for the first and second read, in which case the tool is less likely to detect and trim the read-through.

So when you are working with data of low quality, asymmetric read lengths, mate-paired reads, single reads, small RNAs, or also when working with gene specific primers, it is recommended that you specify a trim adapter read in addition to using the "Automatic read-through adapter trimming" option. It is even possible to use the report of the Trim Read tool to find out what Trim adapter list should be used for the data at hand. Read section 28.2.3 to learn how to create an adapter list.

Below you find a preview listing the results of trimming with the adapter trimming list on 1000 reads in the input file (reads 1001-2000 when the read file is long enough) (figure 28.3). This is useful for a quick feedback on how changes in the parameters affect the trimming (rather than having to run the full analysis several times to identify a good parameter set). The following information is shown:

- **Name.** The name of the adapter.
- **Matches found.** Number of matches found based on the settings.
- **Reads discarded.** This is the number of reads that will be completely discarded. This can either be because they are completely trimmed (when the **Action** is set to Remove adapter and the match is found at the 3' end of the read), or when the **Action** is set to Discard when found or Discard when not found.
- **Nucleotides removed.** The number of nucleotides that are trimmed include both the ones coming from the reads that are discarded and the ones coming from the parts of the reads that are trimmed off.

- **Avg. length** This is the average length of the reads that are retained (excluding the ones that are discarded).

Note that the preview panel is only showing how the trim adapter list will affect the results. Other kinds of trimming (automatic trimming of read-through adapters, quality or length trimming) are not reflected in the preview table.

28.2.3 Trim adapter list

The Trim Reads tool is set by default to detect automatically read-through adapters present in the reads used as input for the tool. We recommend to always enable this option. In addition, you can use a Trim adapter list for a more thorough trimming of read-through adapter in reads of lower quality, or to trim for specific adapters that are not read-through such as small RNAs, gene specific primers, or when working with single or mate-paired reads. In such cases, you have to import or create a Trim adapter list that must be supplied to the Trim Reads tool.

Creating a new Trim adapter list

It is possible to generate a Trim adapter list directly in the workbench. Go to:

File | New | Trim Adapter List

This will create a new empty Trim adapter list. At the bottom of the view, you have the following options that allow you to edit the Trim adapter list:

- **Add Rows.** Add a new adapter.
- **Edit Row.** Edit the selected adapter. This can also be achieved by double-clicking the relevant row in the table.
- **Delete Row.** Delete the selected adapter.

Add the adapter(s) that you would like to use for trimming by clicking on the button **Add Row** (+) found at the bottom of the View Area. Adding an adapter is done in two steps. In the first wizard step (figure 28.4), you enter the basic information about the adapter, and how the trimming should be done relative to the adapter found.

In the second dialog (figure 28.5), you define the scores that will be used to recognize adapters. For each read sequence in the input, a Smith-Waterman alignment [Smith and Waterman, 1981] is carried out with each adapter sequence. Alignment scores are computed and compared to the minimum scores provided for each adapter when setting up the Trim adapter List. If the alignment score is higher or equal to the minimum score, the adapter is recognized and the trimming can happen as specified in the first wizard. If however the alignment score is lower than the minimum score, the adapter is not recognized and trimmed.

Trim adapter

Start by providing the name and sequence of the adapter that should be trimmed away. Use the **Reverse Complement** button to reverse complement the sequence you typed in if it is found in reverse complement in the reads. You can then specify whether you want the adapter to be trimmed on all reads, or more specifically on the first or second read of a pair.

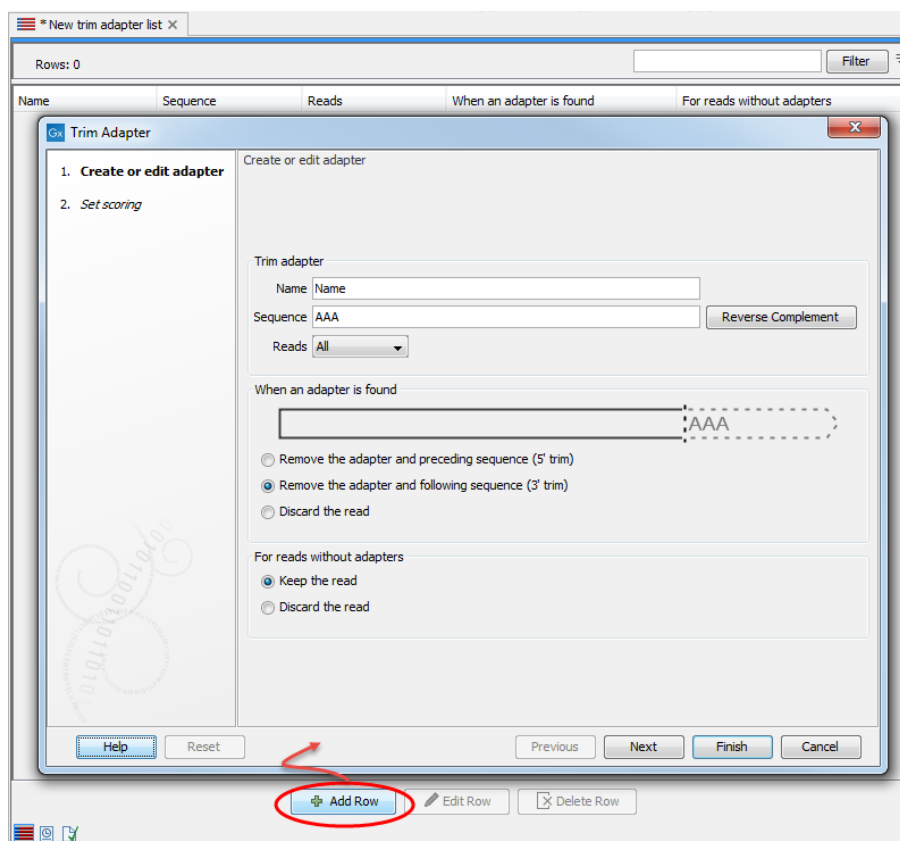


Figure 28.4: Add an adapter to the Trim Adapter List by clicking on the button labeled "Add Row" found at the bottom of the New Trim Adapter view.

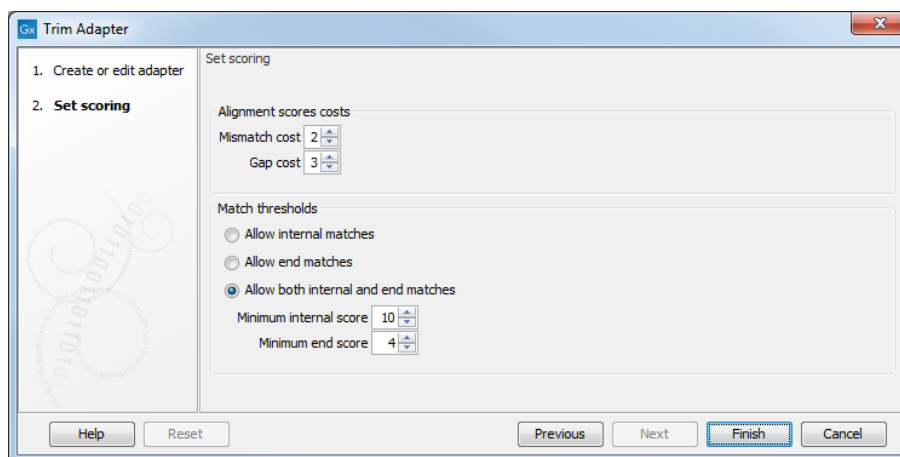


Figure 28.5: Set the scoring used to define what will be considered as adapter.

When an adapter is found

Once you have entered the sequence of the adapter, a visual shows how the adapter will be trimmed, allowing you to decide which option suits your needs best:

- Remove the adapter and preceding sequence (5' trim)

- Remove the adapter and following sequence (3' trim)
- Discard the read. The read will be placed in the list of discarded sequences. This can be used for quality checking the data for linker contamination for example.

For reads without adapters

You can decide here what to do with reads where no adapter was found. This kind of adapter trimming is particularly useful for small RNA sequencing where the remnants of the adapter is an indication that this is indeed a small RNA. Beware of lists where multiple adapters have been set to "Discard the read" when the adapters are not found: only sequences containing **all** the adapters will remain in the list of trimmed reads.

Alignment scores costs

An A,C,G or T in the adapter that matches an A,C,G or T respectively - or a corresponding ambiguity code letter - in a sequence is considered a match and will be awarded 1 point. However, you can decide how much penalty should be awarded to mismatches and gaps:

- **Mismatches** The penalty for mismatches between bases is set as 2 by default.
- **Gap** The penalty for gaps introduced into the alignment is set as 3 by default.

Here are the few examples of adapter matches and corresponding scores (figure 28.6). These examples are all internal matches where the alignment of the adapter falls within the read.

```

CGTATCAATCGATTACGCTATGAATG
a)  ||||| ||| 11 matches - 2 mismatches = 7
    TTCAATCGGTTAC

CGTATCAATCGATTACGCTATGAATG
b)  ||||| ||| 14 matches - 1 gap = 11
    ATCAATCGAT-CGCT

CGTATCAATCGATTACGCTATGAATG
c)  ||||| 7 matches - 3 mismatches = 1
    TTCAATCGGG

```

Figure 28.6: Three examples showing a sequencing read (top) and an adapter (bottom). The examples are artificial, using default setting with mismatch costs = 2 and gap cost = 3.

Match thresholds

Note that there is a difference between an **internal match** and an **end match**. Figures 28.6 and 28.7 shows some examples: An end match happens when the alignment of the adapter starts at the end of the sequence that is being trimmed. An internal match happens when the alignment of the adapter does not start at the end of the sequence that is being trimmed, but rather occurs within the sequence or towards the end that is not being trimmed. Which end is being trimmed is determined by the option chosen in the first dialog: it can be 5' or 3'. Thus, in case of 3' trim, if a match is found at the 5' end, it will be treated as an internal match, because it is on the end of the sequence that is not being trimmed.

This section allows you to decide whether to

- Allow internal matches
- Allow end matches
- Allow both internal and end matches

You can also change the minimum scores for both internal and end score

- Minimum internal score is set to 10 by default
- Minimum end score is set to 4 by default

End matches have usually a lower score, as adapters found at the end of reads may be incomplete.

For example, if your adapter is 8 nucleotides long, it will never be found in an internal position with the settings set as they are by default (the minimum internal score being at 10).

Figure 28.7 shows a few examples with an adapter match at the end.

```

d)      CGTATCAATCGATTACGCTATGAATG
        |||||
        GATTTCGTAT      5 matches = 5 (as end match)

e)      CGTATCAATCGATTACGCTATGAATG
        || ||||
        GATTTCGCATCA    6 matches - 1 mismatch = 4 (as end match)

f)      CGTATCAATCGATTACGCTATGAATG
        |||| ||||
        CGTA-CAATC      9 matches - 1 gap = 6 (as end match)

g)      CGTATCAATCGATTACGCTATGAATG
                |||||
                GCTATGAATG    10 matches = 10 (as internal match)

```

Figure 28.7: Four examples showing a sequencing read (top) and an adapter (bottom). The examples are artificial.

In the first two examples (d and e), the adapter sequence extends beyond the end of the read. This is what typically happens when sequencing small RNAs where you sequence part of the adapter. The third example (f) shows a case that could be interpreted both as an end match and an internal match. However, the workbench will interpret this as an end match, because it starts at beginning (5' end) of the read. Thus, the definition of an end match is that the alignment of the adapter starts at the read's 5' end. The last example (g) could also be interpreted as an end match, but because it is at the 3' end of the read, it counts as an internal match (this is because you would not typically expect partial adapters at the 3' end of a read).

Below (figure 28.8), the same examples are re-iterated showing the results when applying different scoring schemes. In the first round, the settings are:

- When an adapter is found: Remove adapter and the preceding sequence (5' trim)
- Allowing internal matches with a minimum score of 6
- Not allowing end matches

a)	CGTATCAATCGATTACGCTATGAATG TTCAATCGGTTAC	11 matches - 2 mismatches = 7
b)	CGTATCAATCGATTACGCTATGAATG ATCAATCGAT-ACGC	14 matches - 1 gap = 11
c)	CGTATCAATCGATTACGCTATGAATG TTCAATCGGG	7 matches - 3 mismatches = 1
d)	CGTATCAATCGATTACGCTATGAATG GATTCGTAT	5 matches = 5 (as end match)
e)	CGTATCAATCGATTACGCTATGAATG GATTCGCATCA	6 matches - 1 mismatch = 4 (as end match)
f)	CGTATCAATCGATTACGCTATGAATG CGTA-CAATC	9 matches - 1 gap = 6 (as end match)
g)	CGTATCAATCGATTACGCTATGAATG GCTATGAATG	10 matches = 10 (as internal match)

Figure 28.8: The results of trimming with internal matches only. Red is the part that is removed and green is the retained part. Note that the read at the bottom is completely discarded.

A different set of adapter settings could be:

- When an adapter is found: Remove adapter and the preceding sequence (5' trim)
- Allowing internal matches with a minimum score of 11
- Allowing end match with a minimum score of 4

The results of such settings is shown in figure 28.9.

a)	<pre> CGTATCAATCGATTACGCTATGAATG TTCAATCGGTTAC </pre>	11 matches - 2 mismatches = 7
b)	<pre> CGTATCAATCGATTACGCTATGAATG ATCAATCGAT-CGCT </pre>	14 matches - 1 gap = 11
c)	<pre> CGTATCAATCGATTACGCTATGAATG TTCAATCGGG </pre>	7 matches - 3 mismatches = 1
d)	<pre> CGTATCAATCGATTACGCTATGAATG GATTCGTAT </pre>	5 matches = 5 (as end match)
e)	<pre> CGTATCAATCGATTACGCTATGAATG GATTCGCATCA </pre>	6 matches - 1 mismatch = 4 (as end match)
f)	<pre> CGTATCAATCGATTACGCTATGAATG CGTA-CAATC </pre>	9 matches - 1 gap = 6 (as end match)
g)	<pre> CGTATCAATCGATTACGCTATGAATG GCTATGAATG </pre>	10 matches = 10 (as internal match)

Figure 28.9: The results of trimming with both internal and end matches. Red is the part that is removed and green is the retained part.

Click **Finish** to create the trim adapter list. You must now save the generated trim adapter list in the **Navigation Area**. You can do this by clicking on the tab and dragging and dropping the trim adapter list to the desired destination, or you can go to **File** in the menu bar and choose **Save as**.

Creating a Trim adapter list based on the Trim Reads report

Since the automatic option works conservatively with data of low quality, you can benefit from creating a new Adapter Trim List based on the report generated by running the Trim Reads tool a first time.

1. Start the Trim Reads tool.
2. Select the reads you want to analyze (or a subset of these).
3. Leave the Quality trimming settings as they are set by default.
4. In the Adapter trimming step, make sure that the option "Automatic read-through adapter trimming" is selected and that no Adapter Trim List is specified.
5. Leave the Sequence filtering settings at their default value, i.e. with no filtering.

6. In the Result handling step ensure that "Create Report" is selected and click Finish.

Once the process is completed, open the report and scroll down to the last section named "5 Automatic adapter read-through trimming" (as seen in figure 28.10).

5 Automatic adapter read-through trimming

Processed read pairs	119057
Read pairs trimmed	72
Read pairs trimmed (percent)	0.06%

Statistics for read 1

Read 1 trimmed	15
Read 1 trimmed (percent)	0.01%
Read-through sequence	CTGTCTCTTATACACATTCCCAACCCACGAGACCATCACGGATCTCGTATG CCGTCTTCTGCTTTAAAAAAAAAAAAACAACAAGACAACACCAAAATA
Read-through sequence (High confidence)	CT

Statistics for read 2

Read 2 trimmed	65
Read 2 trimmed (percent)	0.05%
Read-through sequence	CTGTCTCTTATACACATCTGACGCTGCCGACGCCTAGTCGAGTGTAGATC CCGGTGGTCCCGGATCATTCAAAACAAAAACCCATGCAACCGCGACC
Read-through sequence (High confidence)	C

Figure 28.10: Use the statistics of the read-through trimming to create a Trim adapter list.

- If the detected "Read-through sequence" is < 10 bp, read-through adapters are not a big issue in your data and it can be trimmed using the "Automatic read-through adapter trimming" on its own. You do not need to re-run the tool with an adapter trimming list.
- If the detected "Read-through sequence" is > or equal to 10 bp, we recommend that you re-run the Trim Reads tool with a Trim adapter list generated using the report results.

To create a Trim adapter list with the read-through sequence from the report:

1. In the report, copy the sequence of the detected "Read-through sequence". If the sequence is long, then copy only the first 19 to 24 bp.
2. Go to **New | Adapter Trim List**.
3. Click the **Add row** button.
4. Type the name of the first adapter, for example Read 1 read-through adapter.
5. Paste the copied sequence.
6. Set the Reads option to **First read**.
7. Choose the option **Remove the adapter and the following sequence (3' trim)**.
8. For reads without adapters choose the option **Keep the Read**.
9. In the Set scoring dialog, leave the default settings and click **Finish**.
10. Repeat for the procedure with the read-through sequence for read 2.

11. Save the Trim adapter list before closing it.

You can now use this Trim adapter list in combination with the "Automatic read-through adapter trimming" option for optimal adapter trimming of all samples in your experiment.

Importing an adapter list

It is possible to import a trim adapter list from an Excel or CSV file, using the Standard import with either the Automatic Import option, or the Force Import as Type: Trim Adapter List option.

To import a trim adapter list, the names of all adapters must be unique - the workbench is unable to accept files with multiple rows containing the same adapter name. The file must also include the following information: Name, Sequence, Reads, When an adapter is found, For reads without adapters, Alignment score

For CSV file, the text between each comma that designates a new column can be quoted, as shown in figure 28.11:

```
"Name","Sequence","Reads","when an adapter is found","For reads without adapters","Alignment score"
"ATC 5' trim all","ACGCTAGTCATA","All","Trim 5' end","Keep the read","Mismatch: 2, Gapcost: 3, Cutoff: 10, Cutoff at end: 4"
"ATC 3' trim first read","ACGCTAGTCAGTCTA","First read","Trim 3' end","Keep the read","Mismatch: 2, Gapcost: 3, Cutoff: 10, Cutoff at end: 4"
"ATC discard second read","ACGCTGTCAGTCAGTCTA","Second read","Trim 3' end","Keep the read","Mismatch: 2, Gapcost: 3, Cutoff: 10, Cutoff at end: 4"
"ATC discard reads when not found","ACGCTAGTCAGTCTA","All","Trim 5' end","Discard the read","Mismatch: 2, Gapcost: 3, Cutoff: 10, Cutoff at end: 4"
```

Figure 28.11: The expected import format for Adapter Lists.

28.2.4 Homopolymer trimming

Configuration for the homopolymer trimming step is shown in figure 28.12.

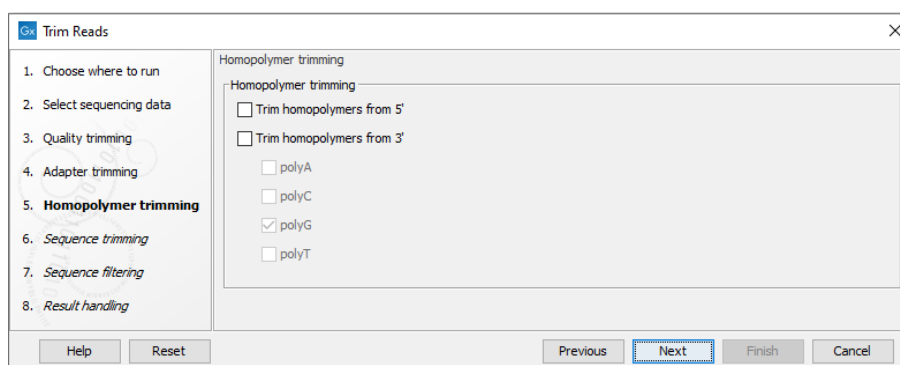


Figure 28.12: Homopolymer trimming.

Homopolymer trimming takes place only if at least one read end type is selected. After selecting the read end(s) to trim, you can select the type of homopolymer stretches to be removed.

How it works

Trimming of each type of homopolymer at each read end is done in the same way.

Using polyG as an example:

- A window of 10 nucleotides at the end of the read is initially checked.

- If fewer than 9 bases are Gs, then checking stops and no bases are trimmed.
- If all 10 bases are Gs, they are marked for trimming.
- If 9 out of 10 bases are Gs, all 10 bases are marked for trimming unless the non-G base is at the end of the 10 bases. In the following examples, where trimming takes place from left to right, the only base that is not marked for trimming is in bold:
 - NGG GGG GGG G
 - GGG GGN GGG G
 - GGG GGG GGN G
 - GGG GGG GGG **N**

The window then slides by one position, to cover 9 of the original bases and 1 additional base, and the steps described above are repeated.

This process continues until the sliding 10-base window contains fewer than 9 Gs. At that point, checking stops and all bases marked to be trimmed are removed.

Examples of the effects of trimming particular sequences:

- Trimming the sequence `CCCCCTCCCCCATATATATATATATCCCCCTCCCC` for polyC at the start and end of the read would result in `ATATATATATATAT`.
- Trimming the sequence `CCCCCTCCCCCATATATATATATATTTTTTTTTTGT` for polyT and polyC at the start and end of the read would result in `ATATATATATA`. Note that a TA is removed at the 3' end. This is because the 10-base window `TATTTTTTTT` contains nine Ts, and thus all 10 bases in this stretch are removed.
- Trimming the sequence `AAAAAAAAAATTTTTTTTTTGT` for polyA and polyT at the start and end of the read would result in the whole sequence being trimmed away.

28.2.5 Sequence trimming

Clicking **Next** will allow you to specify sequence trimming as shown in figure 28.13.

At the top, you can choose to **Remove fixed number of bases** by specifying a number of bases to be removed from either the 3' or the 5' end of the reads.

At least one of the options **Remove on first read** or **Remove on second read (for paired reads)** must be checked. If none are checked, reads are not trimmed.

You can also **Trim reads to a fixed length** of your choice, choosing whether to trim them starting from the 3' or the 5' end.

28.2.6 Sequence filtering

Clicking **Next** will allow you to specify length filtering as shown in figure 28.14.

Here you can choose to **Discard reads below length**. This can be used if you wish to simply discard reads because they are too short. Similarly, you can discard reads above a certain length. This will typically be useful when investigating e.g. small RNAs (note that this is an integral part of the small RNA analysis together with adapter trimming).

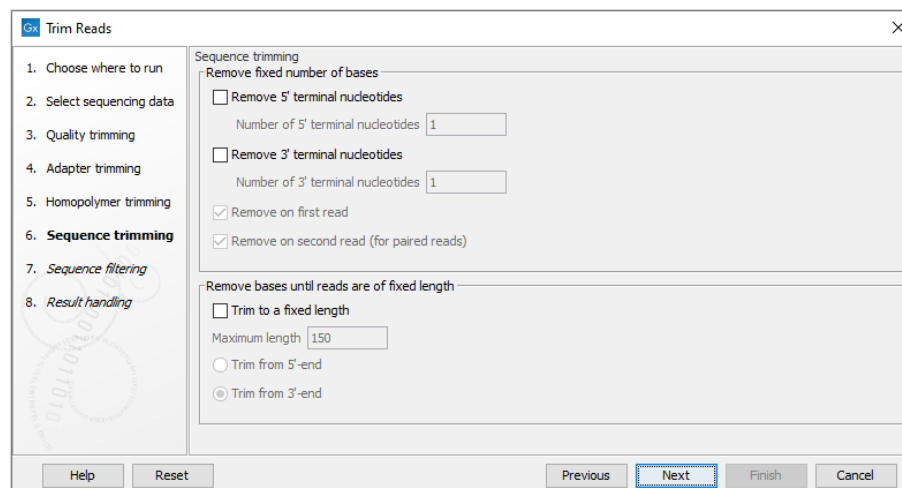


Figure 28.13: Sequence trimming.

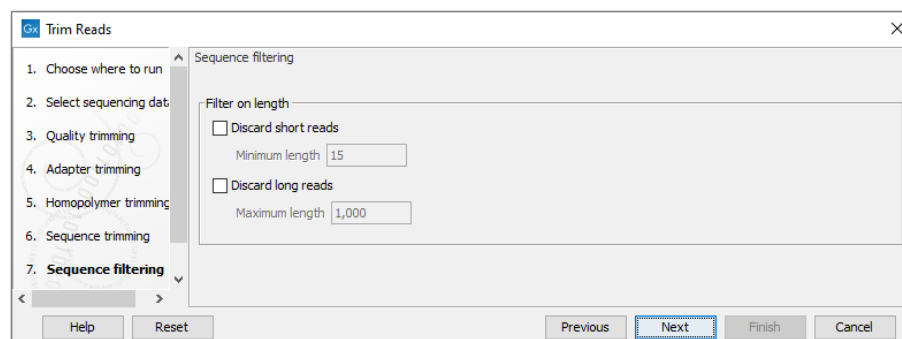


Figure 28.14: Filtering on length.

28.2.7 Trim output

Clicking **Next** will allow you to specify the output of the trimming as shown in figure 28.15.

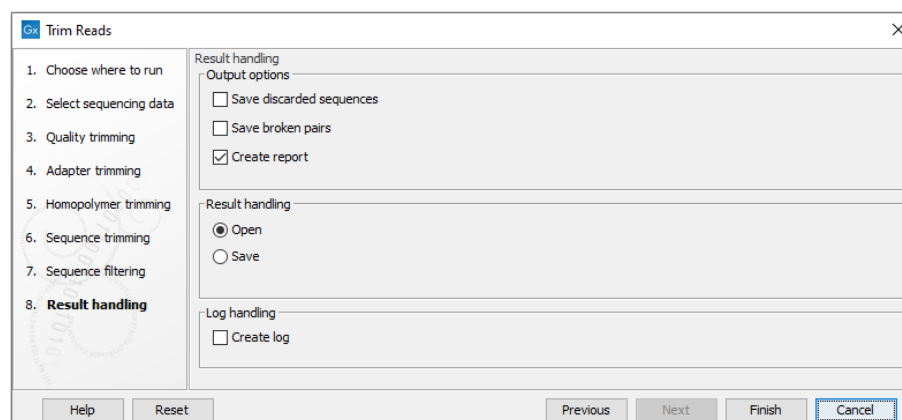


Figure 28.15: Specifying the trim output.

In most case, independently of what option are selected in this dialog, a list of trimmed reads will be generated:

- Sequence elements (individual sequences) selected as input and not discarded during trimming will be output into a single sequence list, as long as one or more of the input sequences were trimmed.
- Sequence lists selected as input will be output as as many corresponding sequence list, assuming that at least one sequence in any one of the sequence lists input was trimmed.

However, if no sequences are trimmed using the parameter settings provided, then no sequence lists are output when running the tool directly. A warning message appears stating that no sequences were trimmed. When the tool is run within a workflow, and if no sequences are trimmed using the parameter settings provided, then all input sequences are passed to the next step of the analysis via the "Trimmed Sequences" output channel.

In addition the following can be output as well:

- **Save discarded sequences.** This will produce a list of reads that have been discarded during trimming. Sections trimmed from reads that are not themselves discarded will not appear in this list.
- **Save broken pairs.** This will produce a list of orphan reads.
- **Create report.** An example of a trim report is shown in figure 28.16. The report includes the following:
 - **Trim summary.**
 - * **Name.** The name of the sequence list used as input.
 - * **Number of reads.** Number of reads in the input file.
 - * **Avg. length.** Average length of the reads in the input file.
 - * **Trimmed sequences.** The number of reads after trimming, not including orphan reads.
 - * **Trimmed (broken pairs).** The number of broken pairs after trimming (orphan reads).
 - * **Total number of reads after trim.** The total number of reads retained after trimming. This includes both paired and orphan reads.
 - * **Percentage trimmed.** The percentage of the input reads that are retained.
 - * **Avg. length after trim.** The average length of the retained sequences.
 - **Read length before / after trimming.** This is a graph showing the number of reads of various lengths. The numbers before and after are overlayed so that you can easily see how the trimming has affected the read lengths (right-click the graph to open it in a new view).
 - **Trim settings** A summary of the settings used for trimming.
 - **Detailed trim results.** A table with one row for each type of trimming:
 - * **Input reads.** The number of reads used as input. Since the trimming is done sequentially, the number of retained reads from the first type of trim is also the number of input reads for the next type of trimming.
 - * **No trim.** The number of reads that have been retained, unaffected by the trimming.
 - * **Trimmed.** The number of reads that have been partly trimmed. This number plus the number from **No trim** is the total number of retained reads.

- * **Nothing left or discarded.** The number of reads that have been discarded either because the full read was trimmed off or because they did not pass the length trim (e.g. too short) or adapter trim (e.g. if **Discard when not found** was chosen for the adapter trimming).
- **Automatic adapter read-through trimming.** This section contains statistics about how many reads were automatically trimmed for adapter read-through. It will also list the two detected read-through sequences for each analyzed sequence list.

1 Trim summary

Name	Number of reads	Avg.length	Number of reads after trim	Percentage trimmed	Avg.length after trim
reads	57.213	228,0	55.754	~100%	232,8

2 Read length before / after trimming

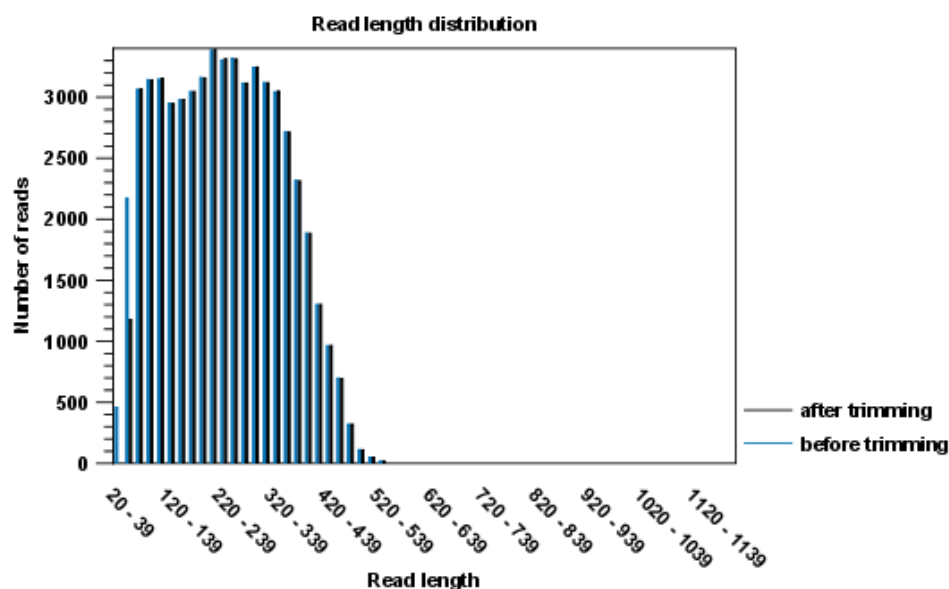


Figure 28.16: A report with statistics on the trim results. Note that the Average length after trimming (232,8bp) is bigger than before trimming (228bp) because 2.000 very short reads were discarded in the trimming process.

If you trim paired data, the result will be a bit special. In the case where one part of a paired read has been trimmed off completely, you no longer have a valid paired read in your sequence list. In order to use paired information when doing assembly and mapping, the Workbench therefore creates two separate sequence lists: one for the pairs that are intact, and one for the single reads where one part of the pair has been deleted. When running assembly and mapping, simply select both of these sequence lists as input, and the Workbench will automatically recognize that one has paired reads and the other has single reads.

When placed in a workflow and connected to another downstream tool or output element, the Trim Reads tool will always generate all outputs (including the report), leading to the following situations:

- When no reads have been trimmed (either because all trimming options were deselected, or because none of the trim options matched any of the reads), the "Trimmed sequences" output will contain all input reads, "Discarded sequences" will be empty, and "Percentage trimmed" will be 100% in the report.
- When all reads have been trimmed, the "Discarded sequences" output will contain all input reads, "Trimmed sequences" will be empty, and "Percentage trimmed" will be 0%.

28.3 Demultiplex Reads

Multiplexing techniques are often used when sequencing different samples in one sequencing run. One method used is to *tag* the sequences with a unique identifier during the preparation of the sample for sequencing [Meyer et al., 2007].

With this technique, each sequence read will have a sample-specific tag, which is a specific sequence of nucleotides before or after the sequence of interest, see figure 28.17.

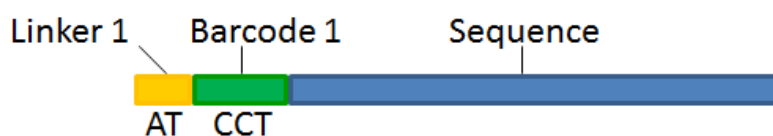


Figure 28.17: Tagging the target sequence, which in this case is single reads from one sample.

Demultiplex Reads looks for matches between reads and these sample-specific tags, also called barcodes or indexes, to group the reads by sample. The reads for each sample can then be used in downstream analyses to generate sample-specific results.

When **Demultiplex Reads** is used within a workflow, the 'Demultiplexed Reads' output channel needs to be connected to an Iterate element. The sets of reads to be analyzed together, i.e. the batch units (see section 14.3.3), are determined by the barcodes. See section 28.3.3 for further details.

Demultiplexing is often carried out on the sequencing machine so that the sequencing reads are already separated according to sample before importing it into the *CLC Genomics Workbench*. This is often the best option, if available.

28.3.1 Running Demultiplex Reads

To run **Demultiplex Reads**, go to:

Tools | Prepare Sequencing Data (📁) | **Demultiplex Reads** (🔗)

Select one or more sequence lists as input. When you click on the button labeled **Next**, you can then specify the details of how the demultiplexing should be performed.

Defining the read structure

The 'Define tags' wizard contains three buttons, which are used to **Add**, **Edit**, and **Delete** the tags that describe the structure of the reads and how the barcode is embedded in the sequences.

Click on **Add** to define the first tag. This will bring up the 'Define tag' dialog (figure 28.18).

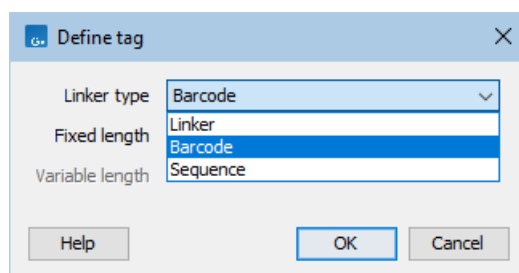


Figure 28.18: *Defining a tag.*

At the top of the 'Define tag' dialog, you can choose the type of tag you wish to define:

- **Linker.** The linker (also known as adapter) is a sequence which should just be ignored - it is neither the barcode nor the sequence of interest. In the example in figure 28.17, the linker is two nucleotides long. For this, you simply define its length - nothing else.
- **Barcode.** The barcode (also known as index) is the stretch of nucleotides used to group the sequences. For this, you simply define the barcode length. The valid sequences for your barcodes are provided in the 'Set barcode options' wizard step, see below.
- **Sequence.** Defines the sequence of interest. You can define a length interval for how long you expect this sequence to be. The sequence part is the only part of the read that is retained in the output. Both barcodes and linkers are removed.

The concept when adding tags is that you add e.g. a linker, a barcode, and a sequence in the desired sequential order to describe the structure of each sequencing read. You can edit and delete elements by selecting them and clicking the buttons below. For the example shown in figure 28.17, the structure should include a linker, a barcode, and a sequence as shown in figure 28.19.

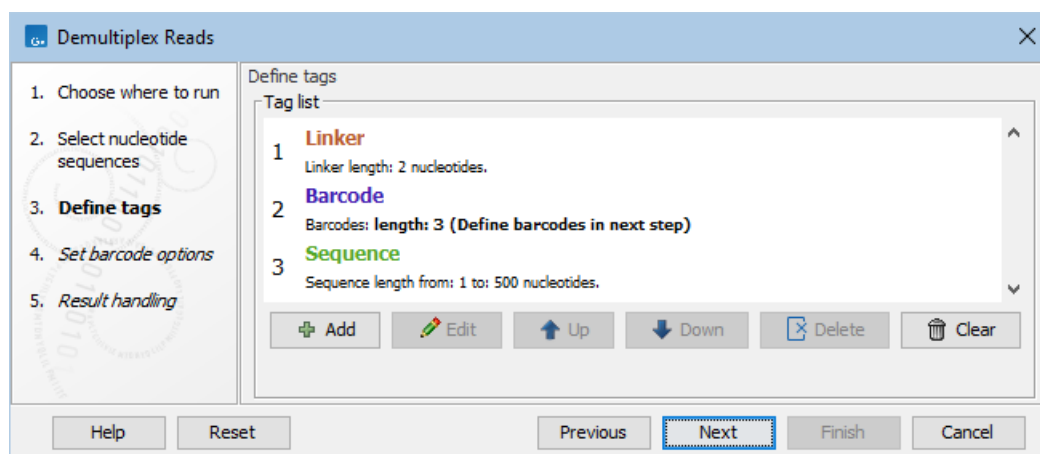


Figure 28.19: *Processing the tags as shown in the example of figure 28.17.*

If the input contains paired reads, there are two wizards for defining the read structure: 'Define tags' for R1 and 'Define tags (mate)' for R2. If the two reads in the read pair have different barcodes such as illustrated in figure 28.20, the read structure would look like this:

R1 : -Linker1-Barcode1-Sequence

R2 : -Linker2-Barcode2-Sequence

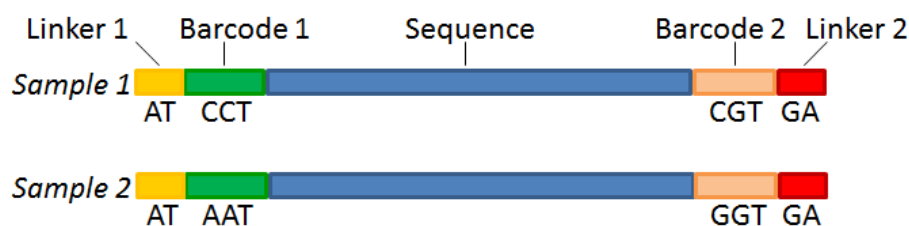


Figure 28.20: Paired reads with linkers and barcodes originating from two different samples.

Defining the barcodes

Click **Next** to set the barcode options (figure 28.21).

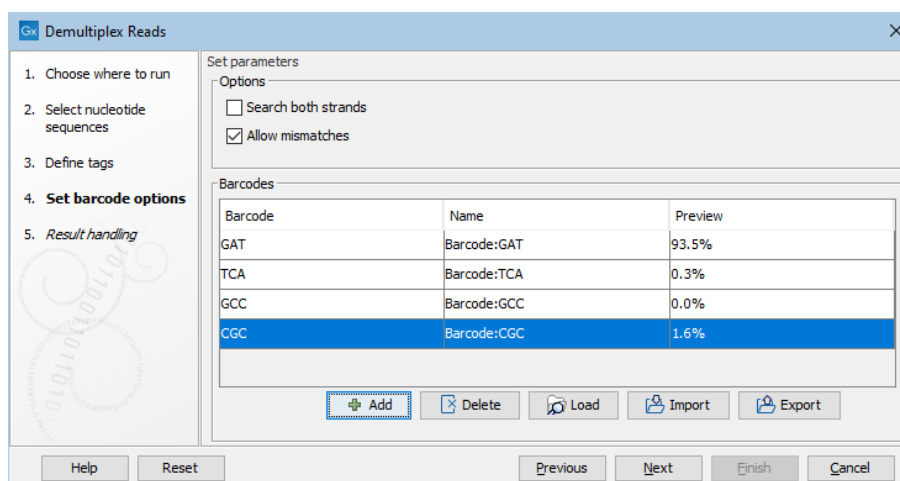


Figure 28.21: Defining the barcodes.

At the top, you can choose to search on both strands for the barcodes.

You can also choose to allow mismatches: only one per barcode will be allowed, regardless of whether the barcodes are on the same read, or distributed on both R1 and R2. Note that if a sequence is one mismatch away from two barcodes, it will not be assigned to either of them.

Barcodes can be provided in several ways:

- **Manually enter barcodes.** Click the **Add** (➡) button, see figure 28.22.
- **Load barcodes from a table element.** Click the **Load** (📄) button. The first two columns in the table element must contain the expected barcodes and their names, respectively. For example:

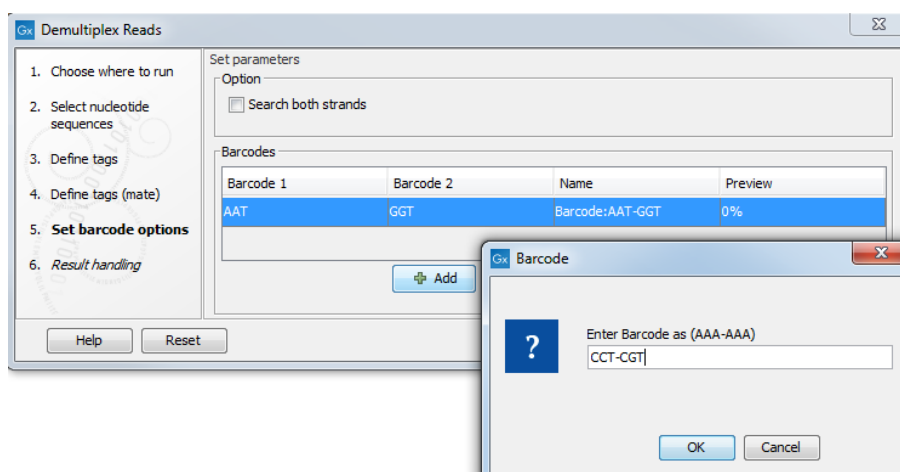


Figure 28.22: The barcodes for the set of paired end reads for sample 1 have already been defined and the barcodes for sample 2 is being entered in the format AAA-AAA, which corresponds to Barcode1-Barcode2 for sample 2 in the example shown in figure 28.20.

Barcode	Name
AAAAAA	Sample1
GGGGGG	Sample2
CCCCCC	Sample2

- **Import barcodes from CSV or Excel format files.** Click on the **Import** (📁) button. The first two columns in the file containing barcodes must contain the expected barcodes and their names, respectively. Any additional columns will be ignored. An acceptable CSV formatted file could look like:
 "AAAAAA", "Sample1"
 "GGGGGG", "Sample2"
 "CCCCCC", "Sample3"

A preview of results (figure 28.21) based on 10,000 reads is presented. With a single input, the preview is based on the first 10,000 reads. When multiple inputs are provided, the 10,000 reads are taken from across the inputs, with the contribution from each input being proportional to the relative size of that input.

If you would like to change the name of the barcode(s), this can be done by double-clicking on the specific name that you would like to change, see figure 28.23.

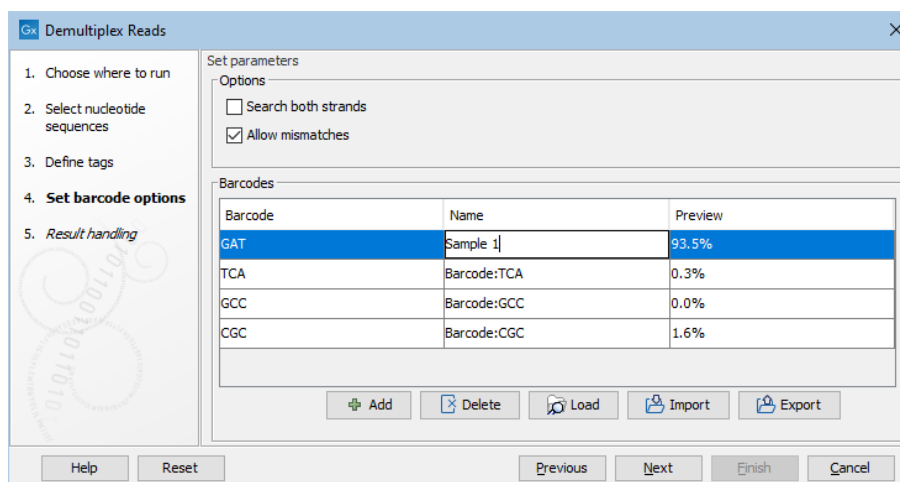


Figure 28.23: The sequence can be renamed by double-clicking on the name.

28.3.2 Output from Demultiplex Reads

The outputs from **Demultiplex Reads** are (see figure 28.24):

- Sequence lists containing the demultiplexed reads, one for each barcode with reads associated with it.
- A sequence list containing the reads without a match to any barcode. The name of this element ends in 'Not grouped'. This output is optional.
- A report summarizing the number of reads identified for each barcode and the number without a match to any barcode ('Not grouped') (figure 28.25). This output is optional.

There is also an option to create subfolders for each sequence list. This can be useful if downstream analyses will be run in batch mode, see section 12.3.

A new sequence list will be generated for each barcode for which reads have been identified, containing all the sequences where this barcode is identified. Both the linker and barcode sequences are removed from each of the sequences in the list, so that only the target sequence remains. This means that you can continue the analysis by doing trimming or mapping. Note that you have to perform separate mappings for each sequence list.

28.3.3 Running Demultiplex Reads in workflows

A number of sequencing kits offer indexing kits where barcodes are coded into plate wells, e.g. the QIAseq UPX sequencing kits. Often each individual well has a specific barcode that is always placed in the plate matrix at the same position. Usually different plates are available and plexing can be done for up to 386 wells (4 x 96 different plate well designs). When running these protocols demultiplexing directly on the sequencing machine is not appropriate and hence the demultiplexing step needs to be done using the **Demultiplex Reads** tool instead.

When analyzing multiple samples it is not always that the full 96 samples plate is used. This result in empty wells that needs to be removed. When running the **Demultiplex Reads** tool in a workflow you get the option to load or import your barcode table, have a preview of what wells

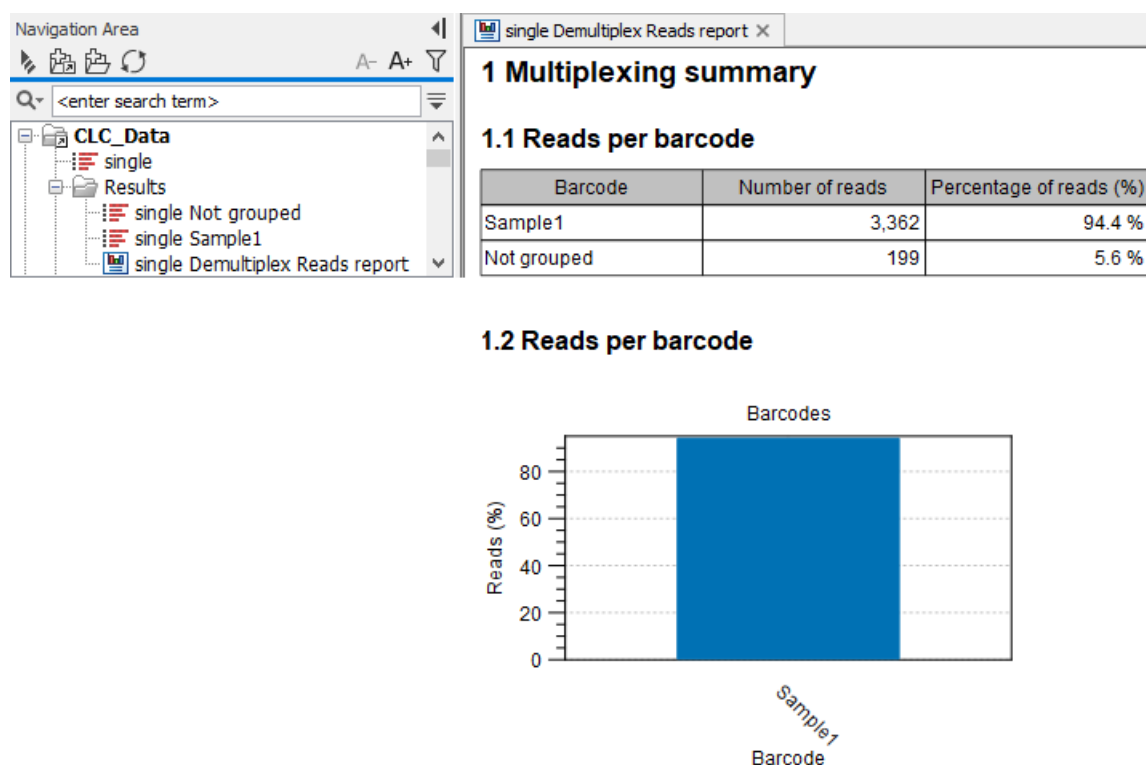


Figure 28.24: Outputs from Demultiplex Reads where the input sequence list named 'single' contained reads matching one barcode CCT named 'Sample1'. Those reads are output in the list named 'single Sample1'. Reads not grouped with any barcode are in the list called 'single Sample1'. The report contains a summary of the number of reads in each of the sequence lists.

have been used and you are given the option to remove the barcodes (wells in the plate) that have not been used, or that for some reason did not produce any or only few reads.

An example of a plate design as well as the barcode preview is shown in figure 28.26 and figure 28.27. A **Delete** button is present and will remove highlight barcodes.

Use the preview functionality to identify barcodes matching reads. When the Preview column is empty you can fill in the percentages by selecting the preview button and the selecting the reads. Use sorting by ticking the Preview column header to rank the barcodes according to percent reads per barcode. It will now be easy to select wells that are not used or have low quality.

You can also just deselect any barcode/well that was not used. For this sorting on plate column number and plate row number as illustrated in figure 28.27 could be helpful.

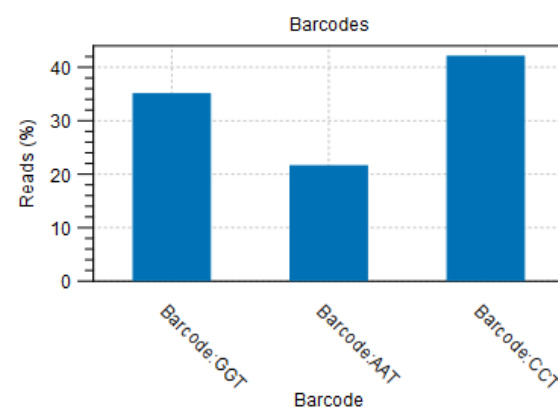
Note: When reads output from **Demultiplex Reads** in a workflow should be further processed, an Iterate element must be connected to the 'Demultiplexed Reads' output channel. The next downstream analysis element should be connected to the output channel of that Iterate element. Each set of reads is then processed individually, either until the workflow ends or until a Collect and Distribute element is encountered.

1 Multiplexing summary

1.1 Reads per barcode

Barcode	Number of reads	Percentage of reads (%)
Barcode:GGT	623,730	35.1 %
Barcode:AAT	385,305	21.7 %
Barcode:CCT	747,915	42.1 %
Not grouped	17,561	1.0 %

1.2 Reads per barcode



1.3 Unmatched barcodes

Barcode:CGT

Figure 28.25: An example of a report showing the number of reads in each group. In this example four different barcodes were used to separate four different samples. No reads have been identified for one of the provided barcodes.

	1	2	3	4	5	6	7	8	9	10	11	12
A	1	9	17									
B	2	10	18									
C	3	11	19									
D	4	12	20									
E	5	13	21									
F	6	14	22									
G	7	15	23									
H	8	16	24									

Figure 28.26: The sample design organisation of an experiment not using all the wells in the plate.

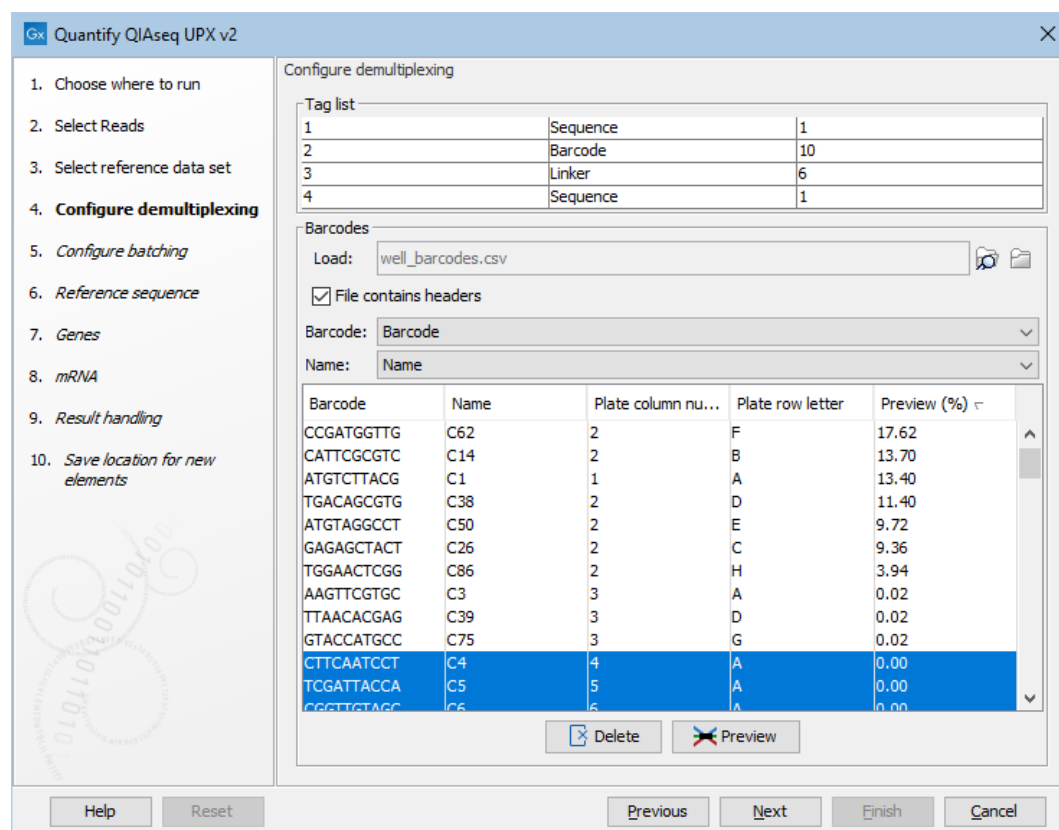


Figure 28.27: A preview of the barcode table that was loaded. Barcodes can be deleted by selecting them, as shown here, and clicking on the Delete button.

Chapter 29

Quality control for resequencing analysis

Contents

29.1 QC for Targeted Sequencing	766
29.1.1 Coverage summary report	769
29.1.2 Per-region statistics	773
29.1.3 Coverage table	775
29.1.4 Coverage graph	775
29.1.5 Gene coverage	776
29.2 Target Region Coverage Analysis	777
29.2.1 Output from Target Region Coverage Analysis	778
29.3 QC for Read Mapping	780
29.3.1 References	781
29.3.2 Mapped read statistics	783
29.3.3 Statistics table for each mapping	785
29.4 Whole Genome Coverage Analysis	786

29.1 QC for Targeted Sequencing


Given an annotation track with the target regions (for example imported from a bed file), this tool will investigate a read mapping to determine whether the targeted regions have been appropriately covered by sequencing reads. It will also give information about how specific the reads map to the targeted regions. The results are provided both as a summary report and as track or table with detailed information about each targeted region.

To create the target regions statistics:

Tools | Quality Control  | **QC for Targeted Sequencing** ()

This opens a wizard where you can select mapping results ()/ ()/ () as seen in figure 29.1.

Clicking **Next** will take you to the wizard shown in figure 29.2.

Click the **Browse** () icon to select an annotation track that defines the targeted regions of your reference genome. You can either import the target regions as an annotation file (see section

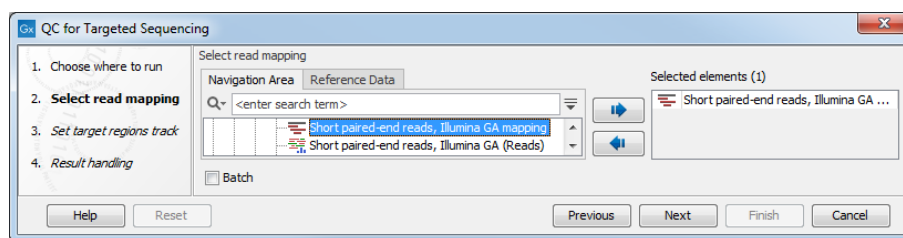


Figure 29.1: Select a read mapping.

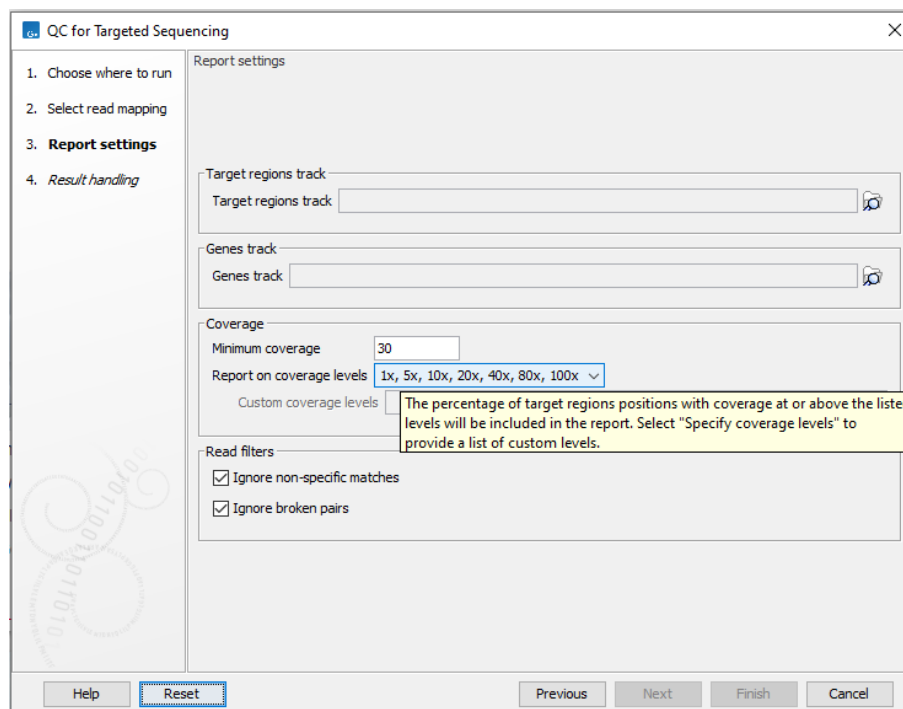


Figure 29.2: Specifying the track of target regions.

7.2) or convert (see section 27.8) from annotations on a reference genome that is already stored in the **Navigation Area**.

If you provide a **Genes track**, coverage metrics will also be calculated for each gene that overlaps one or more of the target regions.

Under **Coverage** you can provide a **Minimum coverage** threshold, i.e., the minimum coverage needed on all positions in a target, in order for that target to be considered covered.

The **Report on coverage levels** allows you, via a drop-down list, to select different sets of predefined coverage thresholds to use for reporting or to specify you own customized list by selecting **Specify coverage levels** as shown in figure 29.3. By selecting Specify coverage levels you get the option to add a list of comma-separated custom coverage levels. As shown in figure 29.3 you will get a warning if the Custom coverage levels field is blank and you will not be able to move on to the next wizard step before you have provided custom coverage levels.

Custom coverage levels must be comma-separated and specified either as plain numbers (20, 30, 40) or in the format 20x, 30x, 40x as shown in figure 29.4.

Finally, you are asked to specify whether you want to **Ignore non-specific matches** and **Ignore broken pairs**. When these are applied reads that are non-specifically mapped or belong to broken

QC for Targeted Sequencing

1. Choose where to run
2. Select read mapping
3. **Report settings**
4. Result handling
5. Save location for new elements

Report settings

Target regions track
Target regions track

Genes track
Genes track

Coverage
Minimum coverage: 30
Report on coverage levels: Specify coverage levels
Custom coverage levels

The percentage of target regions positions with coverage at or above the listed levels will be included in the report. Select "Specify coverage levels" to provide a list of custom levels.

Read filters
☒ Ignore non-specific matches
☒ Ignore broken pairs

Invalid input: Custom coverage levels requires at least one value

Help Reset Previous Next Finish Cancel

Figure 29.3: Selecting *Specify coverage levels* from the drop-down list will allow you to add your own custom coverage levels in the text field by typing in the desired coverage levels. Numbers should be comma-separated.

pairs will be ignored.

Click **Next** to specify the type of output you want (see figure 29.5).

There are five options:

- The report gives an overview of the whole data set as explained in section 29.1.1.
- The track gives information on coverage for each target region as described in section 29.1.2.
- The coverage table outputs coverage for each position in all the targets as described in section 29.1.3.
- The coverage graph outputs a graphical presentation of the coverage for each position in all the targets. Positions outside the targets will have the value 0. The values are calculated by the "Target regions statistics" tool - that is, where broken pairs and multi-hit reads are included or ignored, depending upon what the user has specified in the wizard. On the x-axis is the reference position; on the y-axis is the coverage. The x-axis and y-axis values are identical to those found in the corresponding columns of the coverage table.
- The gene coverage track gives information on coverage for each gene overlapping one or more target regions.

Click **Finish** to create the selected reports.

Figure 29.4: When adding a list of custom coverage levels, numbers should be comma-separated and provided in the format 20, 30, 40 or 20x, 30x, 40x.

Figure 29.5: Specifying how the result should be reported.

29.1.1 Coverage summary report

An example of a coverage report is shown in figure 29.6.

This figure shows only the top of the report. The full content is explained below:

- **Target regions**

- **Summary** This table shows overall coverage information.

1 Target regions

1.1 Summary

Number target regions	5,199
Total length of targeted regions	838,627
Minimum coverage	0
Maximum coverage	1,579
Average coverage	496.1
Median coverage	471.0
Number of target regions with coverage < 30	39
Total length of target regions containing positions with coverage < 30	7,078
Total length of target region positions with coverage < 30	5,441
Total length of target region positions with coverage ≥ 30	833,186
Percentage of target region positions with coverage ≥ 30 (%)	99.4

1.2 Fractions of targets with coverage at least 30

Number of targeted regions for which	Count	Percentage
≥ 100% of the targeted region has coverage at least 30	5,160	99.25
≥ 90% of the targeted region has coverage at least 30	5,162	99.29
≥ 80% of the targeted region has coverage at least 30	5,165	99.35
≥ 70% of the targeted region has coverage at least 30	5,169	99.42
≥ 60% of the targeted region has coverage at least 30	5,169	99.42
≥ 50% of the targeted region has coverage at least 30	5,170	99.44
≥ 40% of the targeted region has coverage at least 30	5,170	99.44
≥ 30% of the targeted region has coverage at least 30	5,171	99.46
≥ 20% of the targeted region has coverage at least 30	5,172	99.48
≥ 10% of the targeted region has coverage at least 30	5,173	99.50
≥ 0% of the targeted region has coverage at least 30	5,199	100.00

Figure 29.6: The report with overviews of mapped reads.

- * **Number target regions.** Number of targeted regions.
- * **Total length of target regions.** Sum of the size of all the targeted regions (this means it is calculated from the annotations alone and is not influenced by the reads).
- * **Minimum coverage.** The minimum observed coverage on a single position.
- * **Maximum coverage.** The maximum observed coverage on a single position.
- * **Average coverage.** For each position in each target region the coverage is calculated and stored: you can see the individual coverages in the **Coverage table** output, figure 29.11). The average coverage is calculated by taking the mean of all the calculated coverages in all the positions in all target regions. Note that if the user has chosen the Read filters options "Ignore non-specific matches" or "Ignore broken pairs", these reads will not contribute to the coverage. Note also that bases in overlapping paired reads will only be counted as 1.
- * **Median coverage.** Is calculated by taking the median of all the calculated

coverages in all the positions in all target regions. As specified above, if the user has chosen the Read filters options "Ignore non-specific matches" or "Ignore broken pairs", these reads will not contribute to the coverage. Note also that bases in overlapping paired reads will only be counted as 1.

- * **Number of target regions with coverage below x.** Number of target regions which have positions with a coverage that is below the user-specified "Minimum coverage" threshold.
- * **Total length of target regions containing positions with coverage below x.**
- * **Total length of target regions with a coverage below x.**
- **Fractions of targets with coverage at least...** A table and a histogram show how many target regions have a certain percentage of the region above the user-specified **Minimum coverage** threshold.
- **Coverage of target regions positions** A first plot shows the coverage level on the x axis, and the number of positions in the target regions with that coverage level. Below is a version of the histogram above zoomed in to the values that lie \pm 3SDs from the median.
- **Minimum coverage of target region positions** This shows the percentage of the targeted regions that are covered by this many bases. The intervals can be specified in the dialog when running the analysis. Default is 1, 5, 10, 20, 40, 80, 100 times. In figure 29.7 this means that 26.58 % of the positions on the target are covered by at least 40 bases.

1.5 Minimum coverage of target region positions

Coverage	
1 x	95.06%
5 x	89.83%
10 x	82.40%
20 x	62.40%
40 x	26.58%
80 x	4.78%
100 x	2.34%

Figure 29.7: Minimum coverage of target region positions of the report.

- **Gene coverage** A per gene table listing the number of target regions in the gene, the number of bases, the percentage of bases passing the coverage thresholds and the median coverage.
- **Targeted region overview**
This section contains two tables: one that summarizes, for each reference sequence, information relating to the *reads* mapped, and one that summarizes, for each reference, information relating to the *bases* mapped (figure 29.8).
Note that, for the table that is concerned with *reads*, reads in overlapping pairs are counted individually. Also note that, for the table that is concerned with *bases*, bases in overlapping paired reads are counted only as one (Examples are given in figures 29.9 and figure 29.10).
 - **Reference** The name of the reference sequence.

2 Targeted region overview

Reference	Total mapped reads	Mapped reads in targeted region	Specificity (%)	Total mapped reads excl ignored	Mapped reads in targeted region excl ignored	Specificity excl ignored (%)
1	4,338,296	2,168,439	49.98	4,338,296	2,168,439	49.98
2	4,929,239	2,384,422	48.37	4,929,239	2,384,422	48.37
3	2,495,513	1,242,395	49.79	2,495,513	1,242,395	49.79
4	2,214,025	768,447	34.71	2,214,025	768,447	34.71

Reference	Total mapped bases	Mapped bases in targeted region	Specificity (%)	Total mapped bases excl ignored	Mapped bases in targeted region excl ignored	Specificity excl ignored (%)
1	344,955,568	252,700,911	73.26	344,955,568	252,700,911	73.26
2	388,499,642	276,219,534	71.10	388,499,642	276,219,534	71.10
3	196,369,744	144,978,039	73.83	196,369,744	144,978,039	73.83
4	173,524,865	91,208,669	52.56	173,524,865	91,208,669	52.56

Figure 29.8: Targeted regions overview of the report: mapped bases.

- **Total mapped reads/bases** The total number of mapped reads/bases on the reference, including reads mapped outside the target regions.
- **Mapped reads/bases in targeted region** Total number of reads in the targeted regions. Note that if a read is only partially inside a targeted region, it will still count as a full read.
- **Specificity** The percentage of the total mapped reads/bases that are in the targeted regions.
- **Total mapped reads/bases excl ignored** The total number of mapped reads/bases on the reference, including reads/bases mapped outside the target regions, excluding the non-specific matches or broken pairs (or the bases in non-specific matches or broken pairs), if the user has enabled the option to ignore those.
- **Mapped reads/bases in targeted region excl ignored** Total number of reads/bases in the targeted regions, excluding the non-specific matches or broken pairs (or the bases in non-specific matches or broken pairs), if the user has enabled the option to ignore those.
- **Specificity excl ignored** The percentage of the total mapped reads/bases that are in the targeted regions.

In addition, two plots called **Distribution of target region length** display the length of the target regions for all regions, and the second one where only the target region lengths that lie within +3SDs of the median target length are shown.

- **Base coverage relative to mean coverage**

- **Base coverage** The percentage of base positions in the target regions that are covered by respectively 0.1, 0.2, 0.3, 0.4, 0.5 and 1.0 times the mean coverage, where the mean coverage is the **average coverage** given in table 1.1. Because this is based on

mean coverage, the numbers can be used for cross-sample comparison of the quality of the experiment.

- **Base coverage plot** A plot showing the relationship between fold mean coverage and the number of positions. This is a graphical representation of the **Base coverage** table above.


- **Mean coverage per target**

Three plots listing the mean coverage for each position of the targeted regions. The first plot shows coverage across the whole target, using a percentage of the target length on the x axis (to make it possible to have targets with different lengths in the same plot). This is reported for reverse and forward reads as well. In addition, there are two plots showing the same but with base positions on the x axis counting from the start and end of the target regions, respectively. These plots can be used to evaluate whether there is a general tendency towards lower coverage at the end of the targeted region, and whether there is a bias in terms of forward and reverse reads coverage.

- **Read count per %GC**

The plot shows the GC content of the reference sequence on the X-axis and the number of mapped reads on the Y-axis. This plot will show if there is a basis caused by higher GC-content in the sequence.

29.1.2 Per-region statistics

In addition to the summary report, you can see coverage statistics for each targeted region. This is reported as a track, and you can see the numbers by going to the table  view of the track. An example is shown in figure 29.9:

- **Chromosome** The name is taken from the reference sequence used for mapping.
- **Region** The targeted region.
- **Name** The annotation name derived from the annotation (if there is additional information on the annotation, this is retained in this table as well).
- **Target region length** The length of the region.
- **Target region length with coverage above...** The length of the region that is covered by at least the **Minimum coverage**.
- **Percentage with coverage above...** The percentage of the positions in the region with coverage at least the **Minimum coverage**.
- **Read count** Number of reads that cover this region. Note that reads that only cover the region partially are also included. Note that reads in overlapping pairs are counted individually (see figures 29.9 and figure 29.10).
- **Broken read count** Number of broken reads that cover this region. Reads are counted in the same way as described for the **Read count**.
- **Non-specific read count** Number of non-specific reads that cover this region. Reads are counted in the same way as described for the **Read count**.

- **Base count** The number of bases in the reads that are covering the target region. Note that bases in overlapping pairs are counted only once (see figures 29.9 and figure 29.10).
- **%GC** The GC content of the region.
- **Min, Max, Mean and Median coverage** Lowest, highest, average and median coverage in the region, respectively.
- **Zero coverage bases** Number of positions with no coverage.
- **Mean and median coverage (excluding zero coverage)** The average and median coverage in the region, excluding any zero-coverage parts.

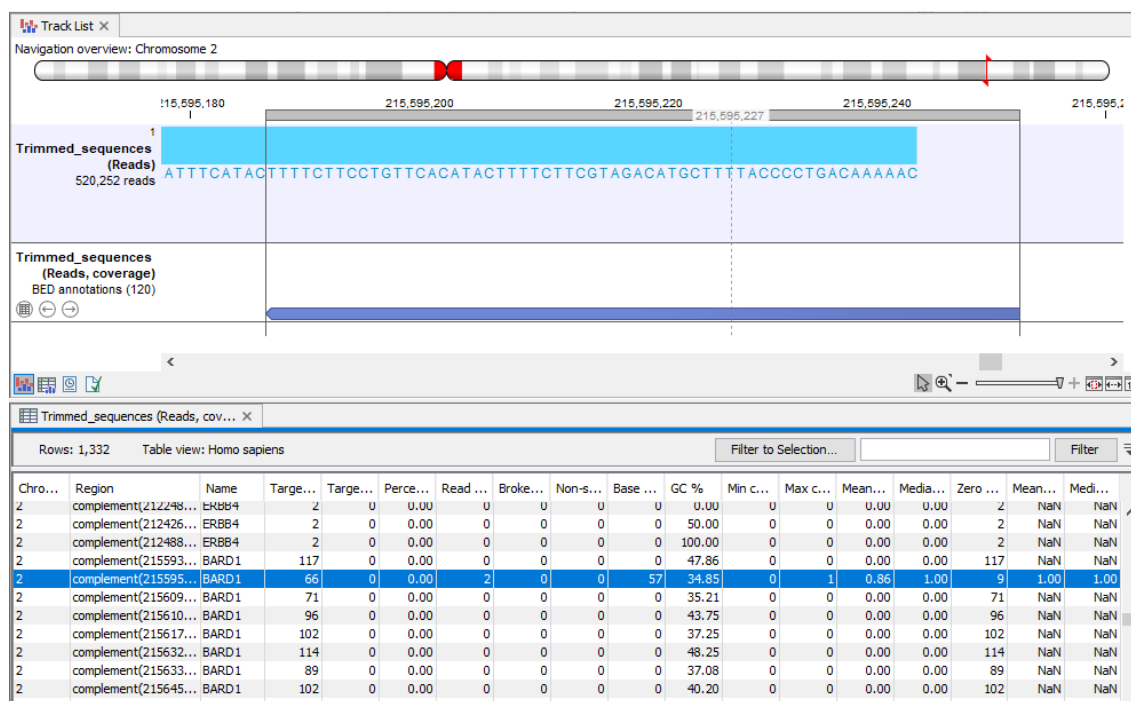


Figure 29.9: A track list containing the target region coverage track and reads track. The target region coverage track has been opened from the track list and is shown in table view. Detailed information on each region is displayed. Only one paired read maps to the region selected.

In the figure, the coverage table and a track list are shown in a split view. When opened in a split view, the two views are linked, that is, clicking on an entry in one view moves the focus in the other view to the relevant item or table row. Creating track lists and opening tracks in linked views is described in section 27.2.

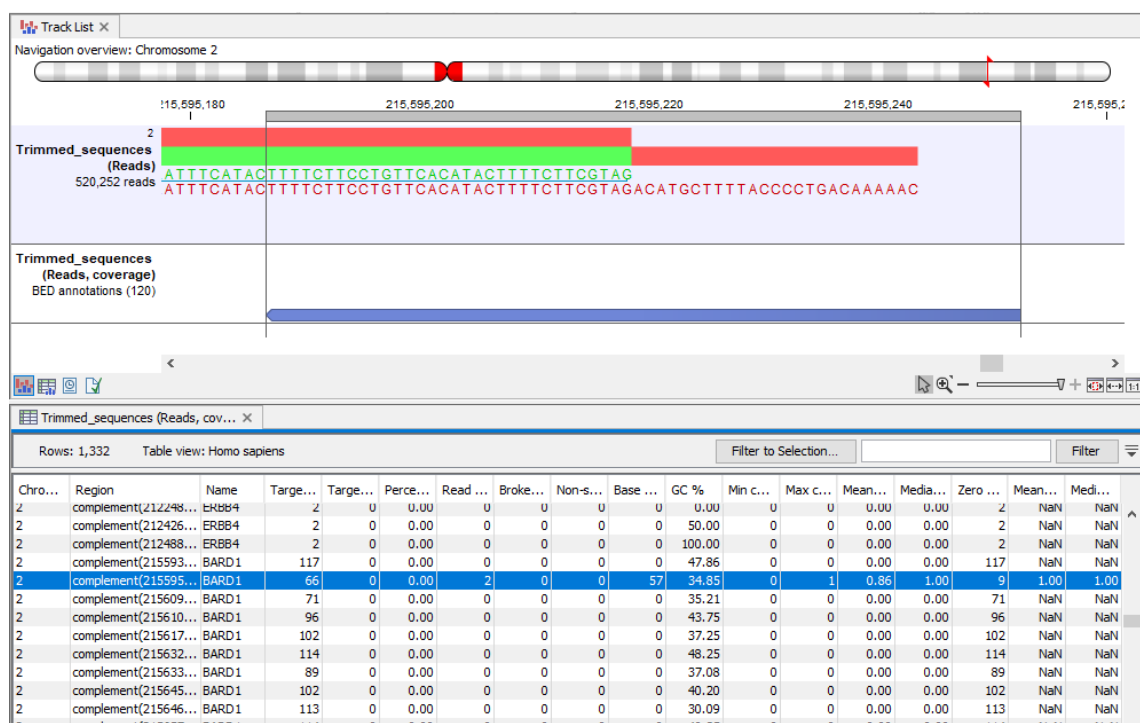


Figure 29.10: The same data as shown in figure 29.9, but now the Show strands of paired reads option in the side-panel of the reads track has been ticked, so that the two reads in the paired read are shown.

29.1.3 Coverage table

Besides standard information such as position etc, the coverage table (figure 29.11) lists the following information for each position in the whole target:

- **Name** The name of the target region.
- **Target region position** The name of the target region.
- **Reference base** The base in the reference sequence.
- **Coverage** The number of bases mapped to this position. Note that bases in overlapping pairs are counted only once. Also note that if the user has chosen the **Ignore non-specific matches** or **Ignore broken pairs** options, these reads will be ignored. (see discussion on coverage in section 29.3.1).

In the figure, the coverage table and a track list are shown in a split view. When opened in a split view, the two views are linked, that is, clicking on an entry in one view moves the focus in the other view to the relevant item or table row. Creating track lists and opening tracks in linked views is described in section 27.2.

29.1.4 Coverage graph

The coverage graph is a graphical presentation of the coverage for each position in all the targets (positions outside the targets will have the value 0). The values are calculated by the "Target regions statistics" tool, and are presented with the reference position on the x-axis and the

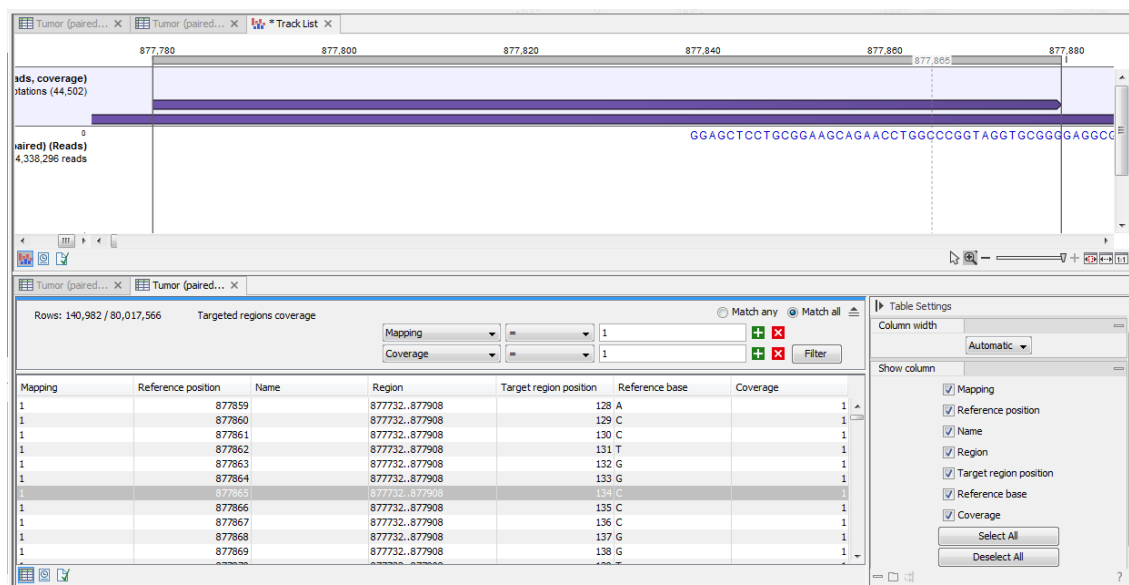


Figure 29.11: The targeted region coverage table for the same region as shown in same as shown in figures 29.9 and figure 29.10.

coverage on the y-axis (see figure 29.12). The x-axis and y-axis values are identical to those found in the columns of the coverage table.

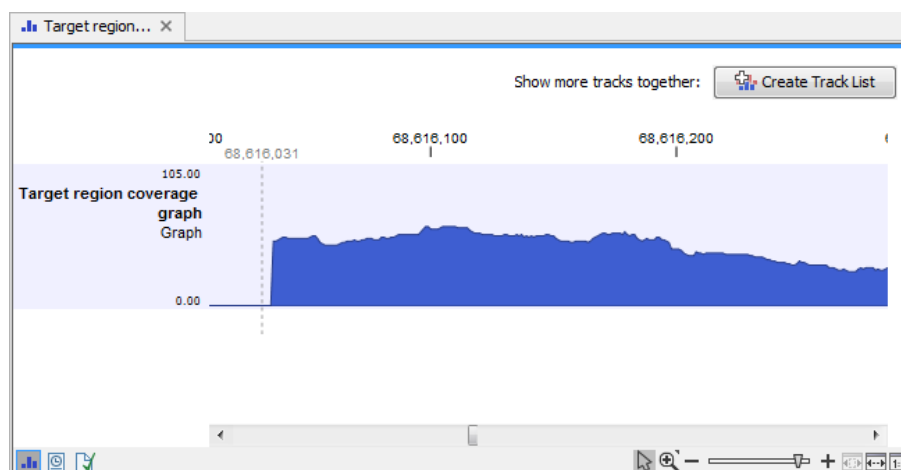


Figure 29.12: An example of a targeted region coverage graph.

29.1.5 Gene coverage

In the gene coverage track you can see coverage statistics for each gene (figure 29.13). You can see the numbers by going to the table (📊) view of the track.

- **Chromosome** The name is taken from the reference sequence used for mapping.
- **Region** The gene region.
- **Gene specific columns** A number of columns from the Genes track used as input to **QC for Targeted Sequencing**. These can for example be name, source and gene_version.

- **Targets** The number of targets in the gene.
- **Target bases** The number of bases in the target regions in the gene.
- **Target \geq coverage (%)** The percentage of bases in targets with coverage above the threshold.

Rows: 331 Table view: Homo sapiens

Chromos...	Region	Name	Targets	Target bases	Target $\geq 100x$ (%)	Target $\geq 80x$ (%)
1	complement(5922871..6052533)	NPHP4	30	4614	100.00	100.00
1	complement(6526152..6580121)	PLEKHG5	25	3699	98.97	98.97
1	10270863..10441661	KIF1B	49	7426	100.00	100.00
1	complement(11845780..11866977)	MTHFR	12	2410	100.00	100.00
1	11994262..12035595	PLOD1	20	2525	100.00	100.00
1	complement(17345217..17380665)	SDHB	8	923	100.00	100.00
1	23037332..23241818	EPHB2	18	3397	99.41	99.59
1	complement(38076951..38100595)	RSPO1	6	865	100.00	100.00
1	complement(40538379..40563375)	PPT1	9	1233	100.00	100.00
1	complement(43212006..43232755)	LEPRE1	15	2721	100.00	100.00
1	complement(45285516..45308735)	PTCH2	22	3932	97.91	97.91
1	45477819..45481247	LRN1	10	1223	100.00	100.00

Figure 29.13: An example of a gene coverage track.

29.2 Target Region Coverage Analysis


The **Target Region Coverage Analysis** tool makes it easy to evaluate and compare multiple samples with respect to a given coverage metric. The tool takes as input one or more per-region statistics tracks generated by **QC for Targeted Sequencing** and outputs a target region track providing statistics across the analyzed samples. In addition, an overlay annotation track (for example a gene track) can be provided to obtain a higher-level summary, where target regions are grouped based on overlap, and coverage statistics are calculated for each group.

The **QC for Targeted Sequencing** tool is described in section [29.1](#)

Running the tool

To launch **Target Region Coverage Analysis**, go to:

Tools | Quality Control  **| Target Region Coverage Analysis** 

In the first dialog (figure [29.14](#)), select one or more per-region statistic tracks  produced by **QC for Targeted Sequencing**. The tracks must be based on the same target region track.

The next dialog allows you to configure the settings for this tool, as shown in figure [29.15](#) and described below.

- **Metric:** Metric column from the per-region statistics tracks for which the QC evaluation will be performed. The available metrics are: GC %, Min coverage, Max coverage, Mean coverage, Median coverage, Mean coverage (excluding zero coverage) and Median coverage (excluding zero coverage).
- **Minimum threshold, individual values:** Minimum threshold for the metric selected above. Each target region in each sample is evaluated separately, and must have at least this value to pass. Values that do not pass this criteria will be highlighted in the table view of the target region output track.

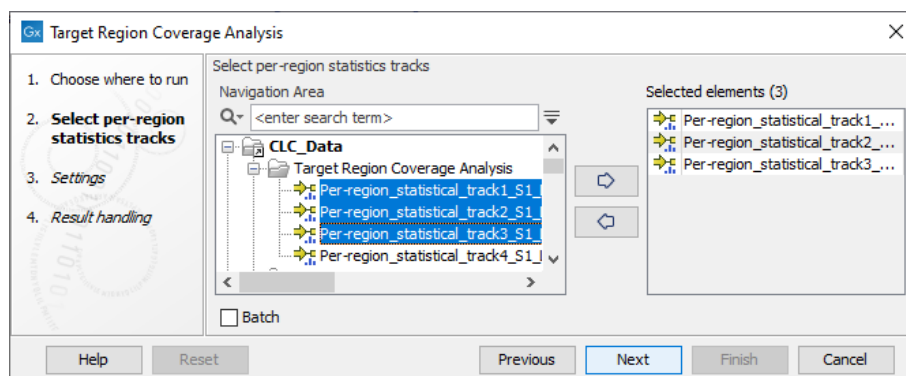


Figure 29.14: Select one or more per-region statistics tracks.

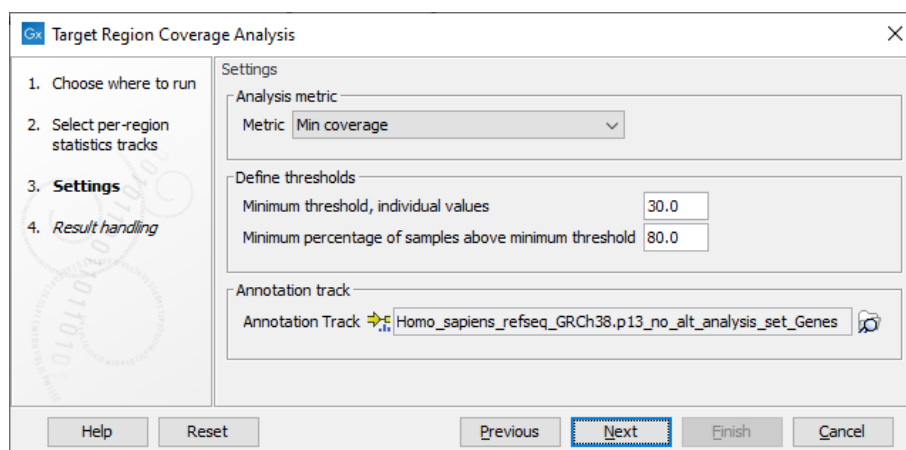


Figure 29.15: Settings of the Target Region Coverage Analysis tool.

- **Minimum percentage of samples above minimum threshold:** The percentage of input samples that should pass the minimum threshold. The value is used to color code the target region output track and make it easier to see which target regions that pass the criteria across samples.
- **Annotation track:** The annotation track (📌) is optional and can be a gene, CDS or mRNA track. If provided, an additional output is produced in which target regions are grouped based on the overlapping annotations. For example, if a gene track is selected, target regions are grouped per gene and the selected metric is combined and reported per gene.

29.2.1 Output from Target Region Coverage Analysis

Two outputs are produced from the **Target Region Coverage Analysis** tool:

- **Target region coverage track:** Target regions annotated with coverage metrics from the individual samples and statistics across all samples. In the table view, fields for which values did not pass the defined threshold will be highlighted. This makes it possible to quickly spot both poor samples that have multiple failing targets and poor target regions that fail across samples. The latter may be indicative of failing primers.
- **Annotation coverage track:** This output is produced only if an annotation track is provided.

The track table view lists cross-sample statistics for each annotation (e.g. gene) that have at least one overlapping target region. Annotations with no overlapping target region are not displayed.

Target region coverage track

The target region coverage track includes the following annotations:

- **Target region length:** Length of the target region.
- **Metric, min:** Sample minimum of the selected metric observed for this target region.
- **Metric, max:** Sample maximum of the selected metric observed for this target region.
- **Metric, mean:** Sample mean of the selected metric observed for this target region.
- **Metric, median:** Sample median of the selected metric observed for this target region.
- **Metric, std dev:** Sample standard deviation of the selected metric observed for this target region.
- **Percentage of samples passing threshold:** Percentage of samples for which the metric is equal to or above the threshold.
- **Individual per-region statistics track metrics:** One column per input track with the individual sample metrics.
- **Annotation column:** Overlapping annotation (Gene, CDS, mRNA). This column is only present if an annotation track was provided.

Annotation coverage track The annotation coverage track provides combined statistics for target regions overlapping the same annotation region. If the target regions correspond to exons and a gene track is selected as annotation track, all exons within a gene are combined and statistics are reported per gene. For each sample, the metric values from overlapping target regions are combined to a single metric value. The selected metric dictates how values are combined: Min coverage values are combined by taking the minimum, Max coverage values are combined by taking that maximum and Mean coverage and GC% values are combined as a weighted average, where each target region is weighted by its length. Median coverage values are combined by calculating the median of the values, however, it should be noted that this is different from calculating the median of all base position coverage values contained in the set of target regions.

The annotation coverage track includes the following annotations:

- **Target regions:** Number of overlapping target regions.
- **Target region length:** Cumulative length of overlapping target regions.
- **Metric, min:** Sample minimum of the selected metric observed for this annotation region.
- **Metric, max:** Sample maximum of the selected metric observed for this annotation region.
- **Metric, mean:** Sample mean of the selected metric observed for this annotation region.

- **Metric, median:** Sample median of the selected metric observed for this annotation region.
- **Metric, std dev:** Sample standard deviation of the selected metric observed for this annotation region.
- **Individual per-annotation statistics track metrics:** One column per input track with the individual sample metrics.

29.3 QC for Read Mapping

Note that this tool can be used to create a detailed report on a read mapping or a de novo assembly.

To create a detailed mapping report, go to:

Tools | Quality Control (📁) | QC for Read Mapping (📊)

This opens a dialog where you can select mapping results (📄)/ (📄)/ (📄), simple contigs from a de novo assembly, or RNA-Seq analysis results (📄).

In the next wizard window "Set contig group" (figure 29.16), you can set thresholds for grouping long and short contigs. Inputs for which thresholds can be specified are simple contigs or stand-alone mappings generated by a tool that works with de novo assemblies. These options are disabled and greyed out if you are working with a read mapping or an RNA-Seq analysis mapping.

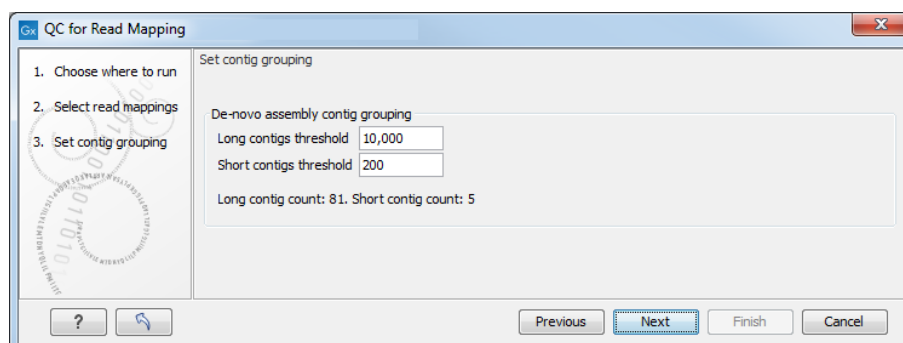


Figure 29.16: Parameters for mapping reports.

The grouping is used to show statistics (e.g., number of contigs, mean length) for the contigs in each group. Note that the de novo assembly in the *CLC Genomics Workbench* per default only reports contigs longer than 200 bp (this can be changed when running the assembly).

In the last dialog (figure 29.17), by checking "Create table with statistics for each mapping", you can create a table showing detailed statistics for each reference sequence (for de novo results the contigs act as reference sequences, so it will be one row per contig).

The first section of the detailed mapping report is a summary of the statistics:

- Reference count
- Type
- Total reference length

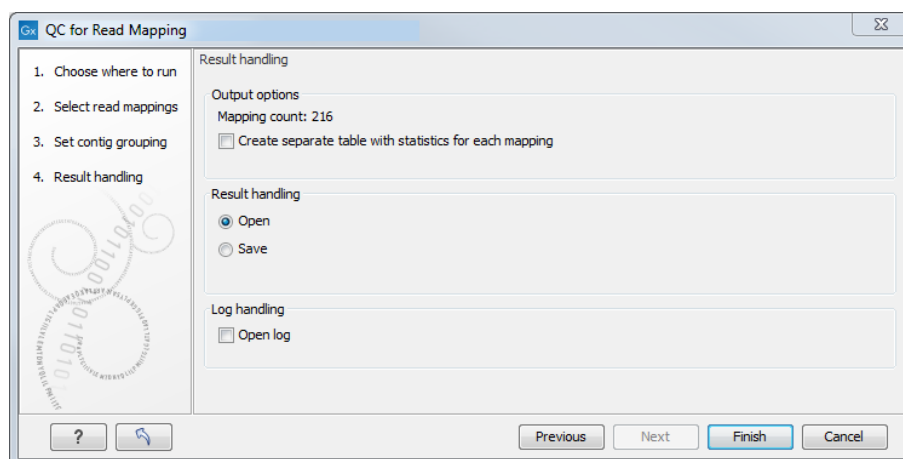


Figure 29.17: Result handling options.

- GC contents in %
- Total read count
- Mean read length
- Total read length

The rest of the report, as well as the optional statistic tables are described in the following sections.

29.3.1 References

The second section of the detailed report concerns the Reference sequence(s).

First, a table gives information about **Reference coverage**, including coverage statistics and GC content of the reference sequence.

The second table gives **Coverage statistics**. A position on the reference is counted as "covered" when at least one read is aligned to it. Note that unaligned ends (faded nucleotides at the ends) that are produced when mapping using local alignment do not contribute to the coverage. Also, positions with an ambiguous nucleotide in the reference (i.e., not A, C, T or G) count as zero coverage regions, regardless of the number of reads mapping across them.

In the example shown in figure 29.18, there is a region of zero coverage in the middle and one time coverage on each side. Note that the gaps to the very right are within the same read which means that these two positions on the reference sequence are still counted as "covered".

In this table, coverage is reported on two levels: including and excluding zero coverage regions. In some cases, you do not expect the whole reference to be covered, and only the coverage levels of the covered parts of the reference sequence are interesting. On the other hand, if you have sequenced the full genome that you use as reference, the overall coverage is probably the most relevant number (i.e. including zero coverage regions).

In the third and fourth subsections, two graphs display **Coverage level distribution**, with and without zero coverage regions. Two bar plots show the distribution of coverage with coverage level on the x-axis and number of positions with that coverage on the y-axis (as seen in figure 29.19).

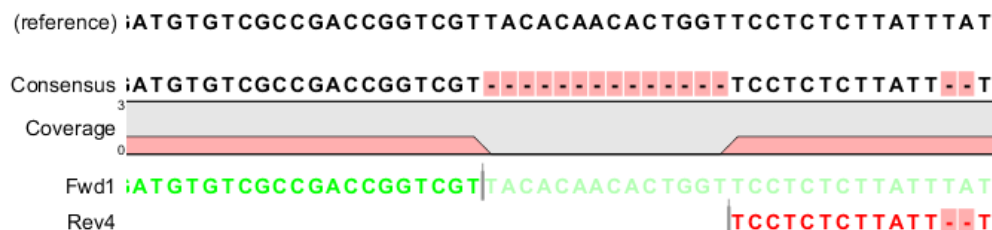


Figure 29.18: A region of zero coverage in the middle and one time coverage on each side. Note that the gaps to the very right are within the same read which means that these two positions on the reference sequence are still counted as "covered".

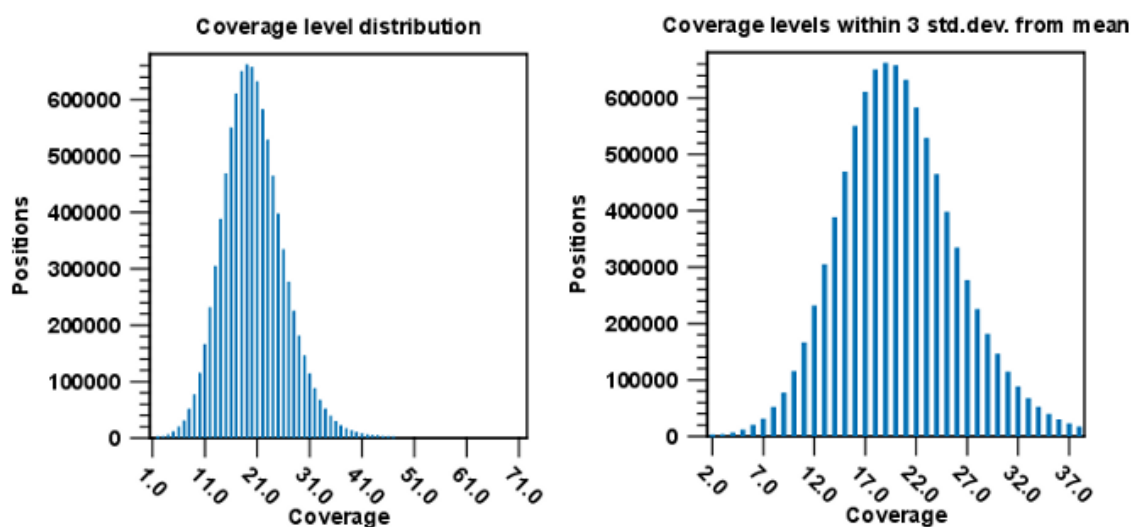


Figure 29.19: Distribution of coverage - to the left for all the coverage levels, and to the right for coverage levels within 3 standard deviations from the mean.

The graph to the left shows all the coverage levels, whereas the graph to the right shows coverage levels within 3 standard deviations from the mean. The reason for this is that for complex genomes, you will often have a few regions with extremely high coverage which will affect the resolution of the graph, making it impossible to see the coverage distribution for the majority of the references. These coverage outliers are excluded when only showing coverage within 3 standard deviations from the mean. Below the second coverage graph there are some statistics on the data that is outside the 3 standard deviations.

Subsection 5 gives some statistics on the **Zero coverage regions**; the number, minimum and maximum length, mean length, standard deviation, and total length.

One of the biases seen in sequencing data concerns GC content. Often there is a correlation between GC content and coverage. In order to investigate this correlation, the report includes in subsection 6 a **Coverage versus GC Content** graph plotting coverage against GC content (see figure 29.20). Note that you can see the GC content for each reference sequence in the table(s) above.

The plot displays, for each GC content level (0-100 %), the mean read coverage of 100bp reference segments with that GC content.

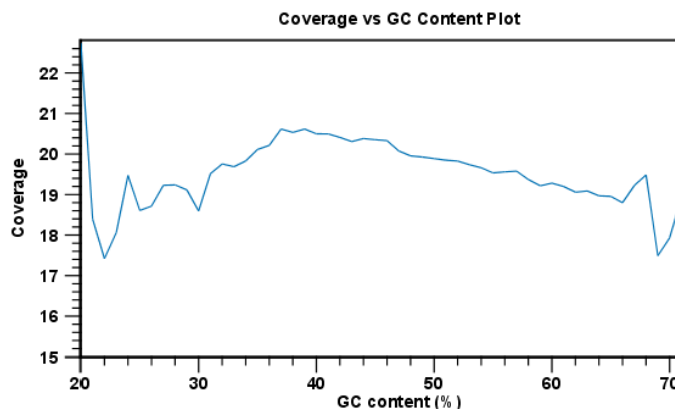


Figure 29.20: The plot displays, for each GC content level (0-100 %), the mean read coverage of 100bp reference segments with that GC content.

For a report created from a de novo assembly, this section finishes with statistics about the reads which are the same for both reference and de novo assembly (see section 29.3.2 below).

29.3.2 Mapped read statistics

This section contains simple statistics for **all mapped reads**, **non-specific matches** (reads that match more than place during the assembly), **non-perfect matches** (reads with one or more mismatches or gaps relative to the reference sequence) and **paired reads**.

Note! Paired reads are counted as two, even though they form one pair. The section on paired reads also includes information about paired distance and counts the number of pairs that were broken due to:

- Wrong distance: When starting the mapping, a distance interval is specified. If the reads during the mapping are placed outside this interval, they will be counted here.
- Mate inverted: If one of the reads has been matched as reverse complement, the pair will be broken (note that the pairwise orientation of the reads is determined during import).
- Mate on other contig: If the reads are placed on different contigs, the pair will also be broken.
- Mate not matched: If only one of the reads match, the pair will be broken as well.

Each subsection contains a table that recapitulates the read count, % of all mapped reads, mean read length and total read length, and for some sections two graphs showing the distribution of match specificity or the distribution of mismatches.

Note that for the section concerning paired reads (see figure 29.21), the distance includes both the read sequence and the insert between them as explained in section 7.3.10.

The following subsections give graphs showing **read length distribution**, **insertion length distribution**, **deletion length distribution**. Two plots of the distribution of insertion and deletion lengths can be seen in figure 29.22 and figure 29.23.

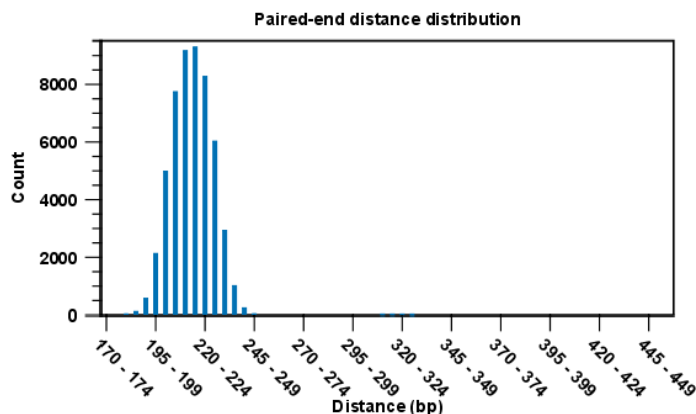


Figure 29.21: A bar plot showing the distribution of distances between intact pairs.

Nucleotide differences in reads relative to a reference gives the percentage of read bases that differ with the reference for all base pairs and a deletion. In the **Nucleotide mapping** section two tables give the counts and percentages of differences between the reads and the reference for each base. Graphs display the relative errors and errors counts between reads to reference and reference to reads, i.e., which bases in the reference are substituted to which bases in the reads. This information is plotted in different ways with an example shown here in figure 29.22.

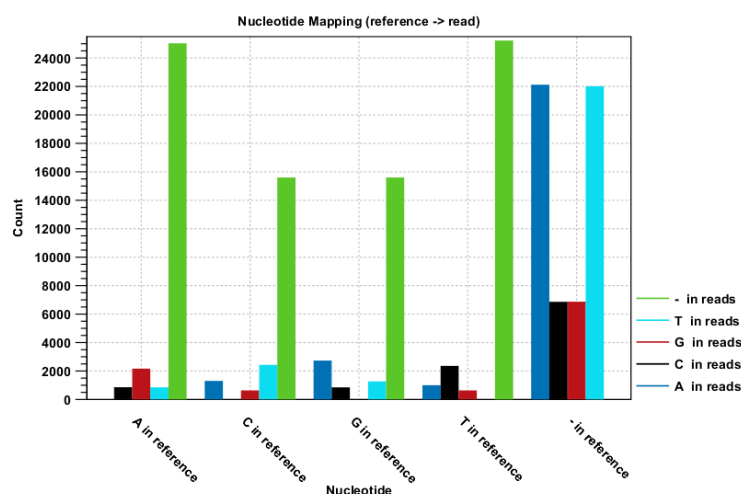


Figure 29.22: The As and Ts are more often substituted with a gap in the sequencing reads than C and G.

This example shows for each type of base in the reference sequence, which base (or gap) is found most often. Please note that only mismatches are plotted - the matches are not included. For example, an A in the reference is more often replaced by a G than any other base.

Below these plots, there are two plots of the **quality values for matches** and **quality values for mismatches**. Next, there is a plot of the mismatch fraction for each read position. Typically with quality dropping towards the end of a read, there will be more mismatches towards the end as the example in figure 29.23 shows.

The last section provides information about **unaligned ends**.

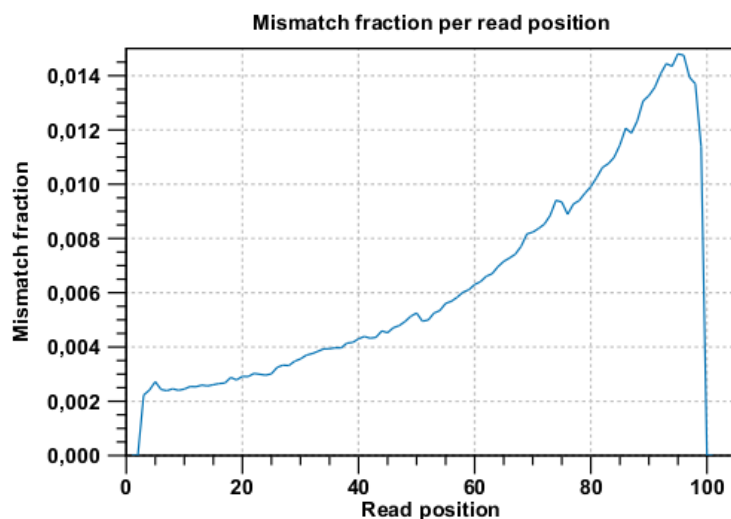


Figure 29.23: There are mismatches towards the end of the reads.

The unaligned ends table contains the following information:

- **Read count** The number of reads with unaligned ends.
- **Read count % of all mapped reads** Percent of mapped reads that have an unaligned end.
- **Positions covered** The number of positions where an unaligned end starts. If multiple unaligned ends start at the same position, these are only counted once.
- **Positions covered in % of reference** Positions covered / positions in the reference * 100
- **Positions covered in % of bases covered** Positions covered / positions that have one or more mapped reads * 100

The table is followed by two plots providing the lengths of unaligned ends and their counts.

29.3.3 Statistics table for each mapping

By checking "Create table with statistics for each mapping", a table showing detailed statistics for each reference sequence will be generated (figure 29.24).

Rows: 35 Individual contigs							
Contig	Reference name	Reference length	Mapped reads	Forward reads	Reversed reads	Reads in aligned pairs	Reads in br...
Mate-pair trimmed (paired) contig 1 mapping	Mate-pair trimmed (paired) contig 1	598431	581188	290527	290661	468898	78160
Mate-pair trimmed (paired) contig 2 mapping	Mate-pair trimmed (paired) contig 2	662986	759762	380307	379455	617740	102074
Mate-pair trimmed (paired) contig 3 mapping	Mate-pair trimmed (paired) contig 3	1276788	1342117	671880	670227	1093648	184842
Mate-pair trimmed (paired) contig 4 mapping	Mate-pair trimmed (paired) contig 4	1942568	2080079	1038727	1041352	1691292	298464
Mate-pair trimmed (paired) contig 5 mapping	Mate-pair trimmed (paired) contig 5	215874	252017	125918	126099	202202	33458
Mate-pair trimmed (paired) contig 6 mapping	Mate-pair trimmed (paired) contig 6	38537	38875	19380	19495	30344	5052
Mate-pair trimmed (paired) contig 7 mapping	Mate-pair trimmed (paired) contig 7	212765	256420	128226	128194	207388	33970
Mate-pair trimmed (paired) contig 8 mapping	Mate-pair trimmed (paired) contig 8	409328	465796	232601	233195	378052	63940
Mate-pair trimmed (paired) contig 9 mapping	Mate-pair trimmed (paired) contig 9	1383	1881	963	918	798	174
Mate-pair trimmed (paired) contig 10 mapping	Mate-pair trimmed (paired) contig 10	387450	457404	228976	228518	373258	59520
Mate-pair trimmed (paired) contig 11 mapping	Mate-pair trimmed (paired) contig 11	49201	55290	27653	27637	43492	7128
Mate-pair trimmed (paired) contig 12 mapping	Mate-pair trimmed (paired) contig 12	106203	123969	61892	62077	99770	16100
Mate-pair trimmed (paired) contig 13 mapping	Mate-pair trimmed (paired) contig 13	68610	78671	39339	39332	62656	10516

Figure 29.24: Statistics table for a read mapping.

- Contig

- Reference name, Latin name, description, length
- Mapped reads
- Forward and Reverse reads
- Reads in aligned pairs
- Reads in broken pairs: wrong distance or mate inverted, mate on other contig, mate not mapped
- Average distance
- Standard deviation distance. Standard deviation of the mapped pairs distances.
- Non-specific and non-perfect matches
- Minimum, maximum, average coverage
- Standard deviation coverage. Standard deviation of the per base coverage.
- Minimum, average coverage excluding zero coverage regions
- Standard deviation excluding zero coverage regions. Standard deviation of the per base coverage, excluding regions without coverage.
- % GC. GC content of the reference sequence.
- Consensus length
- Fraction of reference covered
- Count (zero coverage regions)
- Minimum, maximum, average and total length (zero coverage regions)
- Standard deviation length (zero coverage regions). Standard deviation of the distribution of the lengths of all the zero coverage regions on that contig.

29.4 Whole Genome Coverage Analysis

The Whole Genome Coverage Analysis tool is designed to identify regions in read mappings with unexpectedly low or high coverage. Such regions may be indicative of a deletion or an amplification in the sample relative to the reference. The algorithm fits a Poisson distribution to the observed coverage in the positions of the mapping. This distribution is used as the basis for identifying the regions of 'Low coverage' or 'High coverage'. The user chooses two parameter values in the wizard: (1) a 'Minimum length' and (2) a 'P-value threshold' value. The algorithm inspects the coverage in each of the positions in the read mapping and marks the ones with coverage in the lower or upper tails of the estimated Poisson distribution, using the provided p-value as cut-off. Regions with consecutive positions marked consistently as having low (respectively high) coverage, longer than the user specified 'Minimum length' value are called as 'Low coverage' (respectively 'High coverage') regions or "signatures".

To run the Whole Genome Coverage Analysis tool:

Tools | Quality Control | Whole Genome Coverage Analysis

In the first dialog, select a reads track or read mapping and click **Next**. This opens the dialog shown in figure 29.25.

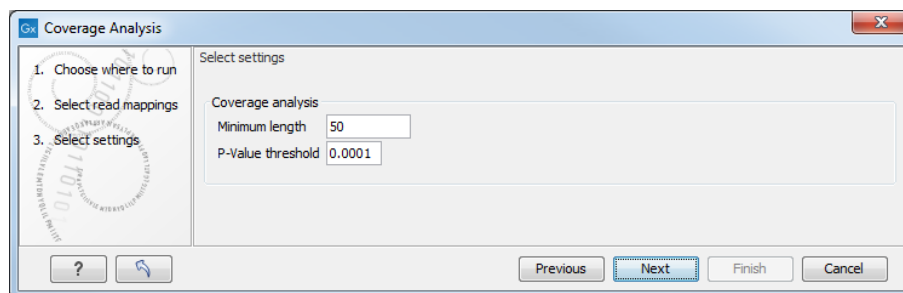


Figure 29.25: Specify the p-value cutoff.

Set the p-value and minimum length cutoff. Click **Next** and specify the result handling (figure 29.26).

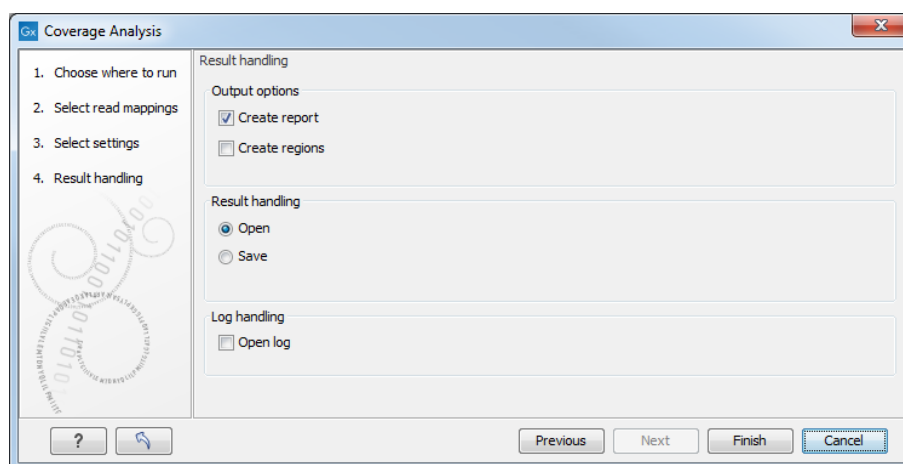


Figure 29.26: Specify the output.

Selecting "Create report" will generate a report made of 2 tables (figure 29.27). The first one, called References, lists per chromosome the number of reads, their length, and how many signatures of unexpectedly low or high coverage was found in the mapping. Signatures are simply regions with a number of consecutive positions exceeding the minimum length parameter, with either low or high coverage. The second table lists on 2 rows low and high coverage signatures found, as well as how many reads were used to calculate these signatures.

Selecting the "Create regions" will generate the annotation track carrying the name of the original file followed by (COV). This file can be visualized as an annotation track or as a table depending on the users choice. The annotation table contains a row for each detected low or high coverage region, with information describing the location, the type and the p-value of the detected region. The p-value of a region is defined as the average of the p-values calculated for each of the positions in the region.

An example of a track output of the Whole Genome Coverage Analysis tool is shown in figure 29.28.

The Whole Genome Coverage Analysis table includes the following columns (figure 29.28):

1 References

Reference	Reads	Length	Signatures
14	574,513	107,349,540	1,656
18	516,977	78,077,248	800
Total	1,091,490	185,426,788	2,456

2 Signatures

Signature	Reads used	Signatures
High Coverage	545,996	2,456
Low Coverage	545,996	0
Total		2,456

Figure 29.27: The report output.

Rows: 7,909 Table view: Homo sapiens

Chromosome	Region	Name	p-value	Length
18	51886795..51887354	High Coverage	1.39E-7	560
18	51887911..51888438	High Coverage	1.13E-7	528
18	51889026..51889406	High Coverage	2.31E-7	381
18	51891981..51892287	High Coverage	1.52E-7	307
18	51898705..51899113	High Coverage	1.87E-7	409
18	51900488..51900773	High Coverage	7.30E-8	286
18	51904439..51904741	High Coverage	1.66E-7	303
18	51933110..51933581	High Coverage	1.49E-7	472
18	52051811..52052012	High Coverage	9.22E-8	202
18	52258271..52258721	High Coverage	1.51E-7	451
18	52262019..52262459	High Coverage	2.91E-8	441
18	52264927..52265493	High Coverage	7.71E-8	567
18	52411949..52412210	High Coverage	5.05E-7	262
18	52544690..52545126	High Coverage	6.20E-8	437
18	52546468..52546831	High Coverage	1.52E-8	364
18	52551440..52551778	High Coverage	1.29E-7	339
18	52555072..52555458	High Coverage	9.10E-8	387
18	52556325..52556743	High Coverage	1.05E-7	419

Create Track from Selection

Table Settings: Column width: Automatic

Show column: ☒ Chromosome ☒ Region ☒ Name ☒ p-value ☒ Length

Figure 29.28: The table output with detailed information on each region.

- **Chromosome** The name is taken from the reference sequence used for mapping
- **Region** The start and end position of this region on the reference sequence
- **Name** The type of annotation: high or low coverage
- **P-Value** The calculated significance p-value for this region
- **Length** The length of the region

For the visual inspection and comparison to known gene/transcripts or other kind of annotations, all region are also annotated on the read mapping.

Chapter 30

Read mapping

Contents

30.1 Map Reads to Reference	790
30.1.1 Selecting the reads	790
30.1.2 References and masking	790
30.1.3 Mapping parameters	791
30.1.4 Mapping paired reads	794
30.1.5 Non-specific matches	796
30.1.6 Gap placement	796
30.1.7 Mapping computational requirements	797
30.1.8 Reference caching	798
30.1.9 Mapping output options	798
30.1.10 Summary mapping report	799
30.2 Map Long Reads to Reference	801
30.2.1 Map Long Reads to Reference output	802
30.3 Reads tracks and stand-alone read mappings	804
30.3.1 Coloring of mapped reads	804
30.3.2 Reads tracks	806
30.3.3 Stand-alone read mapping	813
30.4 Local Realignment	820
30.4.1 Method	820
30.4.2 Realignment of unaligned ends	821
30.4.3 Guided realignment	821
30.4.4 Multi-pass local realignment	823
30.4.5 Known limitations	824
30.4.6 Computational requirements	825
30.4.7 Run the Local Realignment tool	825
30.5 Merge Read Mappings	827
30.6 Remove Duplicate Mapped Reads	828
30.6.1 Algorithm details and parameters	829
30.6.2 Running remove duplicate mapped reads	830
30.7 Extract Consensus Sequence	831

30.1 Map Reads to Reference

Read mapping is a very fundamental step in most applications of high-throughput sequencing data. *CLC Genomics Workbench* includes read mapping in several other tools (such as in the Map Reads to Contigs tool, or for RNA-Seq Analysis), but this chapter will focus on the core read mapping algorithm. At the end of the chapter you can find descriptions of the read mapping reports and a tool to merge read mappings.

In addition, the mapper has a special mode for reads longer than 500bp. It is possible to mix sequence lists with reads that are shorter and reads that are longer than 500bp for the same mapping. In this case the appropriate mapping algorithm will be applied to each of the sequence lists.

The Map Reads to Reference tool does not support reads longer than 100,000bp and is not recommended for Oxford Nanopore reads.

30.1.1 Selecting the reads

To run the **Map Reads to Reference** tool, go to:

Tools | Resequencing Analysis (🔍) | **Map Reads to Reference** (📁)

In the first dialog, select the sequences or sequence lists containing the sequencing data (figure 30.1). Please note that reads longer than 100,000 bases are not supported.

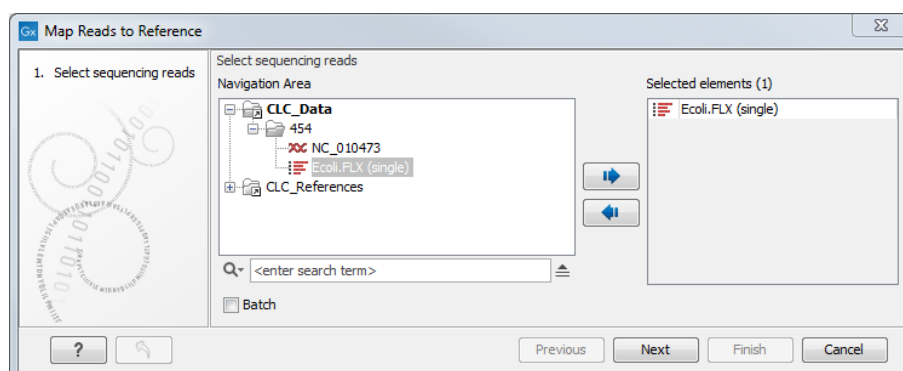


Figure 30.1: Specifying the reads as input. You can also choose to work in batch.

30.1.2 References and masking

When the sequences are selected, click **Next**, and you will see the dialog shown in figure 30.2.

At the top, select one or more reference sequences by clicking the **Browse** (🔍) button. You can select either single sequences, a list of sequences or a sequence track as reference. Note the following constraints:

- single reference sequences longer than 2gb ($2 \cdot 10^9$ bases) are not supported.
- a maximum of 120 input items (sequence lists or sequence elements) can be used as input to a single read mapping run.

Including or excluding regions (masking)

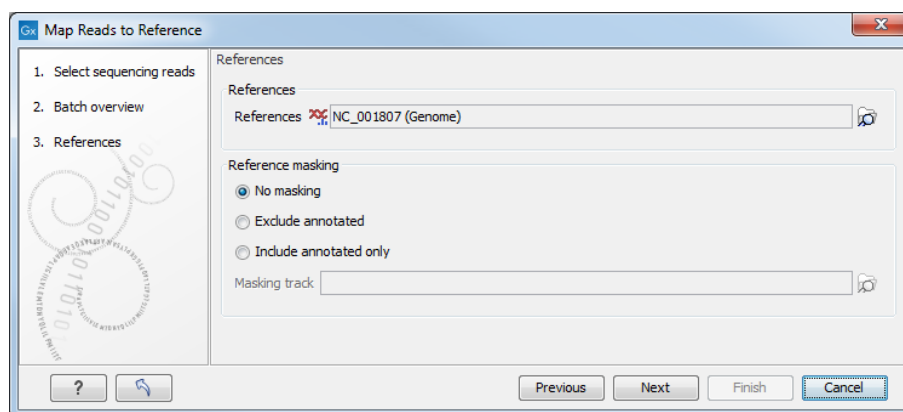


Figure 30.2: *Specifying the reference sequences and masking.*

The next part of the dialog shown in figure 30.2 lets you *mask* the reference sequences. Masking refers to a mechanism where parts of the reference sequence are not considered in the mapping. This can be useful for example when mapping data is captured from specific regions (e.g. for amplicon resequencing). The output will still include the full reference sequence, but no reads will be mapped in the ignored regions.

Note that you should be careful that your data is indeed only sequenced from the target regions. If not, some of the reads that would have matched a masked-out region perfectly may be placed wrongly at another position with a less-perfect match and lead to wrong results for subsequent variant calling. For resequencing purposes, we recommend testing whether masking is appropriate by running the same data set through two rounds of read mapping and variant calling: one with masking and one without. At the end, comparing the results will reveal if any off-target sequences cause problems in the variant calling.

Masking out repeats or using other masks with many regions is not recommended. Repeats are handled well without masking and do not cause any slowdown. On the contrary, masking repeats is likely to cause a dramatic slowdown in speed, increase memory requirements and lead to incorrect read placement.

To mask a reference sequence, first click the **Include** or **Exclude** options, and then click the **Browse** (🔍) button to select a track to use for masking. If you have annotations on a sequence instead of a track, you can convert the annotation type to a track (see section 27.8).

30.1.3 Mapping parameters

Clicking **Next** leads to the parameters for the read mapping (see figure 30.3).

The first parameter allows the mismatch cost to be adjusted:

- **Match score** The positive score for a match between the read and the reference sequence. It is set by default to 1 but can be adjusted up to 10.
- **Mismatch cost** The cost of a mismatch between the read and the reference sequence. Ambiguous nucleotides such as "N", "R" or "Y" in read or reference sequences are treated as mismatches and any column with one of these symbols will therefore be penalized with the mismatch cost.

After setting the mismatch cost you need to choose between linear gap cost and affine gap cost,

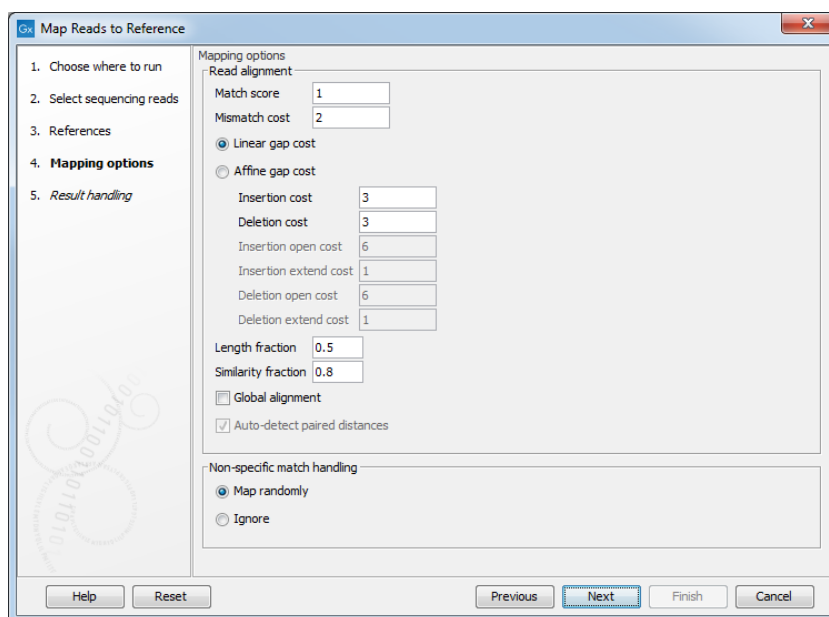


Figure 30.3: Setting parameters for the mapping.

and depending on the model you choose, you need to set two different sets of parameters that control how gaps in the read mapping are penalized.

- **Linear gap cost** The cost of a gap is computed directly from the length of the gap and the insertion or deletion cost. This model often favors small, fragmented gaps over long contiguous gaps. If you choose linear gap cost, you must set the insertion cost and the deletion cost:
 - Insertion cost. The cost of an insertion in the read (a gap in the reference sequence). The cost of an insertion of length ℓ will be $\ell \cdot \text{Insertion cost}$.
 - Deletion cost. The cost of a deletion in the read (gap in the read sequence). The cost of a deletion of length ℓ will be $\ell \cdot \text{Deletion cost}$.
- **Affine gap cost** An extra cost associated with opening a gap is introduced such that long contiguous gaps are favored over short gaps. If you chose affine gap cost, you must also set the cost of opening an insertion or a deletion:
 - Insertion open cost. The cost of opening an insertion in the read (a gap in the reference sequence).
 - Insertion extend cost. The cost of extending an insertion in the read (a gap in the reference sequence) by one column.
 - Deletion open cost. The cost of a opening a deletion in the read (gap in the read sequence).
 - Deletion extend cost. The cost of extending a deletion in the read (gap in the read sequence) by one column.

Using affine gap cost, an insertion of length ℓ is penalized by
 $\text{Insertion open cost} + \ell \cdot \text{Insertion extend cost}$

and a deletion of length ℓ is penalized by

Deletion open cost + $\ell \cdot$ Deletion extend cost

In this way long consecutive gaps get a lower cost per column than small fragmented gaps and they are therefore favored.

Adjusting the cost parameters above can improve the mapping quality, especially when the read error rate is high or the reference is expected to differ significantly from the sequenced organism. For example, if the reads are expected to contain many insertions and/or deletions, it can be a good idea to lower the insertion and deletion costs to allow more of such errors. However, one should also consider the possible drawbacks when adjusting these settings: reducing the insertion and deletion cost increases the risk of mapping reads to the wrong positions in the reference.

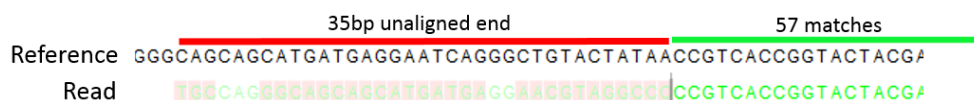


Figure 30.4: An alignment of a read where a region of 35bp at the start of the read is unaligned while the remaining 57 nucleotides matches the reference.

Figure 30.4 shows an example using linear gap cost where the read mapper is unable to map a region in a read due to insertions in the read and mismatches between the read and the reference. The aligned region of the read has a total of 57 matching nucleotides which result in an alignment score of 57 which is optimal when using the default cost for mismatches and insertions/deletions (2 and 3 respectively). If the mapper had aligned the remaining 35bp of the read as shown in figure 30.5 using the default scoring scheme, the score would become: $(26 + 1 + 3 + 57) * 1 - 5 * 2 - 8 * 3 = 53$

In this case, the alignment shown in figure 30.4 is optimal since it has the highest score. However, if either the cost of deletions or mismatches were reduced by one, the score of the alignment shown in figure 30.5 would become 61 and 58, respectively, and thus make it optimal.

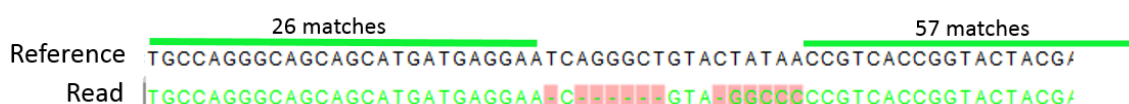


Figure 30.5: An alignment of a read containing a region with several mismatches and deletions. By reducing the default cost of either mismatches or deletions the read mapper can make an alignment that spans the full length of the read.

Once the optimal alignment of the read is found, based on the cost parameters described above, a filtering process determines whether this match is good enough for the read to be included in the output. The filtering threshold is determined by two factors:

- **Length fraction** The minimum percentage of the total alignment length that must match the reference sequence at the selected similarity fraction. A fraction of 0.5 means that at least half of the alignment must match the reference sequence before the read is included in the mapping (if the similarity fraction is set to 1). Note, that the minimal seed (word) size for read mapping is 15 bp, so reads shorter than this will not be mapped.
- **Similarity fraction** The minimum percentage identity between the aligned region of the read and the reference sequence. For example, if the identity should be at least 80% for the

read to be included in the mapping, set this value to 0.8. Note that the similarity fraction relates to the length fraction, i.e. when the length fraction is set to 50% then at least 50% of the alignment must have at least 80% identity (see figure 30.6).

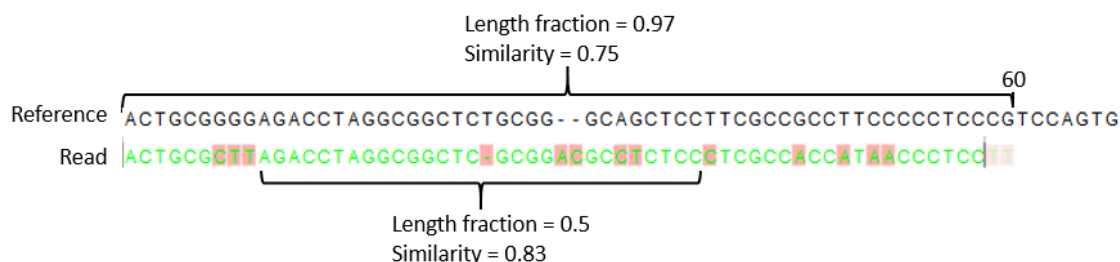


Figure 30.6: A read containing 59 nucleotides where the total alignment length is 60. The part of the alignment that gave rise to the optimal score has length 58 which excludes 2 bases at the left end of the read. The length fraction of the matching region in this example is therefore $58/60 = 0.97$. Given a minimum length fraction of 0.5, the similarity fraction of the alignment is computed as the maximum similarity fraction of any part of the alignment which constitute at least 50% of the total alignment. In this example the marked region in the alignment with length 30 (50% of the alignment length) has a similarity fraction of 0.83 which will satisfy the default minimum similarity fraction requirement of 0.8.

- **Global alignment** By default, mapping is done with **local alignment** of the reads to the reference. The advantage of performing local alignment instead of global alignment is that the ends are automatically left unaligned if there are many differences from the reference at the ends. For many sequencing platforms, the quality of the bases drop along the read, and a local alignment approach is desirable. Note that the aligned region has to be greater than the length threshold set. If **global alignment** is preferred, it can be enabled with a checkbox as shown in figure 30.3.

30.1.4 Mapping paired reads

- **Auto-detect paired distances** At the bottom of the dialog shown in figure 30.3 you can specify how **Paired reads** should be handled. You can read more about how paired data is imported and handled in section 7.3.10. If the sequence list used as input contains paired reads, this option will automatically be enabled - if it contains single reads, this option will not be applicable.

The *CLC Genomics Workbench* offers as the default choice to automatically calculate the distance between the pairs. If this is selected, the distance is estimated in the following way:

1. A sample of 200,000 reads is extracted randomly from the full data set and mapped against the reference using a very wide distance interval.
2. The distribution of distances between the paired reads is analyzed using a method that investigates the shape of the distribution and finds the boundaries of the peak.
3. The full sample is mapped using this distance interval.
4. The **history** (📄) of the result records the distance interval used.

The above procedure will be run for each sequence list used as input, assuming that they do not necessarily share the same library preparation and could have different distributions of paired distances. Figure 30.7 shows an example of the distribution of intervals with and without automatic pair distance interval estimation.

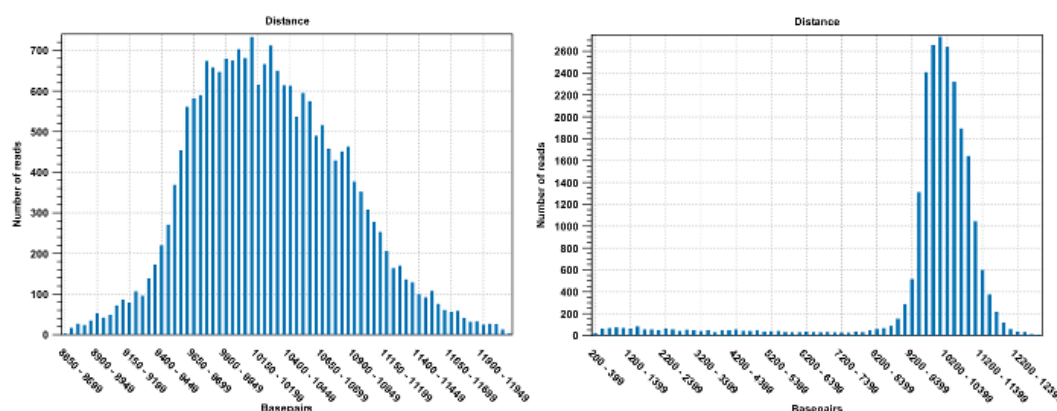


Figure 30.7: To the left: mapping with a narrower distance interval estimated by the workbench. To the right: mapping with a large paired distance interval (note the large right tail of the distribution).

Sometimes the automatic estimation of the distance between the pairs may return a warning "Few reads mapped as pairs so pair distance might not be accurate". This message indicates that fewer than 10.000 paired reads were available for estimation of the paired distance, hence the estimated distance may not be accurate. If in doubt, you may want to disable the option to automatically estimate paired distances and instead manually specify minimum and maximum distances between pairs on the input sequence list.

If the automatic detection of paired distances is not checked, the mapper will use the information about minimum and maximum distance recorded on the input sequence lists (see section 7.3.10).

If a large portion of pairs are flagged 'Broken' we recommend the following:

1. Inspect the detailed mapping report (see section 29.3) to deduce a distance setting interval - and compare this to the estimated distance used by the mapper (found in the mapping history).
2. Open the paired reads list and set a broad paired distance in the Elements tab. Then run a new mapping with the 'auto-detect...' OFF. Make sure to have a report produced. Open this report and look at the Paired Distance Distribution graph. This will tell you the distances that your pairs did map with. Use this information to narrow down the distance setting and perhaps run a third mapping using this.
3. Another cause of excessive amounts of broken pairs is misspecification of the read pair orientation. This can be changed in the Elements tab of the paired reads list prior to running a mapping.

See section 29.3 for further information.

When a paired distance interval is set, the following approach is used for determining the placement of read pairs:

- First, all the optimal placements for the two individual reads are found.

- Then, the allowed placements according to the paired distance interval are found.
- If both reads can be placed independently but no pairs satisfies the paired criteria, the reads are treated as independent and marked as a **broken pair**.
- If only one pair of placements satisfy the criteria, the reads are placed accordingly and marked as uniquely placed even if either read may have multiple optimal placements.
- If several placements satisfy the paired criteria, the pair is treated as a non-specific match (see section 30.1.5 for more information.)
- If one read is uniquely mapped but the other read has several placements that are valid given the distance interval, the mapper chooses the location that is closest to the first read.

30.1.5 Non-specific matches

At the bottom of the dialog, you can specify how **Non-specific matches** should be treated. The concept of Non-specific matches refers to a situation where a read aligns at *more than one position with an equally good score*. In this case you have two options:

- **Random**. This will place the read in one of the positions randomly.
- **Ignore**. This will not include the read in the final mapping.

Note that a read is only considered non-specific when the read matches equally well at several alignment positions. For example, if there are two possible alignment positions and one of them is a perfect match and the other involves a mismatch, the read is placed at the position with the perfect match and it is not marked as a non-specific match.

For paired data, reads are only considered non-specific matches if the entire pair could be mapped elsewhere with equal scores for both reads, or if the pair is broken in which case a read can be categorized as non-specific in the same way as single reads (see section 30.1.4).

When looking at the mapping, the default color for non-specific matches is yellow.

30.1.6 Gap placement

In the case of insertions or deletions in homopolymeric or repetitive regions, the precise placement of the insertion or deletion cannot be determined from the data. An example is shown in figure 30.8.

In this example, three As in the reference (top) have been replaced by two As in the reads (shown in red). The gap is placed towards the 5' end (left side), but could have been placed towards the 3' end with an equally good mapping score for the read as shown in figure 30.9.

Since either way of placing the gap is arbitrary, the goal of the mapper is to place the gaps consistently at the same side for all reads.

Many insertions and deletions in homopolymeric or repetitive regions reported in the 1000 Genomes public database have been identified based on mappings done with tools like BWA and Bowtie, which place insertions or deletions at the left side of a homopolymeric tract. To help

```

TTCTCAAACAAT
TTCTC-AACAAT
TTCTC-AACAAT
TTCTC-AACAAT
TTCTC-AACAAT
TTCTC-AACAAT
TTCTC-AACAAT

```

Figure 30.8: Three As in the reference (top) have been replaced by two As in the reads (shown in red). The gap is placed towards the 5' end, but could have been placed towards the 3' end with an equally good mapping score for the read.

```

TTCTCAAACAAT
TTCTCAA-CAAT
TTCTCAA-CAAT
TTCTCAA-CAAT
TTCTCAA-CAAT
TTCTCAA-CAAT
TTCTCAA-CAAT

```

Figure 30.9: Three As in the reference (top) have been replaced by two As in the reads (shown in red). The gap is placed towards the 3' end, but could have been placed towards the 5' end with an equally good mapping score for the read.

facilitate comparison of variant results with such public resources, the **Map Reads to Reference** tool places insertions or deletions in homopolymeric tracts at the left hand side. However, when comparing to dbsnp variant annotations, it is better to shift variants according to the 3' rule of HGVS. This can be done using the option "Move variants from VCF location to HGVS location" of the Amino Acids Changes tool [32.5.1](#).

30.1.7 Mapping computational requirements

The memory requirements of **Map Reads to Reference** depends on four factors: the size of the reference, the length of the reads, the read error rate and the number of CPU cores available. The limiting factor is often the size of the reference, while the contribution of the other three factors to the total memory consumption is usually small (see below).

A good estimate for the memory required by the base space read mapper to represent a reference is one MB for each Mbp in the reference. For example the human reference genome requires $3200 * 1MB = 3.2GB$ of memory.

An additional 4GB of memory should be reserved for the *CLC Genomics Workbench*, and thus the recommended minimum amount of memory for mapping short high quality reads (e.g. Illumina reads) to the human genome is 8GB. However, when mapping long reads with a high error rate, such as PacBio reads, each CPU core can add several hundred MB to the total memory consumption. Consequently, mapping long reads with high error rate on a machine with many CPU cores, can cause a large increase in the memory requirements for all CLC read mappers.

30.1.8 Reference caching

In some cases, repeated mappings against the same reference will result in a dramatically reduced runtime because the internal data structure used for mapping the reads, which is reference specific, can be reused. This has been enabled by storing files in the system tmp folder as a caching mechanism. Only a certain amount of disk space will be used and once reaching the limit, the oldest files are cleaned up. Consequently, the reference data structure files will automatically have to be recreated if the cache was filled or the tmp folder was cleaned up.

The default space limit is 16 GB.

This value can be changed by going to:

Edit | Preferences | Advanced | Read Mapper

30.1.9 Mapping output options

Clicking **Next** lets you choose how the output of the mapping should be reported (see figure 30.10).

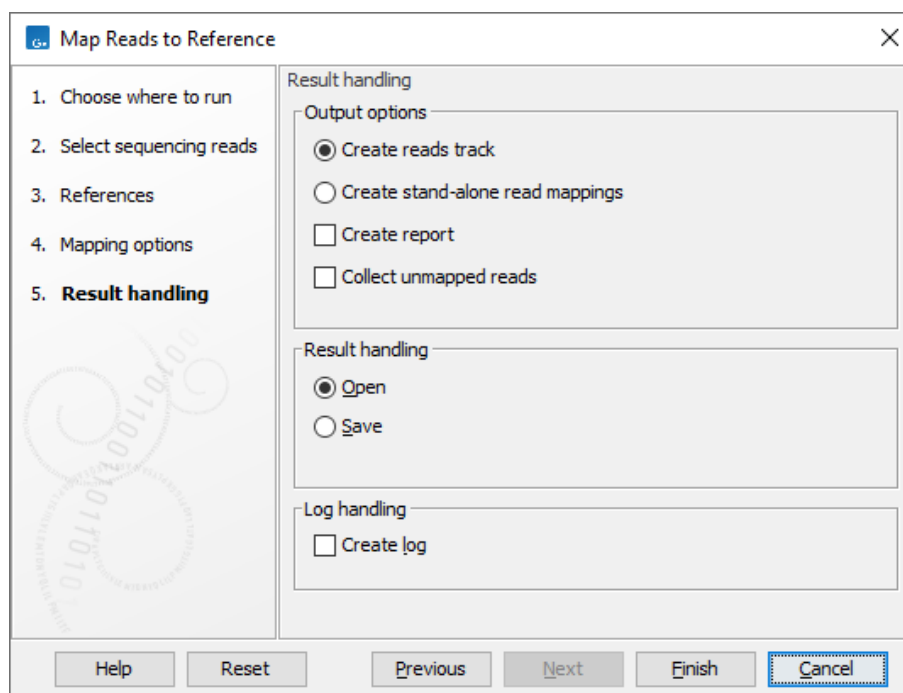


Figure 30.10: Mapping output options.

The main choice in output format is at the top of the dialog - the read mapping can either be stored as a track or as a stand-alone read mapping. Both options have distinct features and advantages:

- **Reads track.** A reads track is best used in the context of a Track List, where additional information about the reference, consensus sequence or annotations can be added and viewed alongside the reads. Details about viewing and editing Reads tracks are described in section 27. Unless any specific functionality of the stand-alone read mapping is required, we recommend to using the tracks output for the additional flexibility it brings in further analysis.

- **Stand-alone read mapping.** This output is more elaborate than the reads track and includes the full reference sequence with annotations. A consensus sequence is created as part of the output. Furthermore, the possibilities for detailed visualization and editing are richer than for the reads track (see section 22.7). However, stand-alone read mappings do not lend themselves well to comparative analyses. Note that if multiple reference sequences are used as input, a read mapping table is created (see section 30.3.3).

Read more about both output types in section 30.3. Note that the choice you make here is not definitive: it is possible to convert stand-alone read mappings to tracks and tracks (reads and annotation tracks) to stand-alone read mappings (see section 27.8).

In addition to the choice between the two main output options, there are two independent output options available that can be (de-)activated in both cases:

- **Create report.** This will generate a summary report as described in section 30.1.10.
- **Collect unmapped reads.** This will collect all the reads that could not be mapped to the reference into a sequence list (there will be one list of unmapped reads per sample, and for paired reads, there will be one list for intact pairs and one for single reads where the mate could be mapped).

Finally, you can choose to save or open the results. Clicking **Finish** will start the mapping.

30.1.10 Summary mapping report

If you choose to create a report as part of the read mapping (see section 30.1.9), this report will summarize the results of the mapping process. An example of a report is shown in figure 30.11. Double click on a graph to see it in full view and access data points in a table.

The information included in the report is:

- **Summary statistics.** A summary of the mapping statistics:
 - **Reads.** The number of reads and the average length.
 - **Mapped.** The number of reads that are mapped and their average length.
 - **Not mapped.** The number of reads that do not map and their average length.
 - **References.** Number of reference sequences.
- **Distribution of read length.** For each sequence length, you can see the number of reads and the distribution in percent. This is mainly useful if you don't have too much variance in the lengths as in e.g. Sanger sequencing data.
- **Distribution of matched reads lengths.** Equivalent to the above, except that this includes only the reads that have been matched to a contig.
- **Distribution of non-matched reads lengths.** Show the distribution of lengths of the rest of the sequences.
- **Paired reads distance distribution.** Section present only when paired reads were used, it displays a graph showing the distribution of paired sequences distances.

1 Mapping summary report

1.1 Summary statistics

	Count	Percentage of reads	Average length	Number of bases	Percentage of bases
References	1	-	20,158.00	20,158	-
Mapped reads	1,000	99.80%	76.00	76,000	99.89%
Not mapped reads	2	0.20%	40.00	80	0.11%
Reads in pairs	996	99.40%	250.00	75,696	99.50%
Broken paired reads	4	0.40%	76.00	304	0.40%
Total reads	1,002	100.00%	75.93	76,080	100.00%

1.2 Distribution of read length

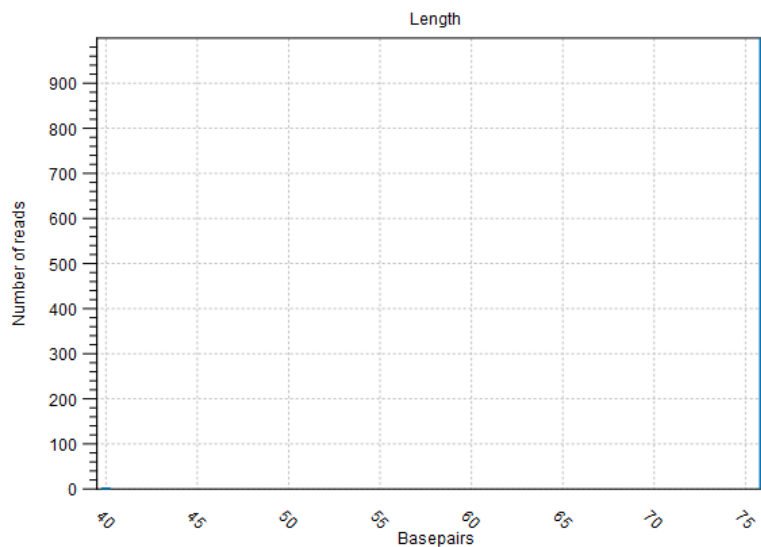


Figure 30.11: The summary mapping report.

You can copy the information from the report by selecting in the report and click **Copy** (📋). You can also export the report in Excel format.

30.2 Map Long Reads to Reference

The Map Long Reads to Reference tool enables aligning long reads to a reference with minimap2 [Li, 2018].

To run the tool, go to:

Tools | Resequencing Analysis (🔧) | **Map Long Reads to Reference** (📄)

Select one or more sequence lists containing long reads.

In the following step, select one or more reference sequences. You can select either individual sequences, a list of sequences or a sequence track as reference (figure 30.12).

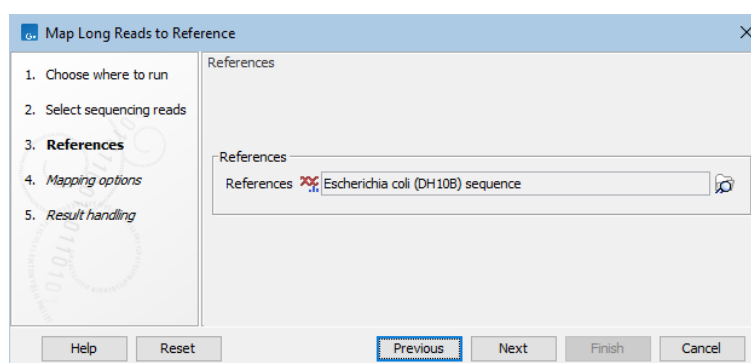


Figure 30.12: Map Long Reads to Reference reference selection.

In the Mapping options dialog, set the read mapping parameters (figure 30.13):

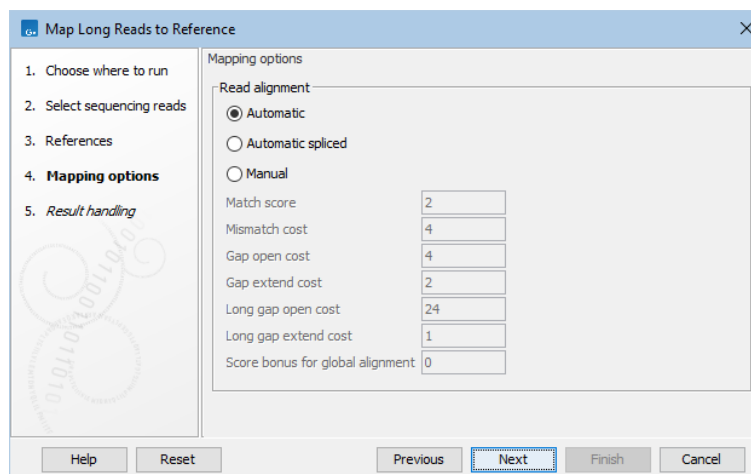


Figure 30.13: Map Long Reads to Reference mapping options.

- **Mapping mode.** Choose between the following modes for parameter setting:
 - *Automatic.* Match cost parameters are set automatically based on the read type of the first input (e.g. Oxford Nanopore or Pacbio HiFi).
 - *Automatic spliced.* Similar to *Automatic*, except it generates spliced alignments, which can be useful for visualizing RNA-seq data.

- *Manual*. Allows match costs to be specified manually. This will overwrite the read type specific match costs that would otherwise be used with the values entered below.
- **Match score**. Score of a match.
- **Mismatch cost**. Cost of a mismatch.
- **Gap open cost**. Cost of starting a gap.
- **Gap extend cost**. Cost of extending a gap.
- **Long gap open cost**. Cost of starting a long gap. Long gaps are typically more expensive to open but cheaper to extend. The alignment will use the cheaper alternative.
- **Long gap extend cost**. Cost of extending a long gap.
- **Score bonus for global alignment**. A bonus that may be added to the alignment score if the alignment encompasses all nucleotides of the read.

For guidance on adjusting match cost parameters, see section 30.1.3. Additional information on parameters are available from the minimap2 documentation.

30.2.1 Map Long Reads to Reference output

Outputs are selected from the dialog shown in figure 30.14.

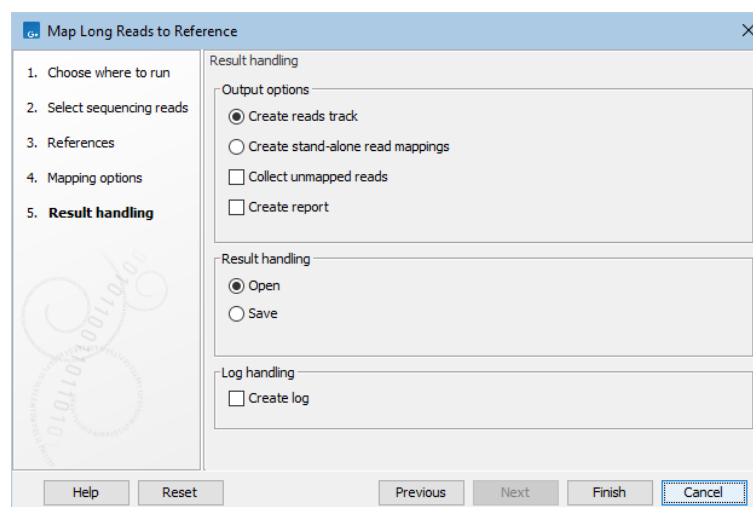


Figure 30.14: Map Long Reads to Reference output settings.

The main choice in output format is at the top of the dialog - the read mapping can either be stored as a track or as a stand-alone read mapping. Both options have distinct features and advantages:

- **Reads track**. A reads track is best used in the context of a Track List, where additional information about the reference, consensus sequence or annotations can be added and viewed alongside the reads. Unless any specific functionality of the stand-alone read mapping is required, we recommend to using the tracks output for the additional flexibility it brings in further analysis.

- **Stand-alone read mapping.** This output is more elaborate than the reads track and includes the full reference sequence with annotations. A consensus sequence is created as part of the output. Furthermore, the possibilities for detailed visualization and editing are richer than for the reads track. However, stand-alone read mappings do not lend themselves well to comparative analyses. Note that if multiple reference sequences are used as input, a read mapping table is created.

Read more about both output types in section [30.3](#).

In addition to the choice between the main output options, the following output are available:

- **Create report.** Creates a summary report.
- **Collect unmapped reads.** Collects all the reads that could not be mapped to the reference into a sequence list.

Map Long Reads to Reference report

1 Mapping summary report

1.1 Summary statistics

Input type: Single reads

	Count	Percentage of reads (%)	Average length	Number of bases	Percentage of bases (%)
References	1	-	4,686,137.00	4,686,137	-
Mapped reads	4	66.67	1,367,462.00	5,469,848	99.64
Not mapped reads	2	33.33	9,952.00	19,904	0.36
Total reads	6	100.00	914,958.67	5,489,752	100.00

1.2 Distribution of read length

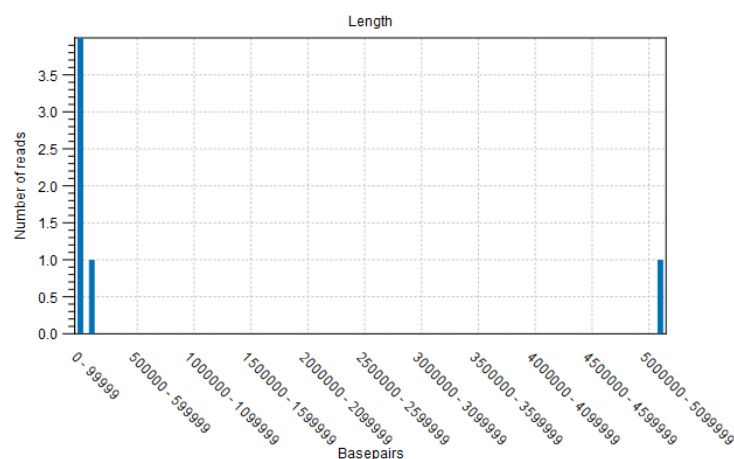


Figure 30.15: Map Long Reads to Reference summary report

The report will summarize the results of the mapping process. An example of the first part of the report is shown in figure [30.15](#).

The information included in the report is:

- **Summary statistics.** A summary of the mapping statistics:

- **References.** The count of reference sequences, the average length, and the total number of bases.
 - **Mapped reads.** The count of mapped reads, the percentage of mapped reads relative to total reads, the average read length, the total number of bases, and the percentage of bases relative to the total read count.
 - **Not mapped reads.** The count of un-mapped reads, the percentage of un-mapped reads relative to total reads, the average read length, the total number of bases, and the percentage of bases relative to the total read count.
 - **Total reads.** The total count of reads, the average read length, and the total number of bases.
- **Distribution of read length.** A visual representation of the occurrence of each specific read length, offering insights into the overall pattern and variability in the dataset.
 - **Distribution of mapped reads lengths.** Equivalent to the above, but including mapped reads only.
 - **Distribution of un-mapped reads lengths.** Equivalent to the above, but including un-mapped reads only.

30.3 Reads tracks and stand-alone read mappings

Two read mapping formats are supported: reads tracks and stand-alone read mappings. As a basic overview:

Reads tracks are designed for viewing alongside other results that are based on the same reference genome coordinates, using a Track List. Most NGS-related analysis functionality has been implemented using tracks, taking advantage of the consistent coordinate system.

Stand-alone read mappings provide rich visualization and editing features. They contain features such as a consensus sequence and coverage graph and can thus be useful when working with a particular read mapping in detail. Further details about working with stand alone read mappings can be found in section [22.7](#).

Read mappings can be converted from stand-alone to reads tracks or vice versa as described in section [27.8](#).

In this section, we describe the meaning of the default coloring of reads in read mappings section [30.3.1](#), the features of reads tracks section [30.3.2](#) and the features of stand-alone read mappings section [30.3.3](#).

30.3.1 Coloring of mapped reads

The mapped reads are colored by default according to the following color code (see figure [30.16](#) for a simplified illustration of the color scheme):

- Single reads mapping in their **forward** direction are **green**.
- Single reads mapping in their **reverse** direction are **red**.

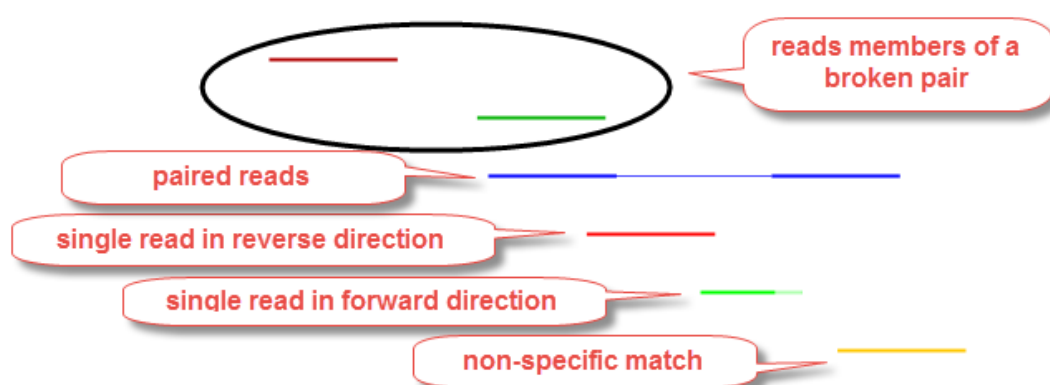


Figure 30.16: Default reads color in a stand-alone read mapping where the read layout is not set to "Packed": you can see that members of a broken pairs are in darker shades than the single reads.

- **Paired reads** are **blue**. **Reverse paired** reads are **light blue** (always in stand-alone read mapping, and only if the option "Highlight reverse paired reads" is checked, as it is by default, in reads tracks). The thick line represents the read itself; the **thin line** represents the **distance between each read** in the pair.
- Reads from **broken pairs** are colored as single reads, i.e., according to their forward/reverse orientation or as a non-specific match. In stand-alone read mappings, reads that are members of a broken pair are highlighted in **darker shades of the read color**, unless the Read layout is set to "Packed". Broken pairs and Single reads cannot be differentiated in tracks.
- **Non-specific matches** are **yellow**. When a read would have matched equally well another place in the mapping, it is considered a non-specific match. This color will "override" the other colors. Note that when mapping to several reference sequences, i.e. chromosomes, a read is considered a double match when it matches more than once across all the chromosomes.
- **Unaligned ends**, that is the part of the reads that is not mapped to the reference (also known as soft-clipped read ends) will be shown with a **faded color**, e.g., light green, light red, light blue or light yellow, depending on the color of the read.
- **Deletions** are shown as **dashed** lines (figure 30.17). **Insertions** with a frequency lower than 1% are shown with a **black vertical line**.

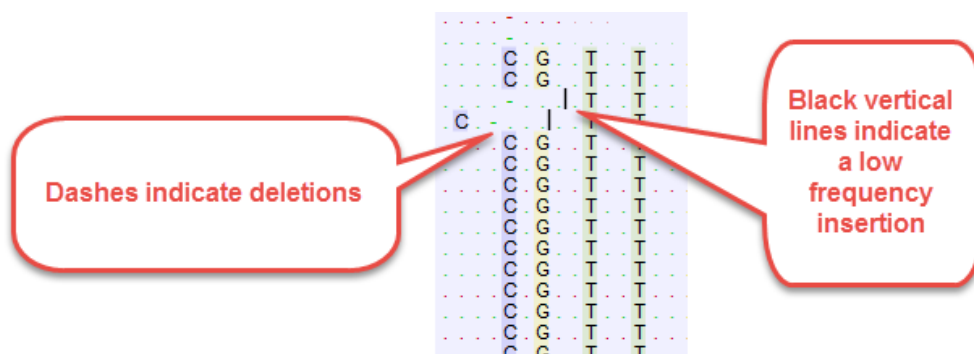


Figure 30.17: Reads track showing deletions as dashed lines, and low frequency insertions as black vertical lines.

- **Mismatches** between the read and reference are shown as narrow vertical lines on the reads (or black letters on a colored background at the nucleotide level) following the **Rasmol color scheme**: A in red, T in green, C in blue, G in yellow (figure 30.18). Ambiguous bases are in gray.

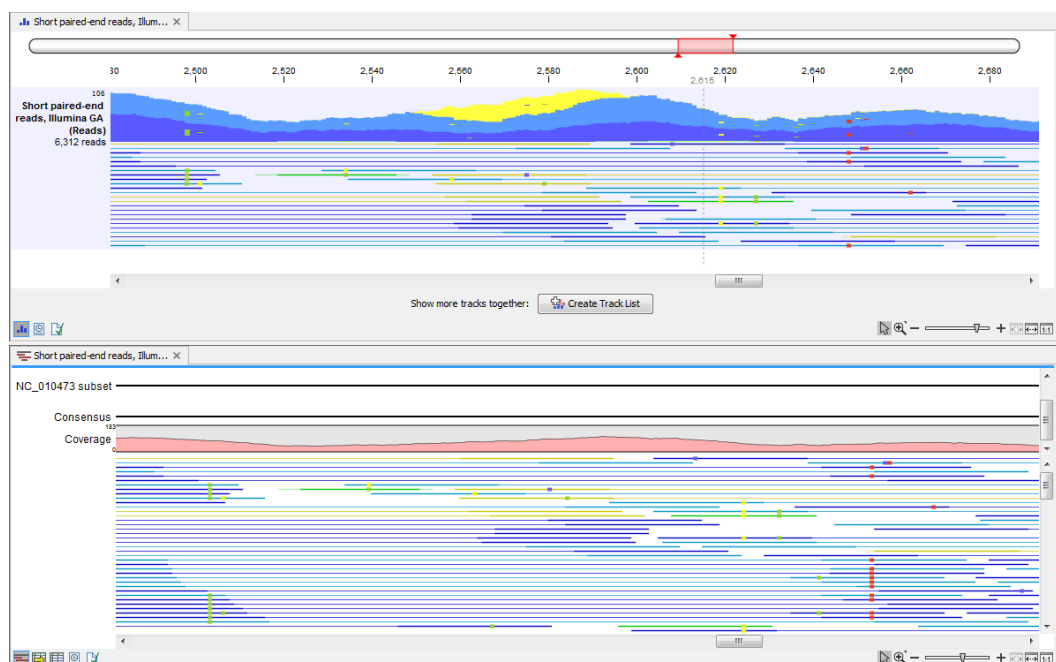


Figure 30.18: Mismatches between the reads and reference are shown as narrow vertical traits following the Rasmol color scheme. A reads track is shown above, a read mapping below.

These default colors can be changed using the side panel as shown in figure 30.19.

If your read mapping or track shows the message 'Too much data for rendering' on a gray background, simply zoom in to see your reads in more detail. This occurs when there are too many reads to be displayed clearly. More specifically, where there are more than 500,000 reads displayed in a reads track, more than 200,000 reads displayed in a read mapping, or when the region being viewed in a read mapping is longer than 200,000 bases. Paired reads count as one in these cases.

30.3.2 Reads tracks

Reads tracks can be included in a track list with other tracks that are based on a compatible genome, allowing for convenient, comparative visual analysis. For example, the track list shown in figure 30.20 contains, from top to bottom, a cytoband overview, a reads track, an annotation track of target regions, and a variant track.

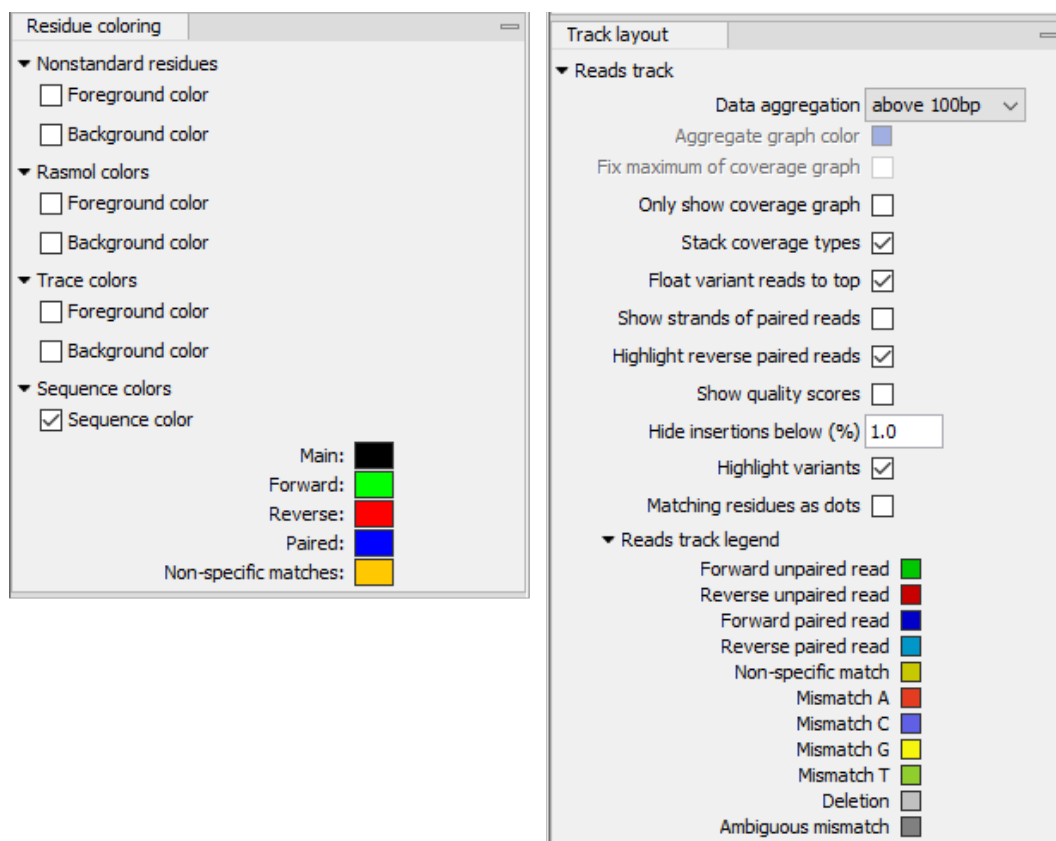


Figure 30.19: Coloring of mapped reads legends for read mappings (left) and reads track (right). Clicking on a color allows you to change it (except for read mappings at the Packed compactness level).

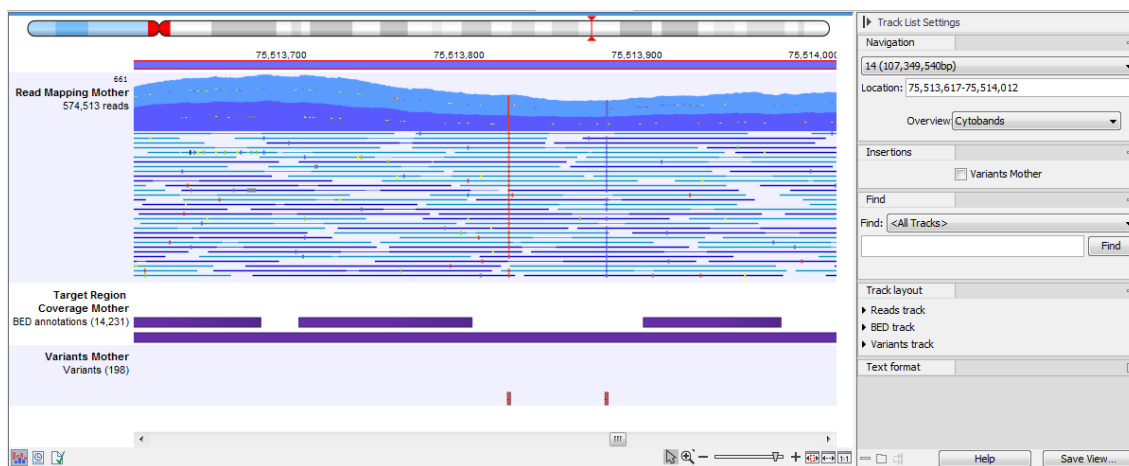


Figure 30.20: A track list containing a reads track, an annotation track and a variant track.

Reads tracks contain only the reads, placed where they mapped using the relevant reference genome coordinates. The information available when viewing a reads track depends on how far you zoom out or in.

Zooming out to an aggregated view

When zoomed out to the point where the mapped reads are aggregated, as shown in figure 30.21,

the height of the graph reflects the read coverage of that region, with the shade of blue representing the maximum, average and minimum coverage of the positions in that region:

- Dark blue: maximum read coverage
- Royal blue: average read coverage
- Light blue: minimum read coverage

A tooltip is shown when hovering the mouse cursor over any position in the track, reporting these values and the length of the region where those values apply.

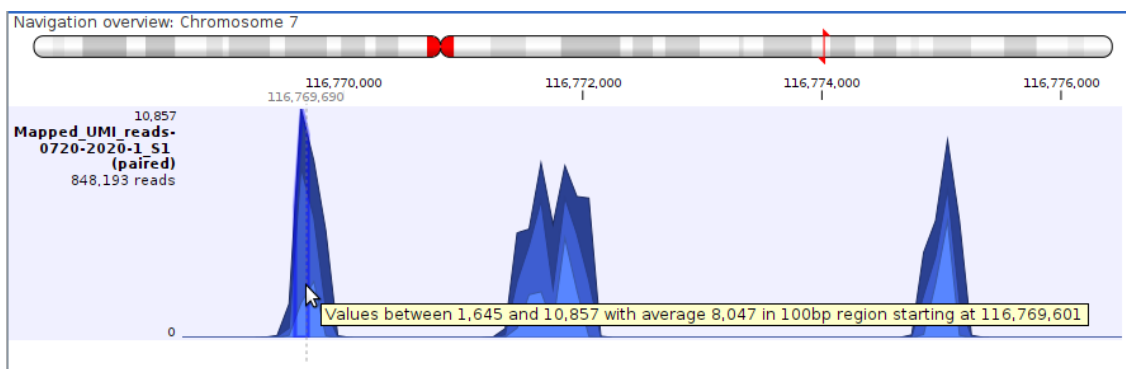


Figure 30.21: Shades of blue in an aggregated reads track represent the maximum, average and minimum reads coverage. Hovering the mouse cursor over a position brings up a tooltip with information about the coverage in that region.

Zooming in to a detailed view

When zoomed in fully, the individual nucleotide bases of each read are displayed (figure 30.22). A few reads are shown in detail, with the rest summarized in an coverage graph above. To see more reads in detail, increase the height of the reads track. For mappings where the coverage is deeper than the track height, a vertical scrollbar will appear when hovering the mouse cursor over the right hand side of the reads track. Use this to scroll through and view the details of all reads in the reads track.

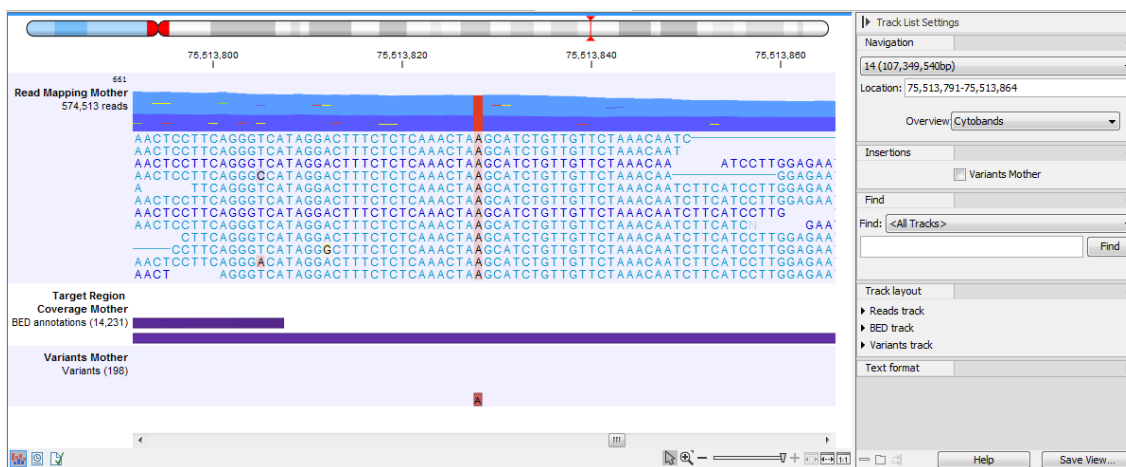


Figure 30.22: Zoom in fully to see the nucleotide bases of each read.

Reads that map across the origin of circular genomes are displayed at both the start and end of the mapping, and are marked with double arrows >> at the ends of the read to indicate that the read continues at the other end of the reference sequence.

When fully zoomed into a reads track, you can:

- Place the mouse cursor on a particular read and right-click to reveal a menu. Choose the option **Selected Read** to open a submenu where there are options for copying the reads, opening it in a new view, or using it as a query in a BLAST search (figure 30.23).

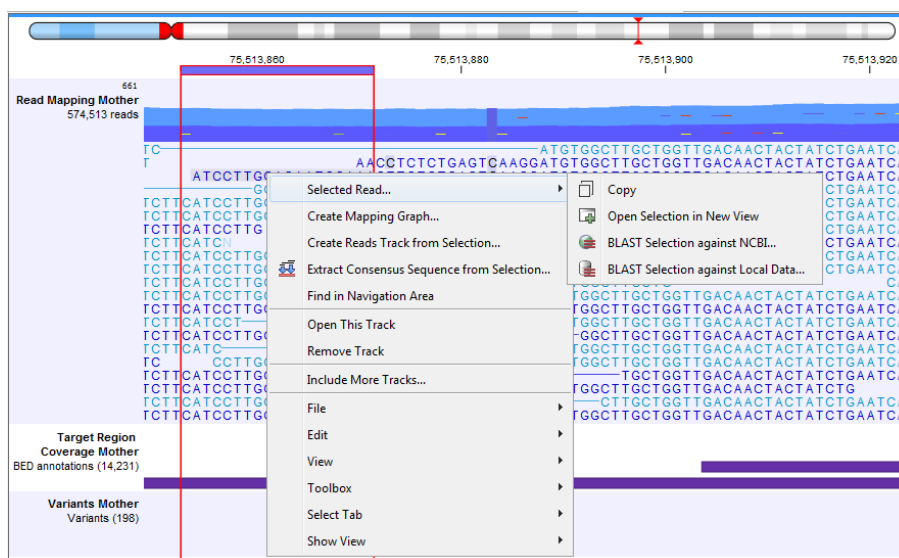


Figure 30.23: Adjusting the height of the track.

- Hover the mouse cursor over a position in a reads track to reveal a tooltip with information about the reads supporting certain base calls, or a deletion, at that position, as well as the directions of those reads (figures 30.24 and 30.25). For overlapping paired reads that disagree, ambiguous bases are represented by their IUPAC codes. Use 'Show strands of paired reads' to show all bases from overlapping paired reads, see section 30.3.2 for more information.

The tooltip uses the following symbols for the counts:

- + for single-end read mapped in forward direction, i.e., the number of green reads
- - for single-end read mapped in reverse direction, i.e., the number of red reads
- p+ for paired-end read mapped in forward direction (one count per pair), i.e., the number of dark blue reads
- p- for paired-end read mapped in reverse direction (one count per pair), i.e., the number of light blue reads
- ? for reads mapped in multiple places, i.e., the number of yellow reads

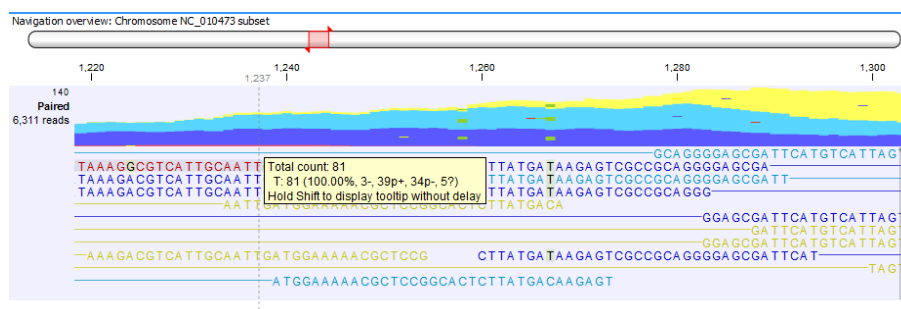


Figure 30.24: Example of tooltip information in a non-aggregated view of a reads track containing paired reads.

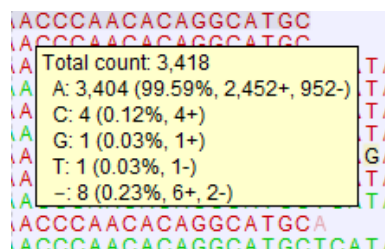


Figure 30.25: Example of tooltip information in a non-aggregated view of a reads track containing single reads. In this example, 8 reads support a deletion.

Tip: With larger mappings, there can be a short delay before the tooltip appears. To speed this up, click on the Shift key while moving the mouse over the reads. Tooltips then appear without delay.

For information on the side panel settings for reads tracks, see section 30.3.2.

Reads tracks Side Panel settings

The options for the **Side Panel** vary depending on which track is shown in the View Area. In figure 30.26 an example is shown for a read mapping.

- **Navigation**

- The first field gives information about which chromosome is currently shown. The drop-down list can be used to jump to a different chromosome.
- Location indicates the start and end positions of the shown region of the chromosome, but can also be used to navigate the track: enter a range or a single location point to get the visualization to zoom in on the region of interest. It is also possible to enter the name of a chromosome (MT: or 5:), the name of a gene or transcript (BRCA" or DHFR-001), or even the range on a particular gene or transcript (BRCA2:122-124).
- The Overview drop-down menu defines what is shown above the track: cytobands, or cytobands with aggregated data (figure 30.27). It can also be hidden all together.

- **Insertions** Only relevant for variant tracks.

- **Find** Not relevant for reads tracks, as it can only be used to search for annotations in tracks. To search for a sequence, use the Find function in the Side Panel of a stand-alone read mapping.

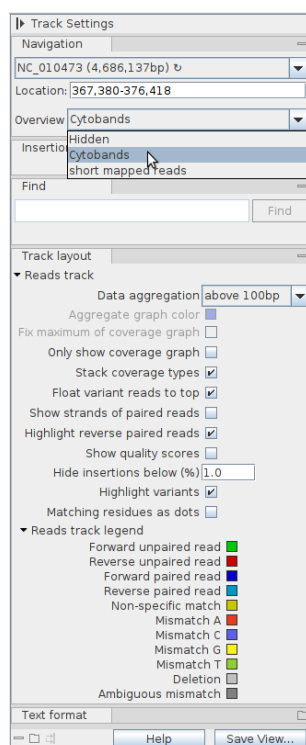


Figure 30.26: The Side Panel for reads tracks.

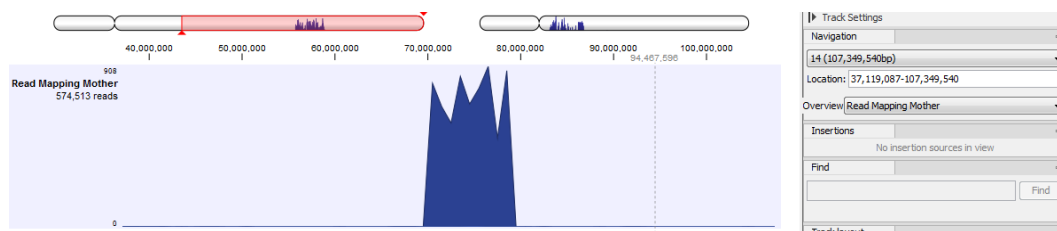


Figure 30.27: Cytobands with aggregated data allows you to navigate easily where needed based on the data location.

- **Track layout** The options for the Track layout varies depending on which track type is shown. The options for a reads track are:
 - **Data aggregation.** Allows you to specify whether the information in the track should be shown in detail or whether you wish to aggregate data. By aggregating data you decrease the detail level but increase the speed of the data display process, which is of particular interest when working with big data sets. The threshold (in bp) for when data should be aggregated can be specified with the drop-down box. The threshold describes the unit (or "bucket") size in base pairs, above which the data will start being aggregated. The bucket size depends on the track length and the zoom level. Hence, a data aggregation threshold with a low value will only show details when zoomed in, whereas a high value means that you can see details even when zoomed out. Please note that when using the high values, it will take longer time to display the data on the screen. Figure 30.26 shows the options for a reads track and an annotation track. The data aggregation settings can be adjusted for each displayed track type.
 - **Aggregate graph color.** Makes it possible to change the graph color.
 - **Fix maximum of coverage graph.** Specifies the maximum coverage to be shown on

the y-axis and makes the coverage on individual reads tracks directly comparable with each other. Applies across all of the read mapping tracks.

- **Only show coverage graph.** When enabled, only the coverage graph is shown and no reads are shown.
- **Stack coverage types.** Shows read specific coverage graph in layers as opposed to on top of each other.
- **Float variant reads to top.** When checked, reads with variations will appear at the top of the view.
- **Show strands of paired reads.** Show strands of paired end reads (see section 30.3.2).
- **Highlight reverse paired reads.** When enabled, read pairs with reverse orientation are highlighted with a light blue color.
- **Show quality scores.** Shows the quality score. Ticking this option makes it possible to adjust the colors of the residues based on their quality scores. A quality score of 20 is used as default and will show all residues with a quality score of 20 or below in a blue color. Residues with quality scores above 20 will have colors that correspond to the selected color code. In this case residues with high quality scores will be shown in reddish colors. Clicking once on the color bar makes it possible to adjust the colors. Double clicking on the slider makes it possible to adjust the quality score limits. In cases where no quality scores are available, blue (the color normally used for residues with a low quality score) is used as default color for such residues.
- **Hide insertions below (%).** Hides insertions where the percentage of reads containing insertions is below this value. To hide all insertions, set this value to 101.
- **Highlight variants.** Variants are highlighted
- **Matching residues as dots.** Replaces matching residues with dots, only variants are shown in letters.
- **Highlight unaligned ends.** Unaligned ends are highlighted. Aligned parts of the reads are faded.
- **Reads track legend.** Shows the coloring of the mapped reads. Colors can be adjusted by clicking on an individual color and selecting from the palette presented.

Overlapping paired reads

Depending on the sequencing protocol paired reads may overlap. If the mates of an overlapping pair disagree on one or more positions (e.g. due to sequencing errors) the viewer will represent this as a paired read with an IUPAC code. Where there is a gap in one of the reads, an N will be inserted. This reflects the way the Workbench variant detection tools would treat disagreements - if two overlapping reads do not agree about the variant base, they are both ignored. If you wish to inspect the mates of overlapping pairs you can check the side panel option 'Show strands of paired reads'. An example is shown in figure 30.28.

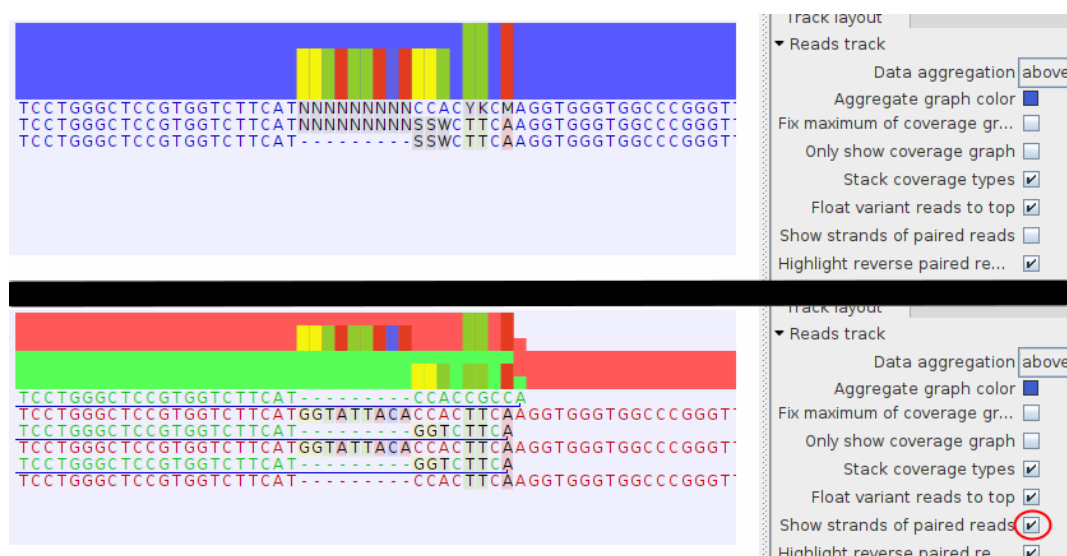


Figure 30.28: Discrepancies in the overlapping region of overlapping paired reads are indicated using Ns and IUPAC codes (top). When the individual reads of each pair are displayed by checking the box "Show strands of paired reads" in the Side Panel, the bases of each member of the pairs are displayed (bottom).

30.3.3 Stand-alone read mapping

Reads can be mapped to linear and circular chromosomes. Read mappings to circular genomes are visualized linearly as shown in figure 30.29.



Figure 30.29: Mapping reads to a circular chromosome. Reads that are marked with double arrows at the ends are reads that map across the starting point of the sequence. The arrows indicate that the alignment continues at the other end of the reference sequence.

Reads that map across the starting point of the sequence are shown both at the start and end of the reference sequence. Such reads are marked with >> at the end of the read to indicate that the alignment continues at the other end of the reference sequence.

Note that it is possible to select a portion of a read and access a right-click menu where you can Copy or Open in a New View the selected portion of the read: these options are available for the reference sequence at all compactness levels, and for individual reads at Low and Not compact levels.

If your read mapping or track shows the message 'Too much data for rendering' on a grey

background, simply zoom in to see your reads in more detail. This occurs when there are too many reads to be displayed clearly. More specifically, where there are more than 500,000 reads displayed in a reads track, more than 200,000 reads displayed in a read mapping, or when the region being viewed in a read mapping is longer than 200,000 bases. Paired reads count as one in these cases.

Read mapping settings

When you open a read mapping, there are many viewing options available in the **Side Panel** for customizing the layout.

Read layout.

- **Compactness.** Set the level of detail to be displayed. The level of compactness affects other view settings as well as the overall view. For example: if **Compact** is selected, quality scores and annotations on the reads will not be visible, even if these options are turned on under the "Nucleotide info" palette. Compactness can also be changed by pressing and holding the Alt key while scrolling with the mouse wheel or touchpad.
 - **Not compact.** This allows the mapping to be viewed in full detail, including quality scores and trace data for the reads, where present. To view such information, additional viewing options under the **Nucleotide info** view settings must also be selected. For further details on these, see section 22.1.1 and section 15.2.1.
 - **Low.** Hides trace data, quality scores and puts the reads' annotations on the sequence. The editing functions available when right-clicking on a nucleotide with compactness set to Low is shown in figure 30.30.

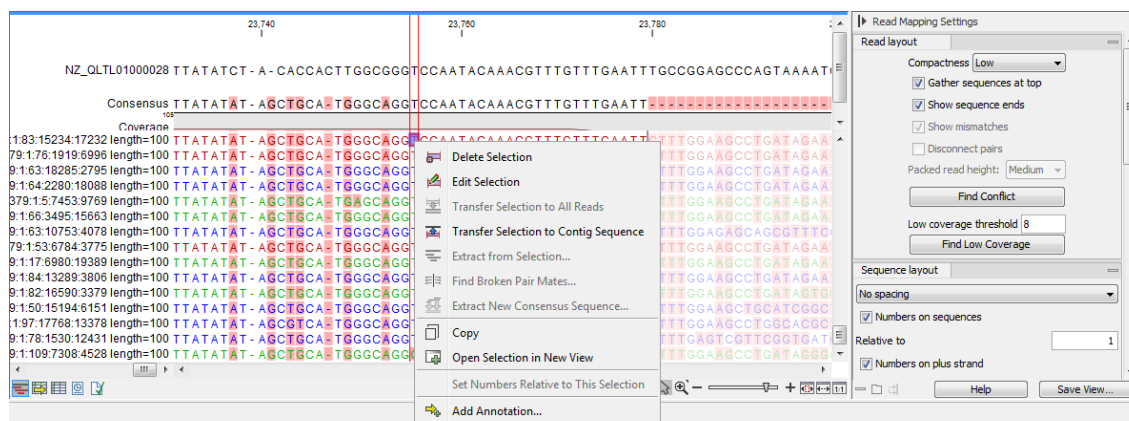


Figure 30.30: An example showing the low compactness setting.

- **Medium.** The labels of the reads and their annotations are hidden, and reads are shown as lines. The residues of the reads cannot be seen, even when zoomed in 100%.
- **Compact.** Like Medium but with less space between the reads.
- **Packed.** This uses all the horizontal space available for displaying the reads (figure 30.31). This differs from the other settings, which stack all reads vertically. When zoomed in to 100%, the individual residues are visible. When zoomed out, reads are represented as lines. Packed mode is useful when viewing large

amounts of data, but some functionality is not available. For example, the read mapping cannot be edited, portions cannot be selected, and color coding changes are not possible.

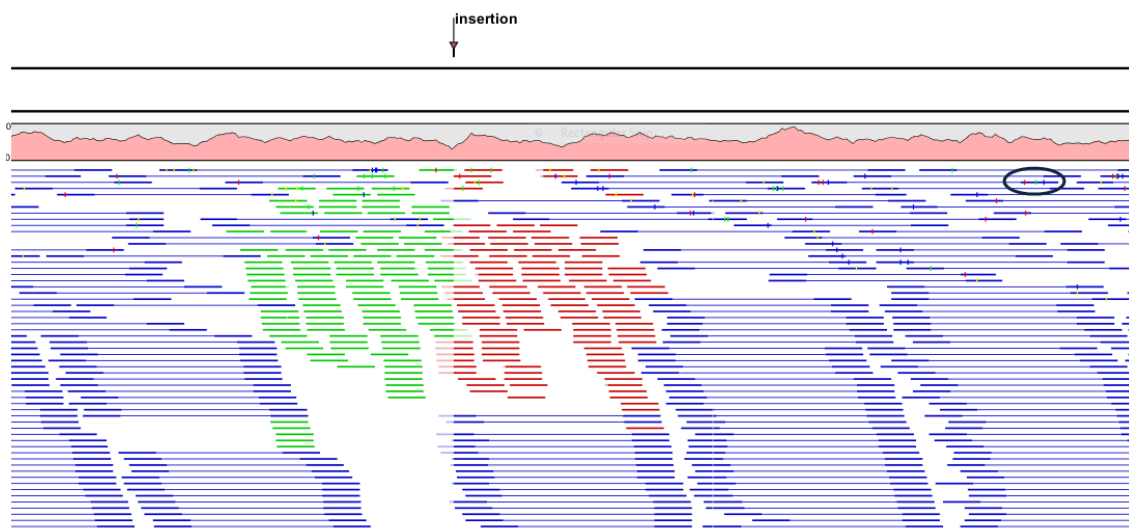


Figure 30.31: An example of the *Packed compactness* setting. Highlighted in black is an example of 3 narrow vertical lines representing mismatching residues.

- **Gather sequences at top.** When selected, the sequence reads contributing to the mapping at that position are placed right below the reference. This setting has no effect when the compactness level is Packed.
- **Show sequence ends.** When selected, trimmed regions are shown (faded traces and residues). Trimmed regions do not contribute to the mapping or contig.
- **Show mismatches.** When selected and when the compactness is set to Packed, based that do not match the reference at that position are highlighted by coloring them according to the Rasmol color scheme. Reads with mismatches are floated to the top of the view.
- **Show strands of paired reads.** When the compactness is set to Packed, display each member of a read pair in full and color them according to direction. This is particularly useful for reviewing overlap regions in overlapping read pairs.
- **Packed read height.** When the compactness is set to "Packed", select a height for the visible reads.

When there are more reads than the height specified, an overflow graph is displayed that uses the same colors as the sequences. Mismatches in reads are shown as narrow vertical lines, using colors representing the mismatching residue. Horizontal line colors correspond to those used for highlighting mismatches in the sequences (red = A, blue = C, yellow = G, and green = T). For example, a red line with half the height of the blue part of the overflow graph represents a mismatching "A" in half of the paired reads at that particular position.

- **Find Conflict.** Clicking this button selects the next position where there is an conflict. Mismatching residues are colored using the default color settings. You can also click on the Space bar of your keyboard to find the next conflict.
- **Low coverage threshold.** All regions with coverage up to and including this value are

considered low coverage. Clicking the 'Find low coverage' button selects the next region in the read mapping with low coverage.

Sequence layout. There is one parameter in this section in addition to those described in section 15.2.1

- **Matching residues as dots.** When selected, matching residues are presented as dots instead of as letters.

Annotation Layout and Annotation Types See section 15.3.1.

Residue coloring. There is one parameter in this section in addition to those described in section 15.2.1.

- **Sequence colors.** This setting controls the coloring of sequences when working in most compactness modes. The exception is Packed mode, where colors are controlled with settings under the "Match coloring" tab, described below.
 - **Main.** The color of the consensus and reference sequence. Black by default.
 - **Forward.** The color of forward reads. Green by default.
 - **Reverse.** The color of reverse reads. Red by default.
 - **Paired.** The color of read pairs. Blue by default. Reads from **broken pairs** are colored according to their orientation (forward or reverse) or as a non-specific match, but with a darker hue than the color of ordinary reads.
 - **Non-specific matches.** When a read would have matched equally well another place in the mapping, it is considered a non-specific match and is colored yellow by default. Coloring to indicate a non-specific match overrules other coloring. For mappings with several reference sequences, a read is considered a non-specific match if it matches more than once *across all the contigs/references*.

Colors can be adjusted by clicking on an individual color and selecting from the palette presented.

Alignment info. There are several parameters in this section in addition to the ones described in section 24.2.

- **Coverage:** Shows how many reads are contributing information to a given position in the read mapping. The level of coverage is relative to the overall number of reads.
- **Paired distance:** Plots the distance between the members of paired reads.
- **Single paired reads:** Plots the percentage of reads marked as single paired reads (when only one of the reads in a pair matches).
- **Non-specific matches:** Plots the percentage of reads that also match other places.
- **Non-perfect matches:** Plots the percentage of reads that do not match perfectly.
- **Spliced matches:** Plots the percentage of reads that are spliced.

Options for these parameters are:

- **Foreground color.** Colors the residues using a gradient, where the left side color is used for low coverage and the right side is used for maximum coverage.

- **Background color.** Colors the background of the residues using a gradient, where the left side color is used for low coverage and the right side is used for maximum coverage.
- **Graph.** Read coverage is displayed as a graph. The data points for the graph can be exported (see section 8.3).
 - **Height.** Specifies the height of the graph.
 - **Type.** The graph can be displayed as Line plot, Bar plot or as a Color bar.
 - **Color box.** For Line and Bar plots, the color of the plot can be set by clicking the color box. If a Color bar is chosen, the color box is replaced by a gradient color box as described under Foreground color.

Nucleotide info See section 15.2.1

Match coloring Coloring of the mapped reads when the Packed compactness option is selected. Colors can be adjusted by clicking on an individual color and selecting from the palette presented. Coloring of bases when other compactness settings are selected is controlled under the "Residue coloring" tab.

Find See section 15.2.1

Text format See section 15.2.1

Mapping table

When several reference sequences are used or you are performing de novo assembly with the reads mapped back to the contig sequences, all your mapping data will be accessible from a table (📄). It means that all the individual mappings are treated as one single file to be saved in the **Navigation Area** as a table.

An example of a mapping table for a de novo assembly is shown in figure 30.32.

Name	Consensus length	Total read count	Average coverage	Reference sequence	Reference length
Mate-pair_trimmed_(paired)_contig_1_mapping	1119751	1179062	84.50	Mate-pair_trimmed_(paired)_contig_1	1127355
Mate-pair_trimmed_(paired)_contig_2_mapping	499654	582379	92.85	Mate-pair_trimmed_(paired)_contig_2	502083
Mate-pair_trimmed_(paired)_contig_3_mapping	1308383	1436319	87.22	Mate-pair_trimmed_(paired)_contig_3	1316508
Mate-pair_trimmed_(paired)_contig_4_mapping	445649	504985	91.09	Mate-pair_trimmed_(paired)_contig_4	448078
Mate-pair_trimmed_(paired)_contig_5_mapping	150782	163994	84.46	Mate-pair_trimmed_(paired)_contig_5	155896
Mate-pair_trimmed_(paired)_contig_6_mapping	444810	440650	77.19	Mate-pair_trimmed_(paired)_contig_6	452804
Mate-pair_trimmed_(paired)_contig_7_mapping	258714	311331	95.06	Mate-pair_trimmed_(paired)_contig_7	263917
Mate-pair_trimmed_(paired)_contig_8_mapping	152864	148495	76.56	Mate-pair_trimmed_(paired)_contig_8	154036
Mate-pair_trimmed_(paired)_contig_9_mapping	1388	1884	120.97	Mate-pair_trimmed_(paired)_contig_9	1388
Mate-pair_trimmed_(paired)_contig_10_mapping	386155	456062	94.81	Mate-pair_trimmed_(paired)_contig_10	387445
Mate-pair_trimmed_(paired)_contig_11_mapping	621231	637299	81.58	Mate-pair_trimmed_(paired)_contig_11	628182
Mate-pair_trimmed_(paired)_contig_12_mapping	58737	55519	73.36	Mate-pair_trimmed_(paired)_contig_12	58737
Mate-pair_trimmed_(paired)_contig_13_mapping	280423	330855	92.50	Mate-pair_trimmed_(paired)_contig_13	284492
Mate-pair_trimmed_(paired)_contig_14_mapping	106203	124055	93.91	Mate-pair_trimmed_(paired)_contig_14	106203
Mate-pair_trimmed_(paired)_contig_15_mapping	2974	6970	190.63	Mate-pair_trimmed_(paired)_contig_15	2974
Mate-pair_trimmed_(paired)_contig_16_mapping	73161	88520	98.48	Mate-pair_trimmed_(paired)_contig_16	74313

Figure 30.32: The mapping table.

The information included in the table is:

- **Name.** When mapping reads to a reference, this will be the name of the reference sequence.
- **Consensus length.** The length of the consensus sequence. Subtracting this from the length of the reference will indicate how much of the reference that has not been covered by reads.

- **Total read count.** The number of reads. Reads with multiple hits on different reference sequences are placed according to your input for **Non-specific matches**
- **Single reads** and **Reads in pair.** Total number of reads, single and/or in pair.
- **Average coverage.** This is simply summing up the bases of the aligned part of all the reads divided by the length of the reference sequence.
- **Reference sequence.** The name of the reference sequence.
- **Reference length.** The length of the reference sequence.
- **Reference common name** and
- **Reference latin name.** Name, common name and Latin name of each reference sequence.

At the bottom of the table there are three buttons that can be used to open or extract sequences. Select the relevant rows before clicking on the buttons:

- **Open Mapping.** Opens the read mapping for visual inspection. You can also open one mapping simply by double-clicking in the table.
- **Extract Consensus/Contigs.** For de novo assembly results, the contig sequences will be extracted. For results when mapping against a reference, the **Extract Consensus** tool will be used (see section 30.48).
- **Extract Subset.** Creates a new mapping table with the mappings that you have selected.

You can export the table in Excel format.

Find broken pair mates

The **Find Broken Pair Mates** tool is not applicable to tracks, but is available only to stand-alone read mappings.

Figure 30.33 shows an example of a read mapping with paired reads (shown in blue). In this particular region, there are some broken pairs (red and green reads). Pairs are marked as broken if the respective orientation or distance between the reads is not right (see general info on handling paired data in section 7.3.10), or if one of the reads do not map at all.

In some situations it is useful to investigate where the mate of the broken pairs map. This would indicate genomic rearrangements, mis-assemblies of de novo assembly etc. In order to see this, select the region in question on the reference sequence, right-click and choose **Find Broken Pair Mates**.

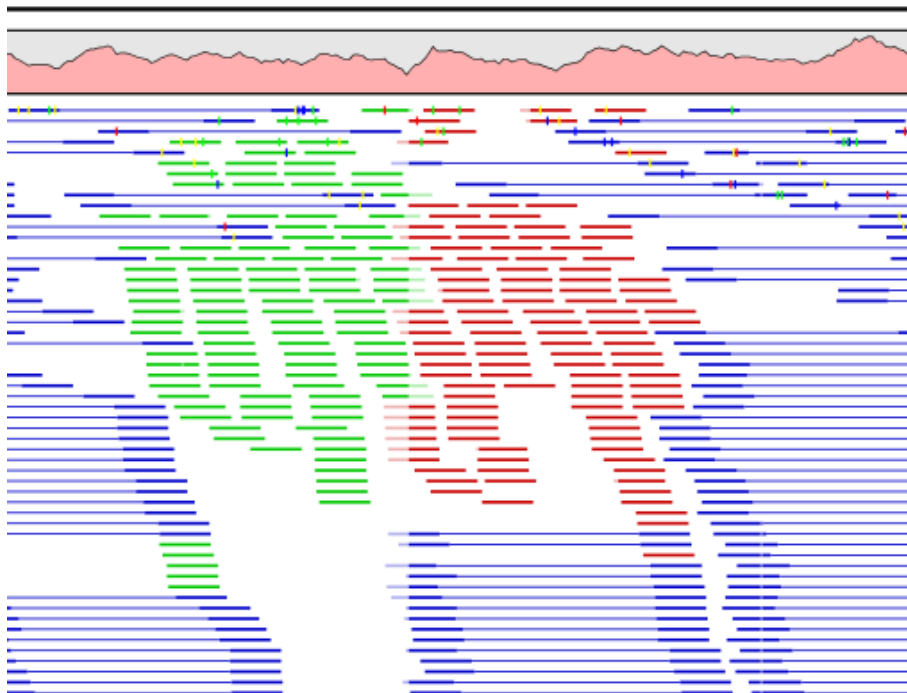
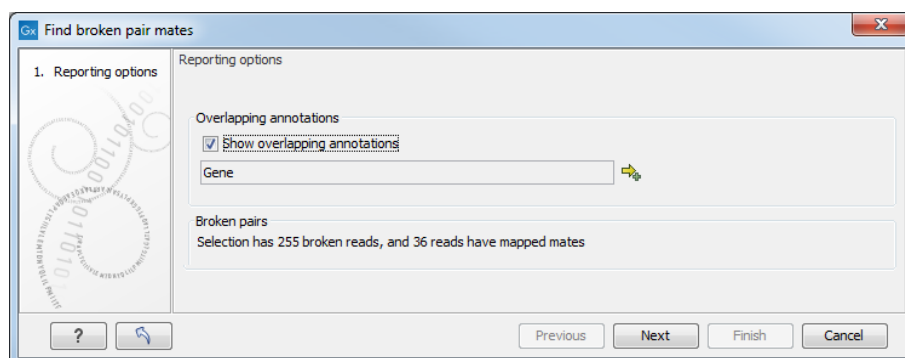
This will open the dialog shown in figure 30.34.

The purpose of this dialog is to let you specify if you want to annotate the resulting broken pair overview with annotation information. In this case, you would see if there are any overlapping genes at the position of the mates.

In addition, the dialog provides an overview of the broken pairs that are contained in the selection.

Click **Next** and **Finish**, and you will see an overview table as shown in figure 30.35.

The table includes the following information for both parts of the pair:

Figure 30.33: *Broken pairs.*Figure 30.34: *Finding the mates of broken pairs.*

Reference The name of the reference sequence where it is mapped

Start and end The position on the reference sequence where the read is aligned

Match count The number of possible matches for the read. This value is always 1, unless the read is a non-specific match (marked in yellow)

Annotations Shows a list of the overlapping annotations, based on the annotation type selected in figure 30.34.

You can select some or all of these broken pairs and extract them as a sequence list for further analysis by clicking the **Create New Sequence List** button at the bottom of the view.

Reference	Start	End	Match count	Mate reference	Mate start	Mate end	Mate match count
Mate-pair_trimmed_(paired)_contig_27	248	346	1	Mate-pair_trimmed_(paired)_contig_27	49897	49976	1
Mate-pair_trimmed_(paired)_contig_27	295	361	1	Mate-pair_trimmed_(paired)_contig_27	39	96	1
Mate-pair_trimmed_(paired)_contig_27	264	319	1	Mate-pair_trimmed_(paired)_contig_27	409	496	1
Mate-pair_trimmed_(paired)_contig_27	191	271	1	Mate-pair_trimmed_(paired)_contig_27	426	525	1
Mate-pair_trimmed_(paired)_contig_27	180	279	1	Mate-pair_trimmed_(paired)_contig_27	454	514	1
Mate-pair_trimmed_(paired)_contig_27	234	310	1	Mate-pair_trimmed_(paired)_contig_27	473	572	1
Mate-pair_trimmed_(paired)_contig_27	227	311	1	Mate-pair_trimmed_(paired)_contig_27	509	589	1
Mate-pair_trimmed_(paired)_contig_27	227	326	1	Mate-pair_trimmed_(paired)_contig_27	509	589	1
Mate-pair_trimmed_(paired)_contig_27	184	278	1	Mate-pair_trimmed_(paired)_contig_27	543	612	1
Mate-pair_trimmed_(paired)_contig_27	281	333	1	Mate-pair_trimmed_(paired)_contig_27	544	643	1
Mate-pair_trimmed_(paired)_contig_27	216	309	1	Mate-pair_trimmed_(paired)_contig_27	545	621	1
Mate-pair_trimmed_(paired)_contig_27	255	308	1	Mate-pair_trimmed_(paired)_contig_27	577	653	1
Mate-pair_trimmed_(paired)_contig_27	277	326	1	Mate-pair_trimmed_(paired)_contig_27	650	749	1
Mate-pair_trimmed_(paired)_contig_27	276	385	1	Mate-pair_trimmed_(paired)_contig_27	7411	7508	1

Figure 30.35: An overview of the broken pairs.

30.4 Local Realignment

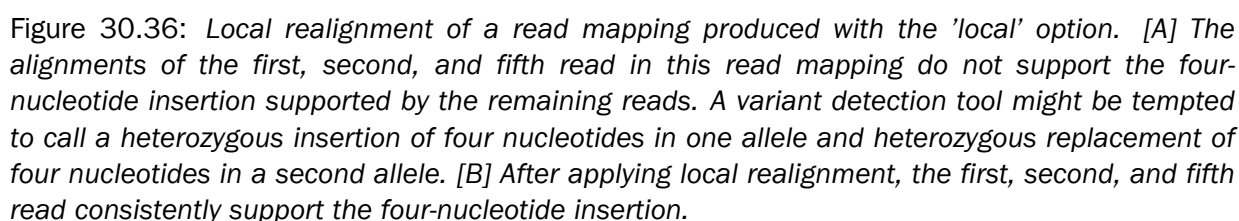
The goal of the local realignment tool is to improve on the alignments of the reads in an existing read mapping. The local realignment algorithm works by exploiting the information available in the alignments of *other* reads when it is attempting to re-align any given read. Most mappers do not use cross-read information as it would be computationally prohibitive to do within the mapping algorithm. However, once the reads have been mapped, local realignment procedures can exploit this information.

Realignment will typically occur in areas around insertions and deletions in the sample reads relative to the reference. In such regions we wish to see our reads mapped with one end of the read on one side of the indel and the rest mapped on the other side. However, the mapper that originally mapped the reads to the reference does not have information about the existence of an indel to use when mapping a given read. Thus, reads that are mapped to such regions, but that only have a short part of the read representing the region on one side of the indel, will typically not be mapped properly across the indel, but instead be mapped with this end unaligned, or into the indel region with many mismatches. The Local Realignment tool can use information from the other reads mapping to a region containing an indel, including reads that are located more centered across the indel and thus have been mapped with ends on either side of the indel. As a result an alternative mapping, as good as or better than the original, can be generated.

Local realignment will typically have an effect on any read mapping, whether the reads were mapped using a local or global alignment algorithm (i.e. with the Global alignment option of the mapping tool unchecked (the default) or checked, respectively). An example of the effect of using the Local Realignment tool on a read mapping made using the local alignment algorithm is shown in figure 30.36. An example in the case of a mapping made using the global alignment algorithm is shown in figure 30.37.

30.4.1 Method

The local realignment algorithm uses a variant of the approach described by Homer et al. [Homer N, 2010]. In the first step, alignment information of all input reads are collected in an efficient graph-based data structure, which is essentially similar to a de-Bruijn graph. This realignment graph represents how reads are aligned to the reference sequence and how reads overlap each other. In the second step, metadata are derived from the graph structure that indicate at which alignment positions realignment could potentially improve the read mapping, and also provides hypotheses as to how reads should be realigned to yield the most concise multiple alignment. In the third step the realignment graph and its metadata are used to actually perform the local



30.4.2 Realignment of unaligned ends

30.4.3 Guided realignment

One limitation of the local realignment algorithm employed is that at least one read must be aligned correctly according to the true indel present in the data. If none of the reads is aligned correctly, local realignment cannot improve the alignment, since it lacks information about how to do so. To overcome this limitation, local realignment can be guided in two ways:

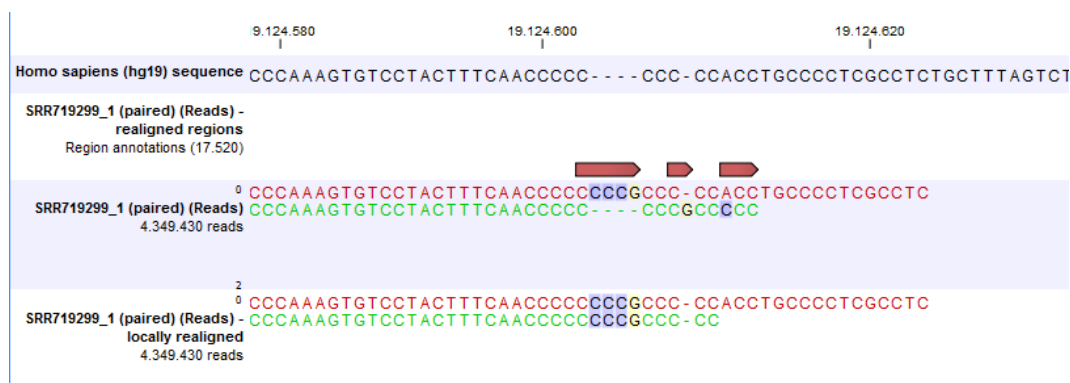


Figure 30.37: Local realignment of a read mapping produced with the 'global' option. Before realignment the green read was mapped with two mismatches. After realignment it is mapped with the inserted 'CCCG' sequence (seen in the alignment of the red read) and no mismatches.

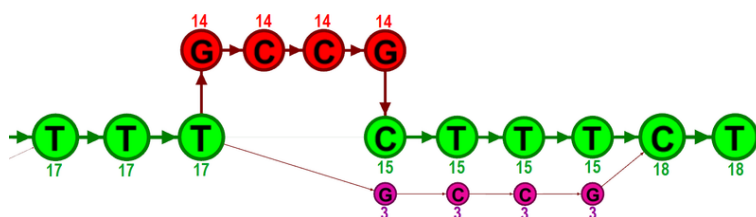


Figure 30.38: The green nodes represent nucleotides of the reference sequence. The four red nodes represent the four-nucleotide insertion observed in fourteen mapped reads. The four violet nodes represent the four mismatches to the reference sequence observed in three mapped reads. During realignment of the original reads, two possible paths through the graph are discovered. One path leads through the four red nodes, the other through the four violet nodes. Since red nodes have been observed in fourteen of the original reads, whereas the violet nodes have only been seen in three original reads, the path through the four red nodes is preferred over the path through the violet nodes.

1. **Guidance variants:** By supplying the Local realignment tool with a track of guidance variants. There are two modes for using the guidance variant track: either the 'un-forced' guidance mode (if the 'Force realignment to guidance-variants' is left un-ticked) or the 'forced' guidance mode (if the 'Force realignment to guidance-variants' is ticked). In the 'unforced' mode, 'pseudo-reads' are given to the local realignment algorithm representing the guidance variants, allowing the local realignment algorithm to explore the paths in the graph corresponding to these alignments. A scoring scheme where alignment to reference is preferred, is employed during first realignment pass, to determine the initial read support for the guidance variants. When more than one realignment pass is selected, the additional realignment passes are carried out using the standard scoring scheme where the most frequently used alignment path is preferred, and a supplementary limited realignment pass is performed in regions with guidance variants, to make up for the different scoring scheme used during first realignment pass. In the 'forced' mode, 'pseudo-references' are given to the local realignment algorithm representing the guidance variants, allowing the reads to be aligned to allele sequences of these in addition to the original reference sequence - with matches being awarded and encouraged equally much. The 'unforced' mode can be used with any guidance variant track as input. The 'force' mode should *only* be used with guidance variants for which there is strong prior evidence that they exist in the data (e.g., the 'InDel' track from the Structural Variants' tool (see Section 31.10) produced on the

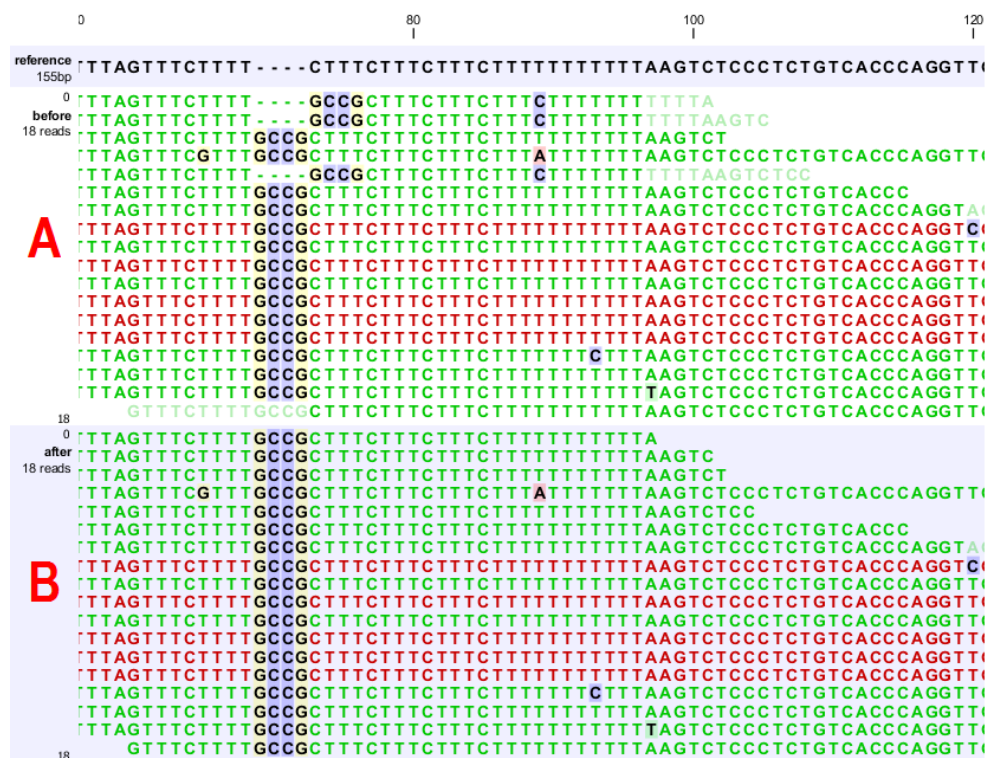


Figure 30.39: [A] The alignments of the first, second, and fifth read in this read mapping do not support the four-nucleotide insertion supported by the remaining reads. Additionally, the first, second, fifth and the last reads have unaligned ends. [B] After applying local realignment the first, second and fifth read consistently support the four-nucleotide insertion. Additionally, all previously unaligned ends have been realigned, because they perfectly match the reference sequence now (see also figure 30.36).

read mapping that is being aligned). Unless you do have strong evidence for the presence of these guidance variants, we do not recommend using the 'forced' mode as it can lead to the introduction of false positives in your alignment and all subsequent analyses.

2. **Concurrent local realignment of multiple samples:** Multiple input read mappings increase the chance to encounter at least one read mapped correctly. This guiding mechanism has been particularly designed for scenarios, where samples are known to be related, such as in family trials.

Figure 30.40 and figure 30.41 show examples that can be improved by guiding the local realignment algorithm.

30.4.4 Multi-pass local realignment

As described in section 30.4.1 the algorithm initially builds the realignment graph using the input read mapping. After the graph has been built the algorithm realigns individual reads based on information inferred from the realignment graph structure and its associated metadata. In some cases repetitive realignment iterations yield even more improvements, because with each realignment iteration the structure of the realignment graph changes slightly, potentially permitting further improvements. Local realignment therefore supports to perform multiple



Figure 30.40: [A] Three reads are misaligned in the presence of a four nucleotide insertion relative to the reference. [B] When applying local realignment without guidance the alignment is not improved. [C] Here local realignment is performed in the presence of the guiding variant track seen in (E). This enables the algorithm to consider alternative alignments, which are accepted whenever they have significant improvements over the original (as in read three that has a comparatively long unaligned-end). [D] If the alignment is performed with the option "Force realignment to guidance-variants" enabled, the realignment will be forced to realign according to the guiding variant track shown in (E), and this will result in realignment of all three reads. [E] The guiding variant track contains, amongst others, the four nucleotide insertion.

iterations implicitly. This is not only considered a convenience feature, but also saves a great deal of runtime by avoiding repeated transfers of large input data sets. For most samples local realignment will quickly saturate in the number of improvements. Generally, two realignment passes are strongly recommended. More than three passes rarely yield further improvements.

30.4.5 Known limitations

The major limitation of the local realignment algorithm is the necessity of at least one read being mapped correctly according to an indel present in the data. Insufficient alignment data results in suboptimal realignments or no realignments at all. As a work-around, local realignment can be guided by supplying a track of variants that enable the algorithm to determine improvements. Further guidance can be achieved by increasing the amount of alignment information and thereby increasing the chance to observe at least one read mapped correctly.

Reads are ignored, but retained in outputs, if:

- Lengths are longer than 50,000 base pairs.
- The alignment is longer than 50,000 base pairs.
- Crossing the boundaries of circular chromosomes.

Guiding variants are ignored, if:

- They are of type "Replacement".



Figure 30.41: [B] Three reads are misaligned in the presence of a four nucleotide insertion into the reference. Applying local realignment without guiding information would not yield any improvements (not shown). [C] Performing local realignment on both samples (A) and (B) enables the algorithm to improve the alignments of sample (B).

- They are longer than 200 bp (set as default value, but can be changed using the Maximum Guidance Variant Length parameter).
- If they are inter-chromosomal structural variations.
- If they contain ambiguous nucleotides.

30.4.6 Computational requirements

The realignment graph is produced using a sliding-window approach with a window size of 250,000 bp. If local realignment is run with multiple passes, then each pass has its own realignment graph. While memory consumption is typically below two gigabytes for single-pass, processor loads are substantial. Realigning a human sample of approximately 50x coverage will take around 24 hours on a typical desktop machine with four physical cores. Building the realignment graph and realignment of reads are parallelized actions, such that the algorithm scales very well with the number of physical cores. Server machines exploiting 12 or more physical cores typically run three times faster than the desktop with only four cores.

30.4.7 Run the Local Realignment tool

To run the **Local Realignment** tool, go to:

Tools | Resequencing Analysis (📁) | Local Realignment (🔍)

Select one or multiple read mappings as input. If one read mapping is selected, local realignment will attempt to realign all contained reads, if appropriate. If multiple read mappings are selected, their reference genome must exactly match. Local realignment will realign all reads from all input read mappings as if they came from the same input. However, local realignment will create one output read mapping for each input read mapping, thereby preserving the affiliation of each read to its sample. Clicking Next allows you to set parameters as displayed in figure 30.42.

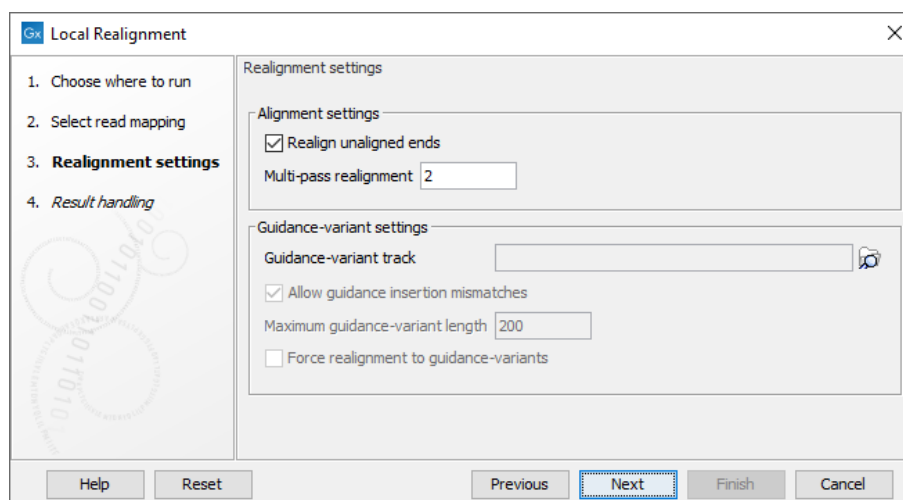


Figure 30.42: Set the realignment options.

Alignment settings

- **Realign unaligned ends** This option, if enabled, will trigger the realignment algorithm to attempt to realign unaligned ends as described in section "Realignment of unaligned ends (soft clipped reads)". This option should be enabled by default unless unaligned ends arise from known artifacts (such as adapter remainders in amplicon sequencing setups) and are thus not expected to be realignable anyway. Ignoring unaligned ends will yield a significant run time improvement in those cases. Realigning unaligned ends under normal conditions (where unaligned ends are expected to be realignable), however, does not contribute a lot of processing time.
- **Multi-pass realignment** This option is used to specify how many realignment passes should be performed by the algorithm. More passes improve accuracy at the cost of longer run time (approx. 25% per pass). Two passes are recommended; more than three passes barely yield further improvements.

Guidance-variant settings

- **Guidance-variant track** A track of variants to guide realignment of reads. Guiding can be used in at least two scenarios: (1) if reads are short or expected variants are long and (2) if cross sample comparisons are performed and some samples are already well genotyped. A track of variants can be produced by any of the variant detection tool, the Indels and Structural Variants tool or by importing variants from external data sources, such as dbSNP, etc.
- **Allow guidance insertion mismatches** This option is checked by default to allow reads to be realigned using guidance insertions that have mismatches relative to the read sequences.
- **Maximum Guidance Variant Length** set at 200 by default but can be increased to include guidance variants longer than 200 bp.
- There are two modes for using the guidance track:
 - **Un-forced** If the 'Force realignment to guidance-variants' is un-ticked the guidance variants are used as 'weak' prior evidence: the initial read support for each guidance

variant will be evaluated using a scoring scheme where alignment to reference is preferred. Any variant track may be used to guide the realignment when the un-forced mode is chosen.

- **Force realignment to guidance-variants** If the 'Force realignment to guidance-variants' is ticked the guidance variants are used as 'strong' prior evidence: a 'pseudo' reference will be generated for each guidance variant, and the alignment of nucleotides to their sequences will be awarded and encouraged as *much* as the alignment to the original reference sequence. Thus, the 'Force realignment to guidance-variants' options should *only* be used when there is prior information that the variants in the guidance variant track are infact present in the sample. This would e.g. be the case for an 'InDel' track produced by the Structural Variant tool (see Section 31.10), in an analysis of the same sample as the realignment is carried out on. Using 'forced' realignment to a general variant data base track is generally *strongly* discouraged.

The next dialog allows specification of the result handling. Under "Output options" it is possible to specify whether the results should be presented as a reads track or a stand-alone read mapping (figure 30.43).

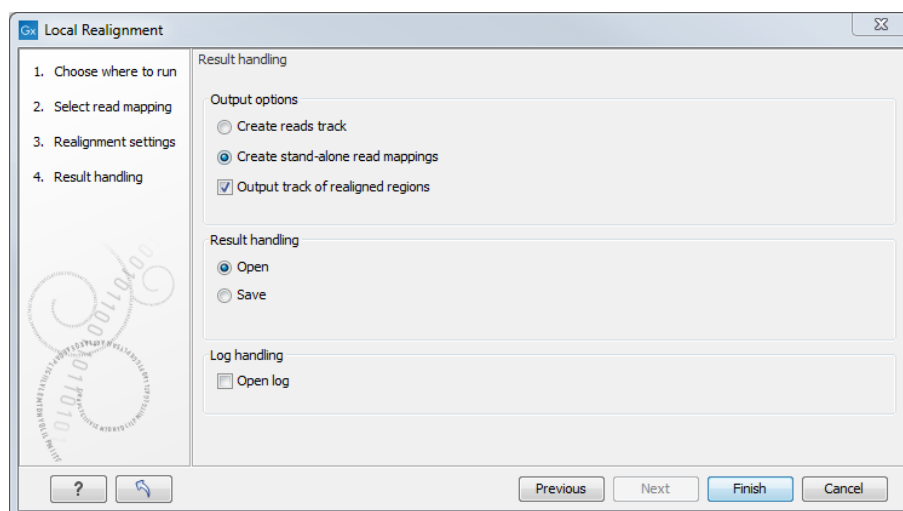


Figure 30.43: An output track of realigned regions can be created.

If enabled, the option **Output track of realigned regions** will cause the algorithm to output a track of regions that help pinpoint regions that have been improved by local realignment. This track has purely informative intention and cannot be used for anything else. Note: The Local Realignment tool is not recommended for Oxford Nanopore or PacBio long reads.

30.5 Merge Read Mappings

If you have performed two mappings with the same reference sequences, you can merge the results using the **Merge Read Mappings** (🔗). This can be useful in situations where you have already performed a mapping with one data set, and you receive a second data set that you want to have mapped together with the first one. In this case, you can run a new mapping of the second data set and merge the results:

Tools | Resequencing Analysis (🔗) | **Merge Read Mappings** (🔗)

This opens a dialog where you can select two or more mapping results, either in the form of tracks or read mappings. If the mappings are based on the same reference sequences (based on the name and length of the reference sequence), the reads will be merged into one mapping. If different reference sequences are used, they will simply be incorporated into the same result file (either a track or a mapping table).

The output from the merge can either be a track or standard mappings (equivalent to the read mapper's output, see section 30.1). For all the mappings that could be merged, a new mapping will be created. If you have used a mapping table as input, the result will be a mapping table. Note that the consensus sequence is updated to reflect the merge. The consensus voting scheme for the first mapping is used to determine the consensus sequence. This also means that for large mappings, the data processing can be quite demanding for your computer.

30.6 Remove Duplicate Mapped Reads

The purpose of this tool is to efficiently remove duplicate reads from a mapping, when duplicate reads have arisen due to the use of PCR amplification (or other enrichment) during sample preparation. This does not mean, however, that this tool should be used on all data that had an amplification step. In fact, use of this tool in the case of RNA-Seq data, amplicon data, or any sample where the start of a large number of reads are purposely at the same reference location, it is not recommended to make use of this tool. The tool may be used on mappings of single end reads, paired end reads or both.

A read duplication can be easily distinguished when mapping reads to a reference sequence as shown in the example in figure 30.44.

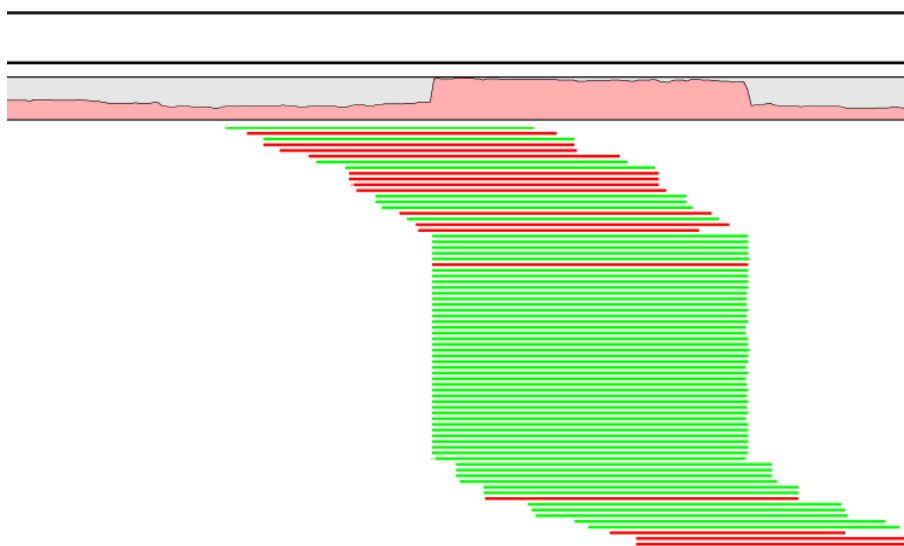


Figure 30.44: *Mapped reads with a set of duplicate reads, the colors denote the strand (green is forward and red is reverse).*

When sequencing library preparation involves a PCR amplification step, it is common to observe multiple reads where identical nucleotide sequences are disproportionately represented in the final results. Thus, to facilitate processing of mappings based on this kind of data, it may be necessary to perform a duplicate read removal step, which flags identical reads and subsequently

removes them from the data set. However, this step is complicated by the low, but consistent, presence of sequencing errors that may cause otherwise identical sequences to differ slightly. Thus, it is important that the duplicate read removal includes some tolerance for nearly identical sequences, which could still be reads from the same PCR artifact.

In samples that have been mapped to a reference genome, duplicate reads from PCR amplification typically result in areas of disproportionately high coverage and are often the cause of significant skew in allelic ratios, particularly when replication errors are made by the enzymes (e.g. polymerases) used during amplification. Sequencing errors incorporated post-amplification can affect both sequence- and coverage-based analysis methods, such as variant calling, where introduced errors can create false positive SNPs, and ChIP-Seq, where artificially inflated coverage can skew the significance of certain locations. By utilizing the mapping information, it is possible to perform the duplicate removal process rapidly and efficiently.

Note! We only recommend using the duplicate read removal if there are amplification steps involved in the library preparation. It is not recommended for RNA-Seq data, amplicon data, or any sample where the start of a large number of reads are purposely at the same reference location.

The method used by the duplicate read removal is to identify reads that share common coordinates (e.g. the same start coordinate), sequencing direction (or mapped strand) and the same sequence, these being the unifying characteristics behind sequencing reads that originate from the same amplified fragments of nuclear material. However, due to the frequent occurrence of sequencing errors, the tool utilizes simple heuristics to prune sequences with small variations from the consensus, as would be expected from errors observed in data from next-generation sequencing platforms. Base mismatch errors that were incorporated during amplification or prior to amplification will be indistinguishable from SNPs and may not be filtered out by this tool.

30.6.1 Algorithm details and parameters

The algorithm operates with the following assumption: Mapped reads from duplicated DNA fragments will share a mapping orientation (e.g. will map to the same strand), and depending on their orientation, will share either a start coordinate (forward reads), an end coordinate (reverse reads) or both (paired end reads).

The parameter "Maximum representation of minority sequence (percent)", if set to 20%, means that two groups of reads differing by one mismatch are duplicates when the smaller group contains at most 20% of the total number of reads in both groups.

The algorithm works by first making groups of reads that are exactly equal. Then pairs of these groups are merged if they are duplicated. If the groups have no mismatches, then they are equal except for read lengths and are always merged. If the groups have one mismatch, then the groups are merged if the small group is less than the bigger group by threshold %. The threshold is halved again for each additional mismatch symbol. If the initial threshold is 20%, then it is 10% for two mismatches, 5% for three mismatches, 2.5% for four mismatches and so on.

The output has one read per group of reads left after the merging steps. The read chosen as output for a group is the read with the highest average base quality score from the biggest exactly equal read group.

30.6.2 Running remove duplicate mapped reads

To run the **Remove Duplicate Mapped Reads** tool, go to:

Tools | Resequencing Analysis (📁) | **Remove Duplicate Mapped Reads** (📋)

This opens a dialog where you can select mapping results in reads tracks (📋) format. Clicking **Next** allows you to set the threshold parameters as displayed in figure 30.45.

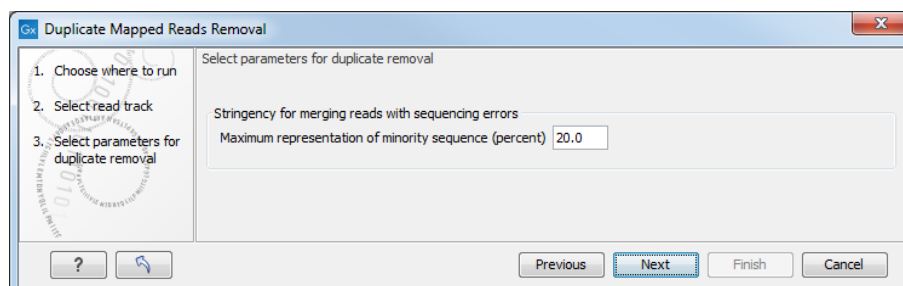


Figure 30.45: Setting the stringency for merging similar reads.

The parameter is explained in detail in section 30.6.1.

Clicking **Next** will reveal the output options. The main output is a list of the reads that remain after the duplicates have been removed. In addition, you can get the following output:

List of duplicate sequences These are the sequences that have been removed.

Report This is a brief summary report with the number of reads that have been removed (see an example in figure 30.46).

Note! The Remove Duplicate Mapped Reads tool may run this before or after local realignment. The order in which these two tools are run should make little if any difference.

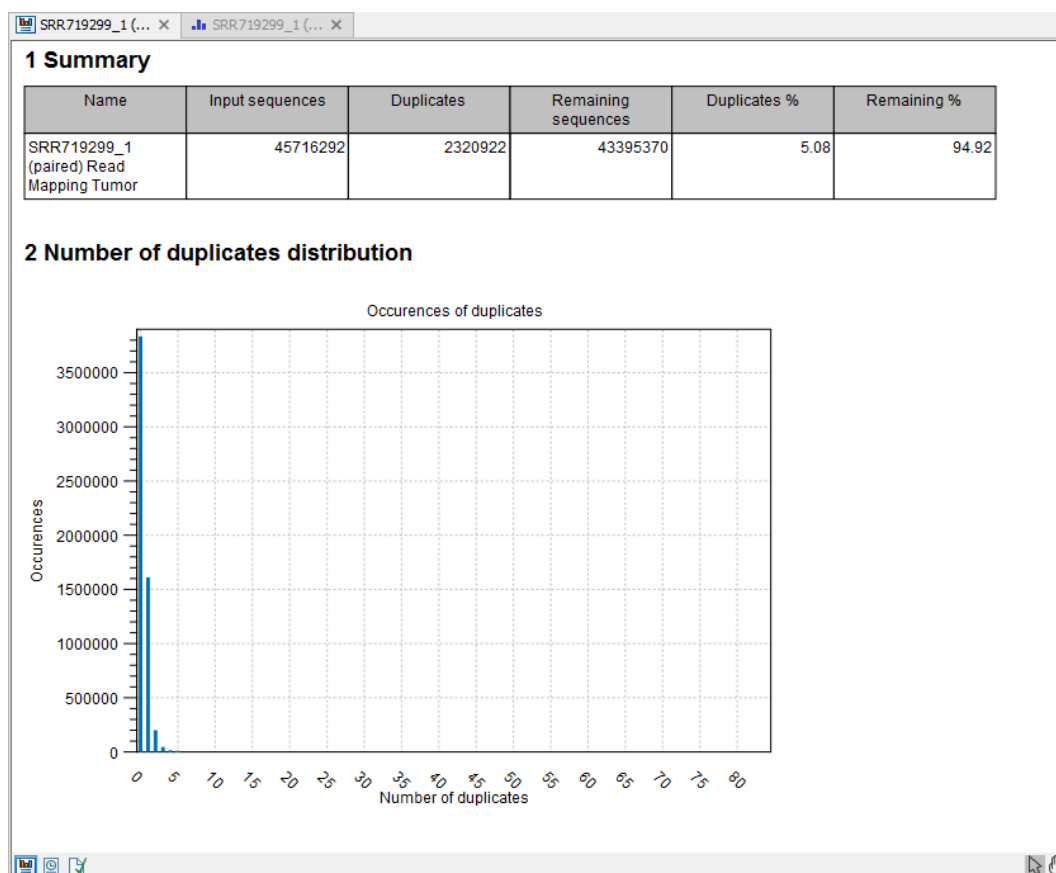


Figure 30.46: Summary statistics on the duplicate mapped reads.

30.7 Extract Consensus Sequence

Extract Consensus Sequence takes as input stand-alone read mappings, read tracks, or nucleotide BLAST results, and outputs a consensus sequence.

Consensus sequences can also be extracted directly from elements opened in the View Area by selecting **Extract New Consensus Sequence** (🔍) from the right-click menu on:

- For stand-alone read mappings: the reference sequence or a selection within it, or the consensus sequence.
- For nucleotide BLAST results: a selection within the query sequence.

To run the tool, go to:

Tools | Resequencing Analysis (🔍) | Extract Consensus Sequence (🔍)

The following options for extracting the consensus sequence can be configured (figure 30.47):

- **Threshold** Coverage above this value is considered high, while coverage at or below it is considered low. Overlapping paired-end reads count as two when calculating the coverage. A higher threshold yields a more reliable consensus but reduces completeness.
- **Low coverage handling** Options for handling regions with low coverage.

Figure 30.47: Options for consensus sequence extraction.

- **Remove regions with low coverage** No consensus is generated in regions of low coverage.
 - * **Join after removal** High coverage regions are joined into a single consensus sequence.
 - * **Split into separate sequences** Each high coverage region produces an individual consensus sequence.
- **Insert 'N' ambiguity symbols** The consensus is set to 'N' at each position with low coverage.
- **Fill from reference sequence** The consensus is set to the reference symbol at each position with low coverage.
- **Conflict resolution** Options for resolving read disagreements at individual positions within high coverage regions.
 - **Vote** The consensus is set to the most supported symbol (see **Use quality score**), excluding ambiguous symbols.
In case of a tie, symbols are chosen in the order: A > C > G > T.

To preserve biological heterozygous variation, see **Insert ambiguity codes**.

- **Insert ambiguity codes** The consensus is set to the IUPAC ambiguity code (section H) that best reflects the variation observed in the reads.

The following options determine which symbols contribute to the ambiguity codes:

- * **Noise threshold** Only symbols with support greater than this value (see **Use quality score**) contribute to the ambiguity code.
- * **Minimum nucleotide count** Only symbols present in at least this number of reads contribute to the ambiguity code.

Positions where no symbol qualifies to contribute to the ambiguity code are not included in the consensus.

- **Use quality score** The support of a symbol is determined by:
 - When unchecked: the percentage of reads containing the symbol.
 - When checked: the sum of the symbol's quality scores, expressed as a percentage of the total quality score at that position.

The following options for annotations the consensus sequence can be configured (figure 30.48):

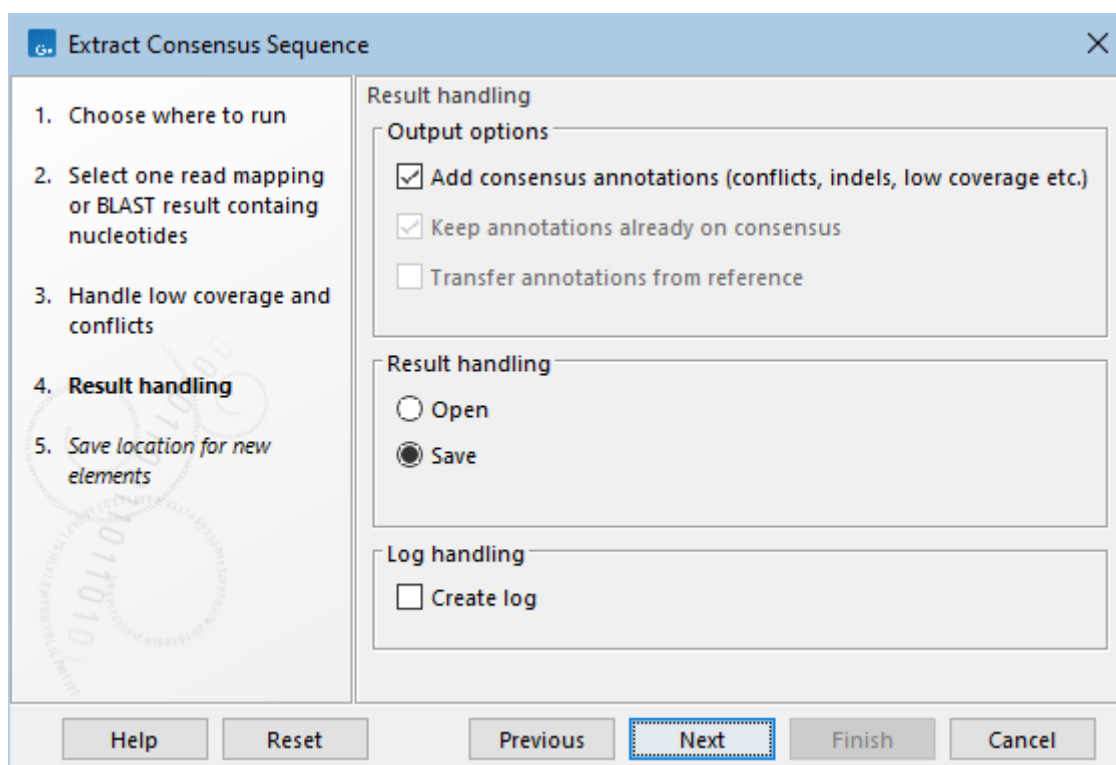


Figure 30.48: Options for adding annotations to the extracted consensus sequence.

- **Add consensus annotations (conflicts, indels, low coverage etc.)** When checked, annotations are added to the consensus sequence to indicate resolved conflicts, deletions relative to the reference, and low coverage regions, provided the **Split into separate sequences** option is not selected.

For inputs containing many reads or long references, many such annotations may be generated.

- **Keep annotations already on consensus** and **Transfer annotations from reference** When checked, annotations present on the consensus or on the reference in the input stand-alone read mapping are copied to the extracted consensus sequence. The copied annotations are placed in regions corresponding to their original location in the input data, although actual coordinates may differ. Annotations may be split if the **Split into separate sequences** option is selected.

These options are not enabled for types of input other than stand-alone read mapping.

Quality scores on the consensus sequence

When quality scores are present in the input, they are propagated to the extracted consensus as follows.

Consider a consensus symbol X , and let Y and Z be quality scores sums at its position, defined as:

- Y : The sum of quality scores from all reads.
- Z : The sum of quality scores from reads supporting X , defined as:
 - Reads containing the symbol X , if the **Vote** option is selected.
 - Reads containing symbols that contribute to the ambiguity code, if the **Insert ambiguity codes** option is selected.

The consensus symbol X is assigned a quality score $Q = Z - (Y - Z)$, bounded to a minimum of 0 and a maximum of 64.

Chapter 31

Variant detection




Contents

31.1 Variant Detection tools	836
31.1.1 Differences in the variants called by the different tools	837
31.1.2 How the variant detection tools work	840
31.1.3 Detailed information about overlapping paired reads	840
31.2 Fixed Ploidy Variant Detection	841
31.3 Low Frequency Variant Detection	843
31.4 Basic Variant Detection	843
31.5 Variant Detection - filters	844
31.5.1 General filters	844
31.5.2 Noise filters	846
31.6 Variant Detection - the outputs	852
31.6.1 Variant tracks	852
31.6.2 The annotated variant table	859
31.6.3 The variant detection report	860
31.7 Fixed Ploidy and Low Frequency Detection tools: detailed descriptions	861
31.7.1 Variant Detection - error model estimation	861
31.7.2 The Fixed Ploidy Variant Detection tool: Models and methods	862
31.7.3 The Low Frequency Variant Detection tool: Models and methods	866
31.8 Copy Number Variant Detection	869
31.8.1 The Copy Number Variant Detection (Targeted) tool	870
31.8.2 Region-level CNV track (Region CNVs)	876
31.8.3 Target-level CNV track (Target CNVs)	877
31.8.4 Gene-level annotation track (Gene CNVs)	879
31.8.5 How to interpret fold-changes when the sample purity is not 100%	880
31.8.6 CNV results report	880
31.8.7 CNV algorithm report	882
31.9 Identify Known Mutations from Sample Mappings	885
31.9.1 Run the Identify Known Mutations from Sample Mappings tool	886
31.9.2 Output from the Identify Known Mutations from Sample Mappings tool	888
31.10 InDels and Structural Variants	890

31.10.1 Run the InDels and Structural Variants tool	890
31.10.2 The Structural Variants and InDels output	894
31.10.3 The InDels and Structural Variants detection algorithm	897
31.10.4 Theoretically expected structural variant signatures	900
31.10.5 How sequence complexity is calculated	904
31.11 Structural Variant Caller for Long Reads	905
31.11.1 Structural Variant Caller for Long Reads settings	905
31.11.2 Structural Variant Caller for Long Reads output	907
31.11.3 Structural Variant Caller for Long Reads algorithm	911

31.1 Variant Detection tools

CLC Genomics Workbench offers three tools for detecting variants.

- **Fixed Ploidy Variant Detection** () - described in detail in section 31.2
- **Low Frequency Variant Detection** () - described in detail in section 31.3
- **Basic Variant Detection** () - described in detail in section 31.4

They are designed for the analysis of different types of samples and they differ in their underlying assumptions about the data, and hence in their assessments of when there is enough information in the data for a variant to be called. An overview of these differences is given in figure 31.1. None of the three tools are recommended for Oxford Nanopore or PacBio long reads.

Variant caller	Applications	Data	Variant detected	Comments	Examples of recommended applications
Fixed Ploidy	Germline variants	A sample for which the ploidy can be assumed known	Will detect variants whose representation in the reads is in accordance with the assumed ploidy	Will discard variants whose representation in the reads is likely due to sequencing errors or mapping artefacts	Germline variant calling
Low Frequency	Germline and somatic variants	A sample with unknown/mixed ploidy	Will detect variants whose representation in the reads is in accordance with the presence of a variant in a proportion of the reads	Will discard variants whose representation in the reads is likely due to sequencing errors	Any application where you are looking to detect low frequency (such as non-germline) variants
Basic	Any position that shows at least a specified number and frequency of nucleotide bases that differ from the reference base – irrespective of whether this may be explained by sequencing errors	Any	Will detect any variant observed in the reads	Will call a variant in any position that shows at least a specified number and frequency of nucleotide bases that differ from the reference base – irrespective of whether this may be explained by sequencing errors	exploratory read mapping applications - but not standard variant calling applications

Figure 31.1: An overview of the variant detection tools.

To run a variant detection tool, go to:

Tools | Resequencing Analysis () | **Variant Detection** ()

and choose the appropriate tool. In the first dialog of each detection tool, you are asked to specify the **reads track** or read mapping to analyze. The user is next asked to set the parameters that are specific for the variant detection tool. The three tools, their assumptions, and the tool-specific parameters are described later in their respective sections.

All variant detection tools will call:

- SNVs - single nucleotide variants
- MNVs - neighboring SNVs, where there is evidence they occur together
- small to medium-sized insertions and deletions - insertions and deletions fully represented within a single read
- replacements - neighboring SNVs and insertions or deletions

31.1.1 Differences in the variants called by the different tools

Because the tools differ in their underlying assumptions about the data, different variants may be called on the same data set using the same filter settings (see section 31.5). In general,

- the **Basic Variant Detection** tool calls the highest number of variants. It runs relatively quickly because it does not do any error-model estimation.
- the **Low Frequency Variant Detection** tool calls only a subset of the variants called by the Basic Variant Detection tool. The variants called by the Basic Variant Detection tool but **not** called by the Low Frequency Variant Detection tool usually originate from sequencing errors. The Low Frequency Variant Detection tool is the slowest of the three variant callers as it estimates an error-model and does not just consider variants within a specified ploidy model.
- the **Fixed Ploidy Variant Detection** tool calls a subset of the variants called by the Low Frequency Variant Detection tool. The variants called by the Low Frequency Variant Detection tool but **not** called by the Fixed Ploidy Variant Detection tool likely originate from mapping or sequencing errors.

The following examples show a Track list view of the variants detected by the three different variant detection tools for a particular data set with the same the filter settings. The top three variant tracks contain the results of the variant detection tools. The numbers of variants called are shown on the left side in brackets under the variant track names. The track 'basicV2' contains the results of the Basic Variant Detection tool, the track 'LowFreq' contains the results of the Low Frequency Variant Detection tool and the track 'FixedV2' contains the results of the Fixed Ploidy Variant detection tool. The other variant tracks display comparisons between results of the different tools. The particular comparisons is described in the name of each of these tracks.

Figure 31.2 highlights a variant reported by the Basic Variant Detection tool but not by the other variant detection tools. The information in the table view of the Basic Variant Detection results track ('basicV2') reveals that the variant is present at a low frequency (3 reads) in a high coverage position (209 reads), suggesting that is not a true variant but rather a sequencing error.

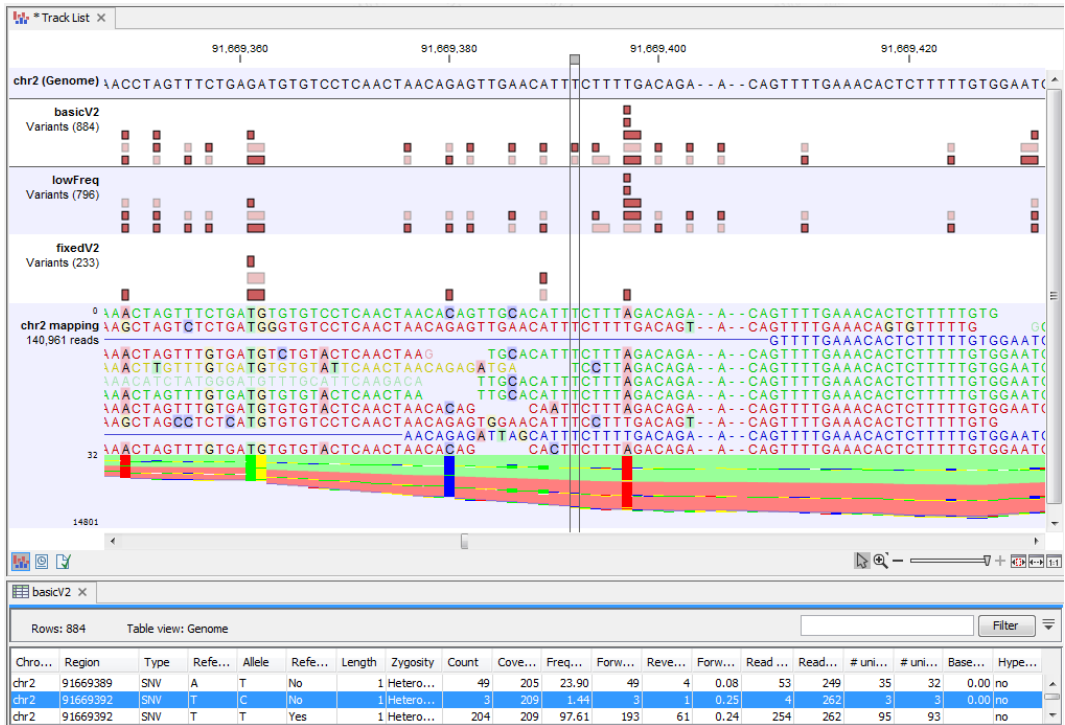


Figure 31.2: Case where a variant is detected only using the Basic Variant Detection tool.

Figure 31.3 shows variant calls produced by the three variant detection tool with the same data and general filter settings. As expected, the Basic Variant Detection tool reports the most variants (884), the Fixed Ploidy reports the fewest (233), and the Low Frequency Variant Detection tool detects a number between these two (796). But note that in the track named 'inLowFreqV2-notInBasicV2' that there are 9 variants reported by the Low Frequency Variant Detection tool that are not reported by the Basic Variant detection tool. It is because these variants are considered as several SNVs by the Low Frequency Variant Detection tool when they were part of a more complex MNV in the Basic Variant Detection results. In the case of the variant highlighted in figure 31.3, the Low Frequency Variant Detection calls for one variant in results track ('lowFreq'), while the Basic Variant Detection called a heterozygous 2 bp MNV in results track ('basicV2'). Here, the Low Frequency Variant Detection tool called only one of the two SNVs of that MNV. The second SNV of the MNV was not deemed to be supported by the evidence in the data when error modelling was carried out and so was not reported.



Figure 31.3: Case where variants can be detected as SNV by a tool and MNV by another.

Figure 31.4 shows a variant that is detected by both the Basic and the Low Frequency Variant Detection tools, but not by the Fixed Ploidy Variant Detection tool when a ploidy of 2 was specified. The information in the table view of the Low Frequency Variant Detection results track ('lowFreq') reveals that the highlighted variant is present in 29 reads in an area with coverage 204, a ratio inconsistent with what can be expected from a diploid sample, thus preventing the stringent Fixed Ploidy Variant Detection tool to call it as a variant. It is also unlikely that this variant was caused by sequencing error. The most likely explanation for the presence of this variant is that it originated from an error in the mapping of the reads. This happens if reads are mapped to a reference area that does not represent their true source, using for example an incomplete reference or one from a too distantly related organism.

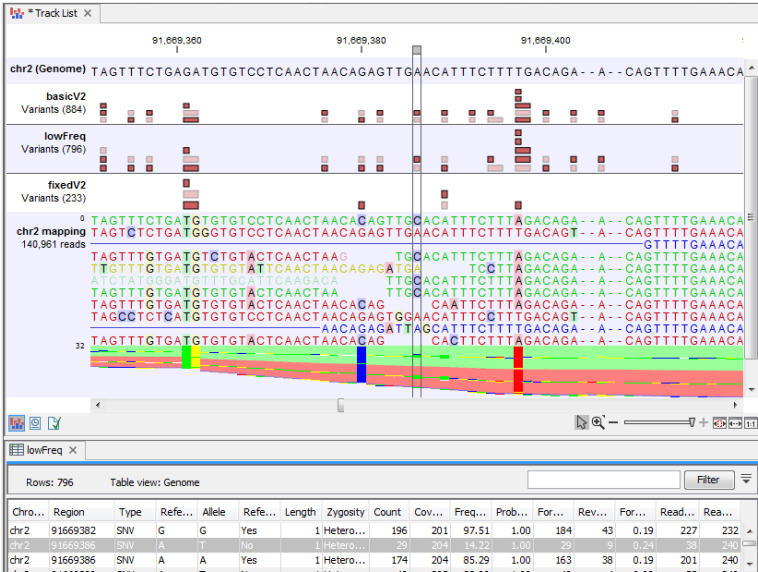


Figure 31.4: Case where a variant does not fit the ploidy assumption.

31.1.2 How the variant detection tools work

Each variant detection tool operates in a similar fashion, following successive and iterative steps while using common filters to call for variants. Before you start the tool, the wizard will take you through the different filters you can set to define which of the single polymorphisms detected should be called as a variant. The following sections describe the individual characteristics and the specific assumptions of the three variant detection tools. The filtering and output options common to the tools are described in detail in section 31.5 and section 31.6.

The steps of the Variant Detection tools are as follow:

1. The tool identifies all possible variants from either the total input dataset or a subset of it, depending on how the following filters have been set:
 - **Reference masking** settings select the areas of the mapping that should be inspected for variants. Note that variants extending up to 50 nt beyond a target region will be reported in full. Variants extending more than 50 nt beyond a target region will be trimmed to only include the first 50 nt beyond the target region.
 - **Read filter** settings select for the reads that should be considered in the assessment.
 - **Count and coverage filters** select for sites meeting coverage, frequency and absolute count requirements set for the analysis. Half the value of each parameter is used During the first stage of variant detection, when single position variants are initially being considered. This ensures that multiple position variants, which are built up from the single position variants, are not missed due to too stringent filtering early on. The full values for the cut-offs are applied later during the variant detection process.
 - **Noise filters** specify requirements for a read to be included, considering the quality and neighborhood composition of the area surrounding a potential variant.
2. At this stage, for the Fixed Ploidy and Low Frequency Variant Detection tools only, site-specific information is used to iteratively estimate error models. These error models are then used to distinguish true variants from likely sequencing errors. Potential single nucleotide variants are only be kept if the model containing the variant is significantly better than the model without the variant. Full details for the Fixed Ploidy Variant Detection tool are given in section 31.2 and 31.3.
3. The tool checks each position for other features such as read direction, base qualities and so on using the cut-off values specified in the **Noise filters** (see section 31.5).
4. The tool checks for complex variants by taking the single position variants identified in the steps above and checking if neighboring variants are present in *the same read*. If so, the tool 'joins' these SNVs into MNVs, longer insertions or deletions, or into replacements. Note that SNVs are joined only when they are present in the same read as this provides evidence that the variants appear contiguously in the sample.
5. Finally the tool applies the full cut-off values supplied for the **Count and coverage filters** to the single and multiple position variants obtained during the previous step.

31.1.3 Detailed information about overlapping paired reads

Paired reads that overlap introduce additional complexity for variant detection. This section describes how this is handled by *CLC Genomics Workbench*.

When it comes to **coverage** in the overlapping region, each pair is contributing once to the coverage. Even if there are indeed two reads in this region, they do not both contribute to coverage. The reason is that the two reads represent the same fragment, so they are essentially treated as one.

When it comes to counting the number of **forward and reverse reads**, including the forward/reverse reads balance, each read contribute. This is because this information is intended to account for systematic sequencing errors in one direction, and the fact that the two reads are from the same fragment is less important than the fact that they are sequenced on different strands.

If the two overlapping reads do not agree about the variant base, they are both ignored. Please note that there can be a special situation with the basic variant detection: If the two reads disagree, and one read does not pass the quality filter, the other read will contribute to the variant just as if there had been only that read and no overlapping pair.

31.2 Fixed Ploidy Variant Detection

The Fixed Ploidy Variant Detection tool relies on two models:

1. A model for the possible 'site-types' depends on the user-specified ploidy parameter: For a diploid organism there are two alleles and thus the site types are A/A, A/C, A/G, A/T, A/-, C/C, and so on until -/-.
2. A model for the sequencing errors that specifies the probabilities of having a certain base in the read but calling a different base. The error model is estimated from the data prior to calling the variants (see section 31.7.1).

The Fixed Ploidy algorithm will, given the estimated error model and the data observed in the site, calculate the probabilities of each of the site types. One of those site types is the site that is homozygous for the reference - that is, it stipulates that whatever differences are observed from the reference nucleotide in the reads is due to sequencing errors. The remaining site-types are those which stipulate that at least one of the alleles in the sample is different from the reference. The sum of the probabilities for these latter site types is the posterior probability that the sample contains at least one allele that differs from the reference at this site. We refer to this posterior probability as the 'variant probability'.

The Fixed Ploidy Variant Detection tool has two parameters: the 'Ploidy' and the 'Variant probability' parameters (figure 31.5):

- The 'ploidy' is the ploidy of the analyzed sample. The value that the user sets for this parameter determines the site types that are considered in the model. For more information about ploidy please see section 31.2.
- The 'Required variant probability' is the minimum probability value of the 'variant site' required for the variant to be called. Note that it is not the minimum value of the probability of the individual variant. For the Fixed Ploidy Variant detector, if a variant site - and not the variant itself - passes the variant probability threshold, then the variant with the highest probability at that site will be reported even if the probability of that particular variant might be less than the threshold. For example if the required variant probability is set to 0.9 then the individual probability of the variant called might be less than 0.9 as long as the probability of the entire variant site is greater than 0.9.

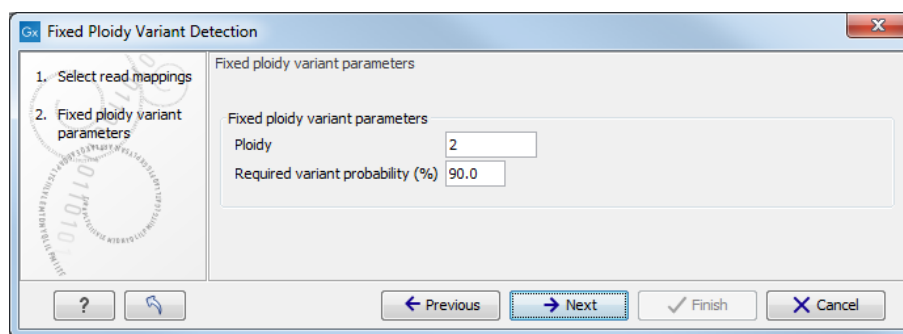


Figure 31.5: *The Fixed Ploidy Variant Detection parameters.*

As the Fixed Ploidy Variant Detection tool strongly depends on the model assumed for the ploidy, the user should carefully consider the validity of the ploidy assumption that he makes for his sample. The tool allows ploidy values up to and including 4 (tetraploids). For higher ploidy values the number of possible site types is too large for estimation and computation to be feasible, and the user should use the Low Frequency or Basic Variant Detection Tool instead.

Ploidy and sensitivity The Fixed Ploidy Variant Detection tool has two parameters. The ploidy level you set defines the statistical model that will be used during the variant detection analysis and thereby also defines what will be reported. The number of alleles that variant may have depends on the value that has been chosen for the ploidy parameter. For example, if you chose a ploidy of 2, then the variant at a site could be a homozygote (two alleles the same in the sample, but different to the reference), or a heterozygote (two alleles different than each other in the sample, with at least one of them different from the reference). If you had chosen a ploidy of three, then the variant at a site could be a homozygote (three alleles the same in the sample, but different to the reference), or a heterozygote (three alleles different than each other in the sample, with at least one of them different from the reference).

The variant probability parameter defines how good the evidence has to be at a particular site for the tool to report a variant at that location. If the *site* passes this threshold, then the *variant* with the highest probability at that site will be reported.

Sensitivity of the tool can be altered by changing these parameters: to increase sensitivity, you could decrease the variant probability setting - more sites are being reported - or increase the ploidy - adding extra allele types.

For example, a sample with a ploidy of 2 has many C and a few G at a particular location where the reference is a T. There is high enough evidence that the actual position is different than the reference, so the variant with the highest probability at this location will be reported. In the diploid model, all the possibilities will have been tested (e.g. A|A, A|C....C|C, C|G. C|T....and so on). In this example, C|C had the highest probability, and as long as the relative prevalence of Gs is low compared to Cs - that is, the probability of C|C stays higher than C|G - C|C will be reported. But in a case where the sample has a ploidy of 3, the model will test all the triploid possibilities (e.g. A|A|A, A|A|C, A|A|G....C|C|A, C|C|C, C|C|G.... and so on). For the same site, if the evidence in the reads results in the variant C|C|G having a higher probability than C|C|C, then it would be the variant reported. This shows that by increasing ploidy we have increased sensitivity of the tool, reporting a variant that represents the reads with G as well as the ones reporting a C at a particular position. Note: Oxford Nanopore and PacBio long reads are not recommended for this tool.

31.3 Low Frequency Variant Detection

As the Fixed Ploidy Variant Detection tool, the Low Frequency Variant Detection tool relies on

1. A statistical model for the analyzed sample and
2. A model for the sequencing errors.

A statistical test is performed at each site to determine if the nucleotides observed in the reads at that site could be due simply to sequencing errors, or if they are significantly better explained by there being one (or more) alleles. If the latter is the case, a variant corresponding to the significant allele will be called with an estimated frequency.

The Low Frequency Variant Detection tool has one parameter (figure 31.6):

- **Required Significance:** this parameter determines the cut-off value for the statistical test for the variant not being due to sequencing errors. Only variants that are at least this significant will be called. The lower you set this cut-off, the fewer variants will be called.

The Low Frequency Variant Detection tool is suitable for analysis of samples of mixed tissue types (such as cancer samples) in which low frequent variants are likely to be present, as well as for samples for which the ploidy is unknown or not well defined. The tool also calls more abundant variants, and can be used for analysis of samples with ploidy larger than four. Note that, as the tool looks for all variants, abundant as well as low frequency ones, analysis will generally be slower than those of the other variant detection tools. In particular it will be very slow - possibly prohibitively so - for samples with extremely high coverage, or a very large number of variants (as in cases where the sample differs considerably from the reference). Note: Oxford Nanopore and PacBio long reads are not recommended for this tool.

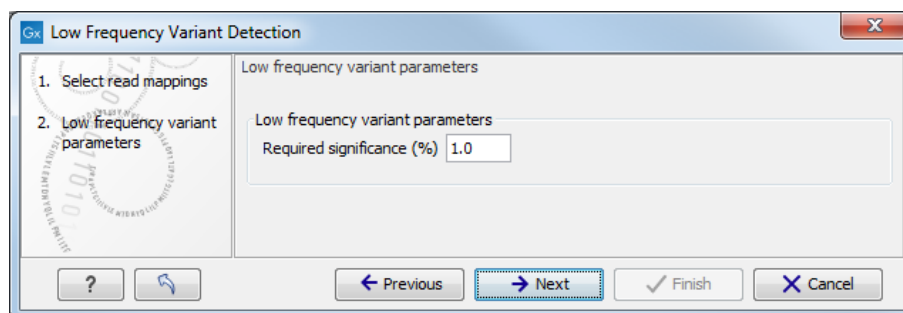


Figure 31.6: The Low Frequency Variant Detection parameters.

For a more in depth description of the Low Frequency Variant Detection tool see section 31.7).

31.4 Basic Variant Detection

The Basic Variant Detection tool does not rely on any assumptions on the data, and does not estimate any error model. It can be used on any type of sample. It will call a variant if it satisfies the requirements that you specify when you set the filters (see section 31.5). The tool has a single parameter (figure 31.7) that is specific to this tool: the user is asked to specify the 'ploidy' of the sample that is being analyzed. The value of this parameter does not have an impact on which variants are called - it will merely determine the contents of the 'hyper-allelic' column

that is added to the variant track table: variants that occur in positions with more variants than expected given the specified ploidy, will have 'Yes' in this column, other variants will have 'No' (see section 31.6 for a description of the outputs). Note: Oxford Nanopore and PacBio long reads are not recommended for this tool.

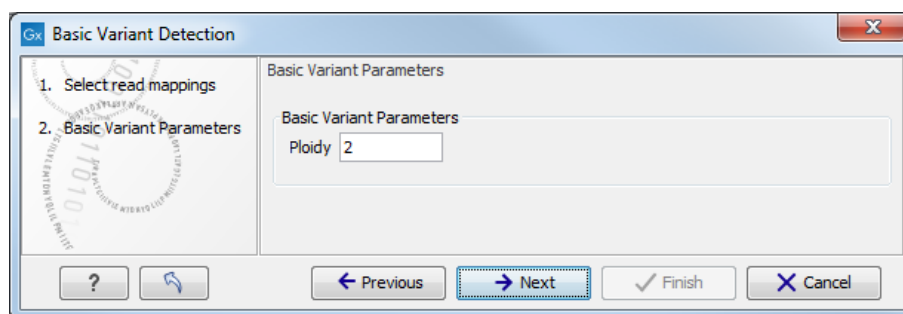


Figure 31.7: The Basic Variant Detection parameters.

31.5 Variant Detection - filters

The variant detectors offer a number of filters for which the user will set values in two wizard steps, the 'General filters' and the 'Noise filters'. Note that the tools add for most filters the values calculated and the bases filtered on as annotations to the variants (see section 31.6). This means that the filtering information is available in the variant track and that the user can choose to perform the filtering in a post-processing step from the variant track table rather than applying the filtering during the variant calling detection step.

The filters are described below.

31.5.1 General filters

The General filters relate to the regions and reads in the read mappings that should be considered, and the amount of evidence the user wants to require for a variant to be called (figure 31.8):

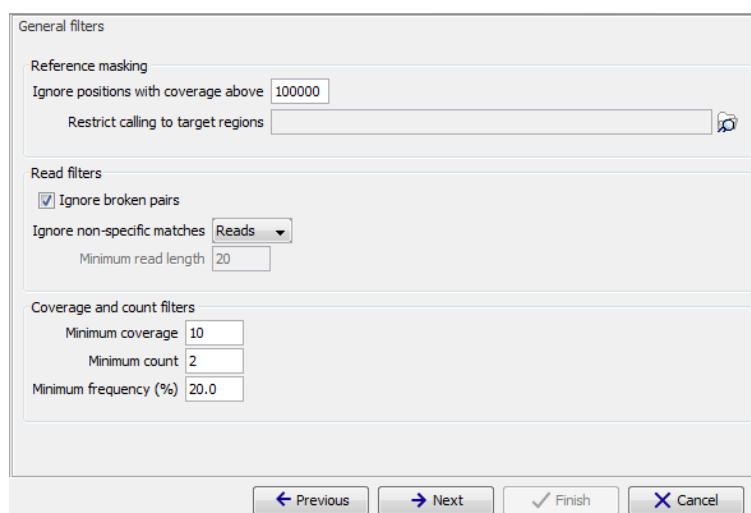


Figure 31.8: General filters. The values shown are those that are default for Fixed Ploidy Variant detection.

Note on the use of the Low Frequency Variant Detection tool with Whole Genome Sequencing

data: The default settings for the Low Frequency Variant Detection tool are optimized for targeted resequencing protocols, and NOT whole genome sequencing (e.g. cancer gene panels) where it is not uncommon to have modest coverage for most part of the mapping, and abnormal areas (typically repeats around the centromeres) with very high coverage. Looking for low frequency variants in high coverage areas will exhaust the machine memory because there will be many low frequency variants due to some reads originating from near identical repeat sequences or simple sequencing errors. In order to run the tool on WGS data the parameter **Ignore positions with coverage above** should be adjusted to a lower number (typically 1000).

Reference masking The 'Reference masking' filters allow the user to only perform variant calling (including error model estimation) in specific regions. In addition to selecting an annotation track, there are two parameters to specify:

- **Ignore positions with coverage above:** All positions with coverage above this value will be ignored when inspecting the read mapping for variants. The option is highly useful in cases where you have a read mapping which has areas of extremely high coverage as are areas around centromeres in whole genome sequencing applications for example.
- **Restrict calling to target regions:** Only positions in the regions specified will be inspected for variants. However, note that insertions situated directly to the right of a target region will also be included in the variant track because their reference allele is included inside the target.

Read filters The Read filters determine which reads (or regions) should be considered when calling the variants.

- **Ignore broken pairs:** When ticked, reads from broken pairs are ignored. Broken pairs may arise for a number of reasons, one being erroneous mapping of the reads. In general, variants based on broken pair reads are likely to be less reliable, so ignoring them may reduce the number of spurious variants called. However, broken pairs may also arise for biological reasons (e.g. due to structural variants) and if they are ignored some true variants may go undetected. Please note that ignored broken pair reads will not be considered for any non-specific match filters.
- **Non-specific match filter:** Non-specific matches are likely to come from repeat region whose exact mapping location is uncertain. In general, variants based on non-specific matches are likely to be less reliable. However, as there are regions in the genome that are entirely perfect repeats, ignoring non-specific matches may have the effect that true variants go undetected in these regions.

There are three options for specifying to which 'extent' the non-specific matches should be ignored:

- 'No': they are not ignored.
- 'Reads': they are ignored.
- 'Region': when this option is chosen no variants are called in regions covered by at least one non-specific match. In this case, the minimum length of reads that are allowed to trigger this effect has to be stated, as really short reads will usually be non-specific even if they do not stem from repeat regions.

Coverage and count filters These filters specify absolute requirements for the variants to be called. Note that suitable values for these filters are highly dependent on the coverage in the sample being analyzed:

- **Minimum coverage:** Only variants in regions covered by at least this many reads are called.
- **Minimum count:** Only variants that are present in at least this many reads are called.
- **Minimum frequency:** Only variants that are present at least at the specified frequency (calculated as 'count'/'coverage') are called.

These values are calculated for each of the detected candidate variants. If the candidate variant meets the specified requirements, it is called. Note that when the values are calculated, only the 'countable reads' - the reads chosen by the user to NOT be ignored - are considered. For example, if the user had specified to ignore reads from broken pairs, they will not be countable. This is also the case for non-specific reads, and for reads with bases at the variant position that does not fulfill the base quality requirements specified by the 'Base Quality Filter' (see the section on 'Noise filters' below). Also note that overlapping paired reads only count as one read since they only represent one fragment.

31.5.2 Noise filters

The 'Noise filters' examine each candidate variant at a more detailed level and filter out those that are likely the result of systematic errors and/or biases, such as those coming from samples preparation steps or sequencing protocol (figure 31.9). These filters should be used with care as there is always the risk of not calling a real variant that has the characteristics of a systematically induced one.

Figure 31.9: Noise filters.

Quality filters

- **Base quality filter:** The base quality filter can be used to ignore the reads whose nucleotide at the potential variant position is of dubious quality. This is assessed by considering the quality of the nucleotides in the region around the nucleotide position. There are three parameters to determine the base quality filter:
 - **Neighborhood radius:** This parameter determines the region size. For example if a neighborhood radius of five is used, a nucleotide will be evaluated based on the nucleotides that are 5 positions upstream and 5 positions downstream of the examined site, for a total of 11 nucleotides. Note that, near the end of the reads, eleven nucleotides will still be considered by offsetting the region relative to the nucleotide in question.
 - **Minimum central quality:** Reads whose central base has a quality below the specified value will be ignored. This parameter does not apply to deletions since there is no 'central base' in these cases.
 - **Minimum neighborhood quality:** Reads for which the minimum quality of the bases is below the specified value will be ignored.

Figure 31.10 gives an example of a variant called when the base quality filter is NOT applied, and not called when it is. When switching on the 'Show quality scores' option in the side panel of the reads it becomes visible that the reads that carry the potential 'G' variant tend to have poor quality. Note that the error in the example shown is a 'typical' Illumina error: the reference has a 'T' that is surrounded by stretches of 'G', the 'G' signals 'drowning' the signal of the 'T'. As all reads that have a base with quality less than 20 in this potential variant position are ignored when the 'Base quality filter' is turned on, no variant is called, most likely because it now does not meet the requirements of either the 'Minimum coverage', 'Minimum count' or 'Minimum frequency' filters.

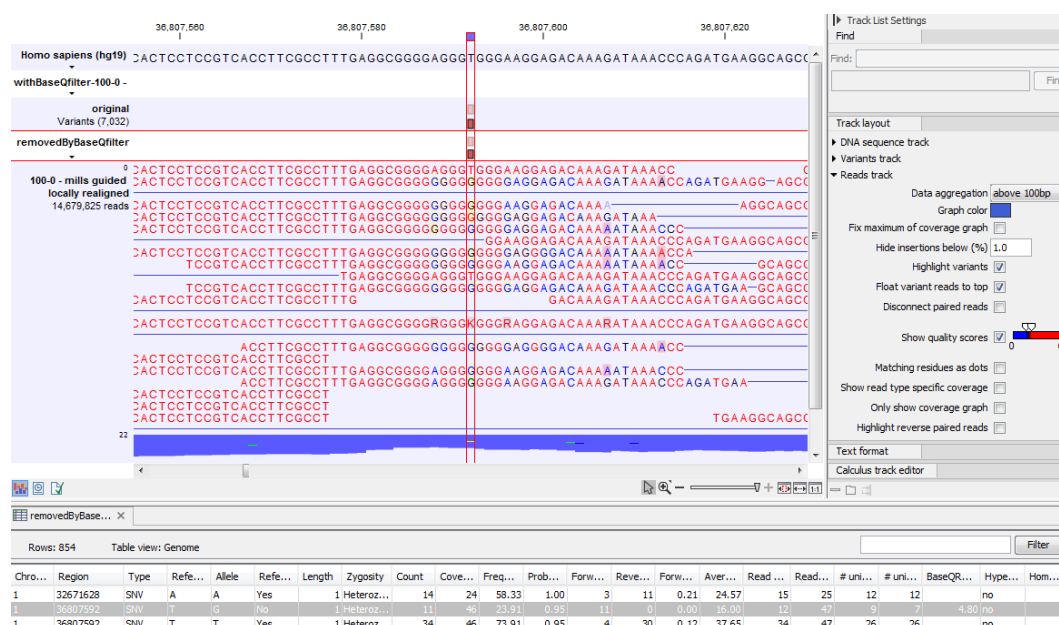


Figure 31.10: *Example of a variant called when the base quality filter is NOT applied, and not called when it is.*

Direction and position filters

Many sequencing protocols are prone to various types of amplification induced biases and errors. The 'Read direction' and 'Read position' filters are aimed at providing means for weeding out variants that are likely to originate from such biases.

- **Read direction filter:** The read direction filter removes variants that are almost exclusively present in either forward or reverse reads. For many sequencing protocols such variants are most likely to be the result of amplification induced errors. Note, however, that the filter is **NOT suitable for amplicon data**, as for this you will not expect coverage of both forward and reverse reads. The filter has a single parameter:
 - **Direction frequency:** Variants that are not supported by at least this frequency of reads from each direction are removed.
- **Relative read direction filter:** The relative read direction filter attempts to do the same thing as the 'Read direction filter', but does this in a statistical, rather than absolute, sense: it tests whether the distribution among forward and reverse reads of the variant carrying reads is different from that of the total set of reads covering the site. The statistical, rather than absolute, approach makes the filter less stringent. The filter has one parameter:
 - **Significance:** Variants whose read direction distribution is significantly different from the expected with a test at this level are removed. The lower you set the significance cut-off, the fewer variants will be filtered out.
- **Read position filter:** The read position filter is a filter that attempts to remove systematic errors in a similar fashion as the 'Read direction filter', *but* that is also **suitable for hybridization-based data**. It removes variants that are located differently in the reads carrying it than would be expected given the general location of the reads covering the variant site. This is done by categorizing each sequenced nucleotide (or gap) according to the mapping direction of the read and also where in the read the nucleotide is found; each read is divided in five parts along its length and the part number of the nucleotide is recorded. This gives a total of ten categories for each sequenced nucleotide and a given site will have a distribution between these ten categories for the reads covering the site. If a variant is present in the site, you would expect the variant nucleotides to follow the same distribution. The read position filter carries out a test for whether the read position distribution of the variant carrying reads is different from that of the total set of reads covering the site. The filter has one parameter:
 - **Significance:** Variants whose read position distribution is significantly different from the expected with a test at this level, are removed. The lower you set the significance cut-off, the fewer variants will be filtered out.

Figure 31.11 shows an example of a variant that is removed by the 'Read direction' filter. To see the direction of the reads, you must adjust the viewer settings in the 'Reads track' side panel to 'Show strands of paired reads'. Note that variant calling was done ignoring non-specific matches and broken pair reads, so only the 16 intact forward paired reads (the green reads) are considered. In this example there was no intact reverse reads.

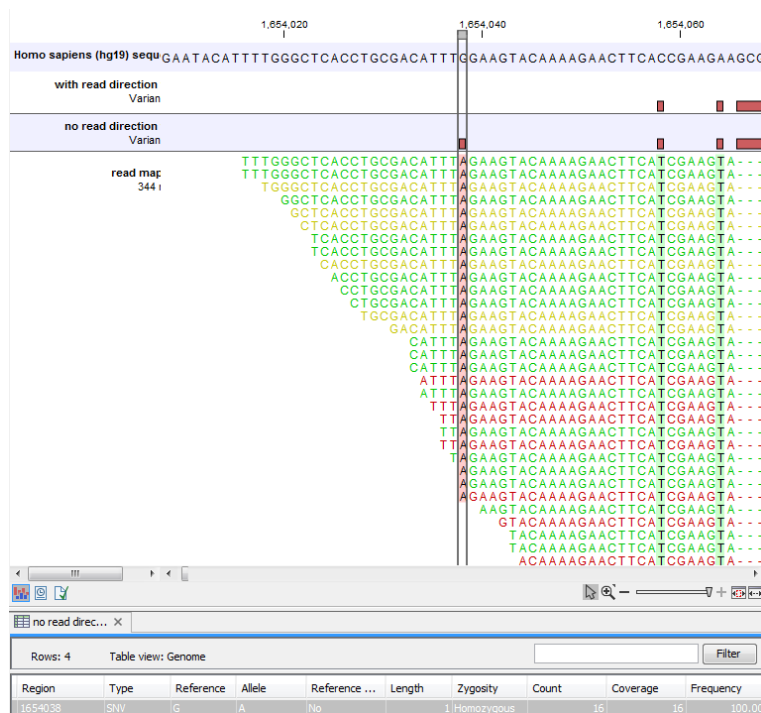


Figure 31.11: Example of a variant that is removed by the 'Read direction' filter.

Figure 31.12 shows an example of a variant that is removed by the 'Read position' filter, but not by the 'Read direction' filter. This variant is only seen in a set of reads having a similar start position, while reads that start in a different location do not contain this variant (e.g., none of the reads that start after position 186,641,600 carry the variant). This could indicate the incorporation of an incorrect base during the library preparation process rather than a true biological variant. The purpose of the 'Read position' filter is to reduce the presence of these types of variants. As with all noise filters, the more stringent the setting, the more likely you are to remove false positives and enrich your result for true positive variant calls but comes with the risk of filtering out true positives as well.

Understanding the type of false positive this filter is intended to remove will help you to determine what makes sense for your data set. For example, if your sequencing data did not include a PCR step or hybrid capture step, you may wish to use more lax settings for this filter (or not use it at all).

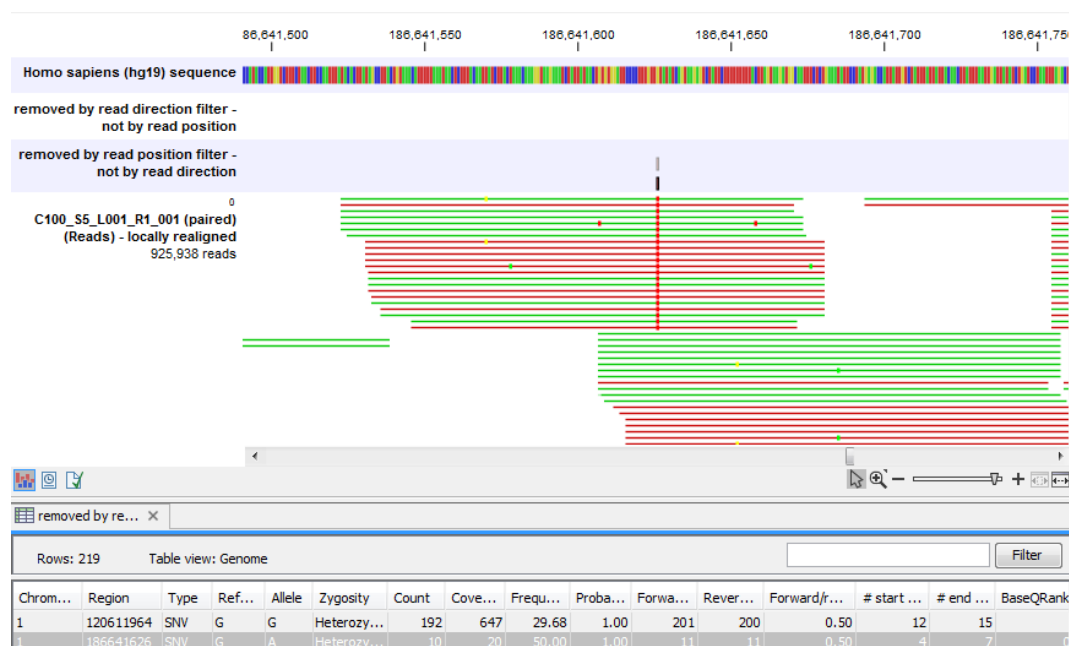


Figure 31.12: A variant that is filtered out by the Read position filter but not by the Read direction filter.

Technology specific filters

- **Remove pyro-error variants:** This filter can be used to remove insertions and deletions in the reads that are likely to be due to pyro-like errors in homopolymer regions. There are two types of such errors: They may occur either at (1) the immediate ends of homopolymer regions or (2) as an 'overspill' a few nucleotides downstream of a homopolymer region. In case (1) the exact numbers of the same number of nucleotide is uncertain and a sequence like "AAAAAAAA" is sometimes reported as "AAAAAAAAA". In case (2) a sequence like "CGAAAAAGTCG" may sometimes get an 'overspill' insertion of an A between the T and C so that the reported sequence is C "CGAAAAAGTACG". Note that the removal is done in the reads as a very first step, before calling the initial 1 bp variants.

There are two parameters that must be specified for this filter:

- **In homopolymer regions with minimum length:** Only insertion or deletion variants in homopolymer regions of at least this length will be removed.
- **With frequency below:** Only insertion or deletion variants whose frequency (ignoring all non-reference and non-homopolymer variant reads) is lower than this threshold will be removed.

Note that the higher you set the **With frequency below** parameter, the more variants will be removed. Figure 31.13 shows an example of a variant that is called when the pyro-error filter with minimum length setting 3 and frequency setting 0.5 is used, but that is filtered when the frequency setting is increased to 0.8. The variant has a frequency of 55.71.

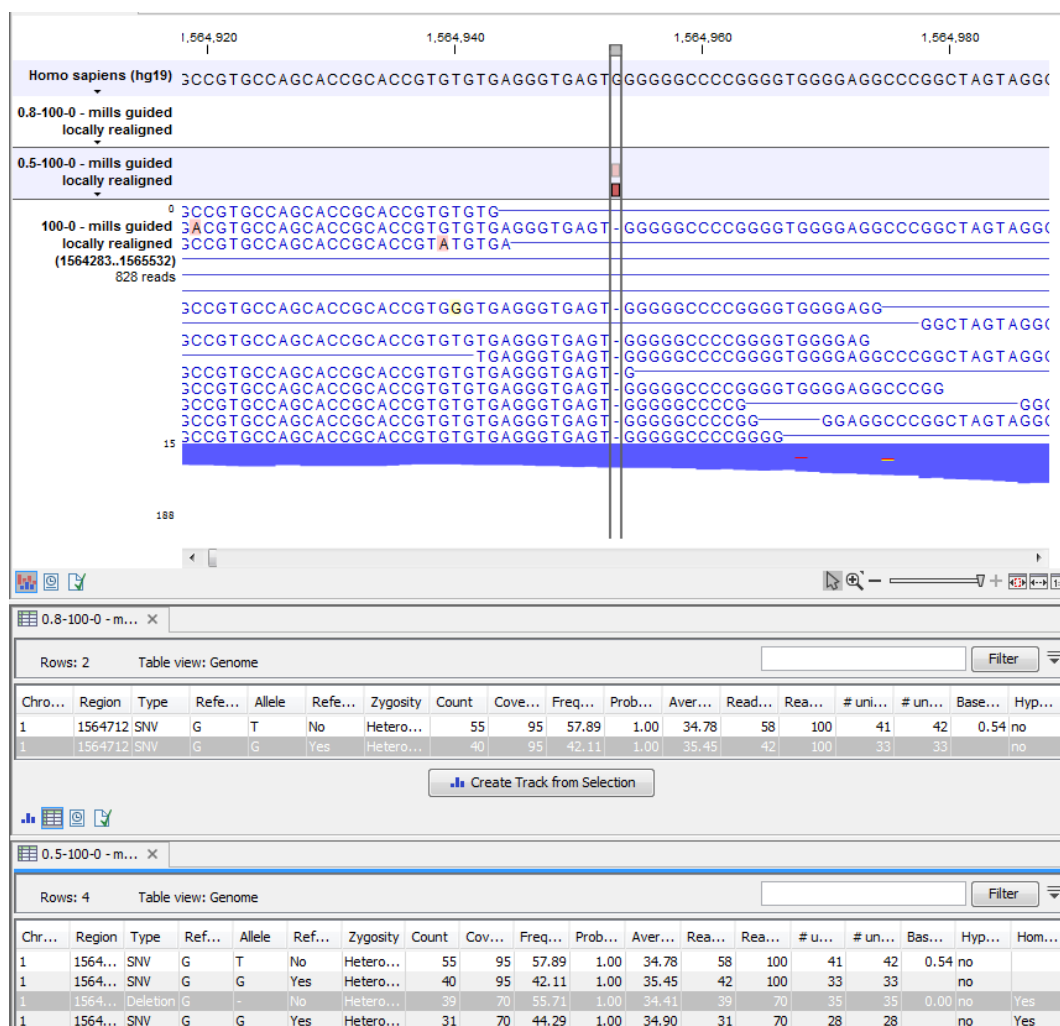


Figure 31.13: An example of a variant that is filtered out when the pyro-error filter is applied with settings 3 and 0.8, but not with settings 3 and 0.5.

In addition to the example above, a simple example is provided below in figure 31.14 to illustrate the difference between variant frequency and pyro-variant removal frequency (where non-reference and non-homopolymer variant reads are ignored).

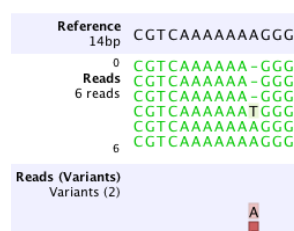


Figure 31.14: An example of a simple read mapping with 6 mapped reads. Three of them indicate a deletion, two match the reference, and one read is an A to T SNP

The read with the T variant is not counted when calculating the frequency for the homopolymer deletion, because we only want to estimate how often a homopolymer variant appears for a given

allele, and the T read is not from the same allele as the A and gap reads.

For the deletion, the variant frequency will be 50 percent, if it is reported. This is because it appears in 3 of 6 reads.

However, the pyro-variant removal frequency is 0.6, because it appears in 3 of 5 reads that come from the same allele. Thus the deletion will only be removed by the pyro-filter if the **With frequency below** parameter is above 0.6 and the **In homopolymer regions with minimum length** parameter is less than 7.

31.6 Variant Detection - the outputs

The Variant Detection Tools have the following outputs: a variant track, an annotated variant table and a report (figure 31.15). The report contains information on the estimated error model and, as only the Fixed ploidy and the Low Frequency Variant Detection tool uses an error model, the report is only available for those, and not for the Basic Variant Detection tool.

The outputs are described below. Note that this description can also be applied to variant files that were imported as GVF or VCF files - as described in section 7.2), or downloaded from external databases (e.g. dbSNP, HapMap, or 1000genomes) - as described in section 7.2)).

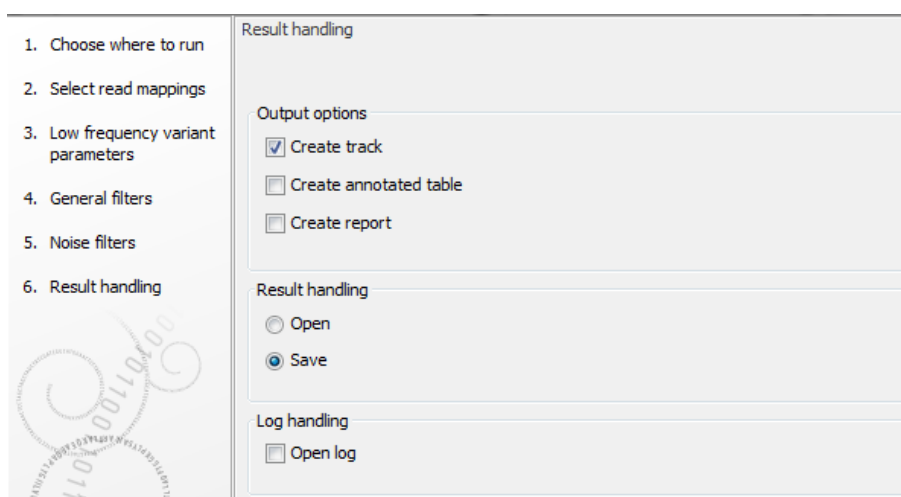


Figure 31.15: *Output options.*

31.6.1 Variant tracks

A variant track (figure 31.16) usually contains the following information for each variant:

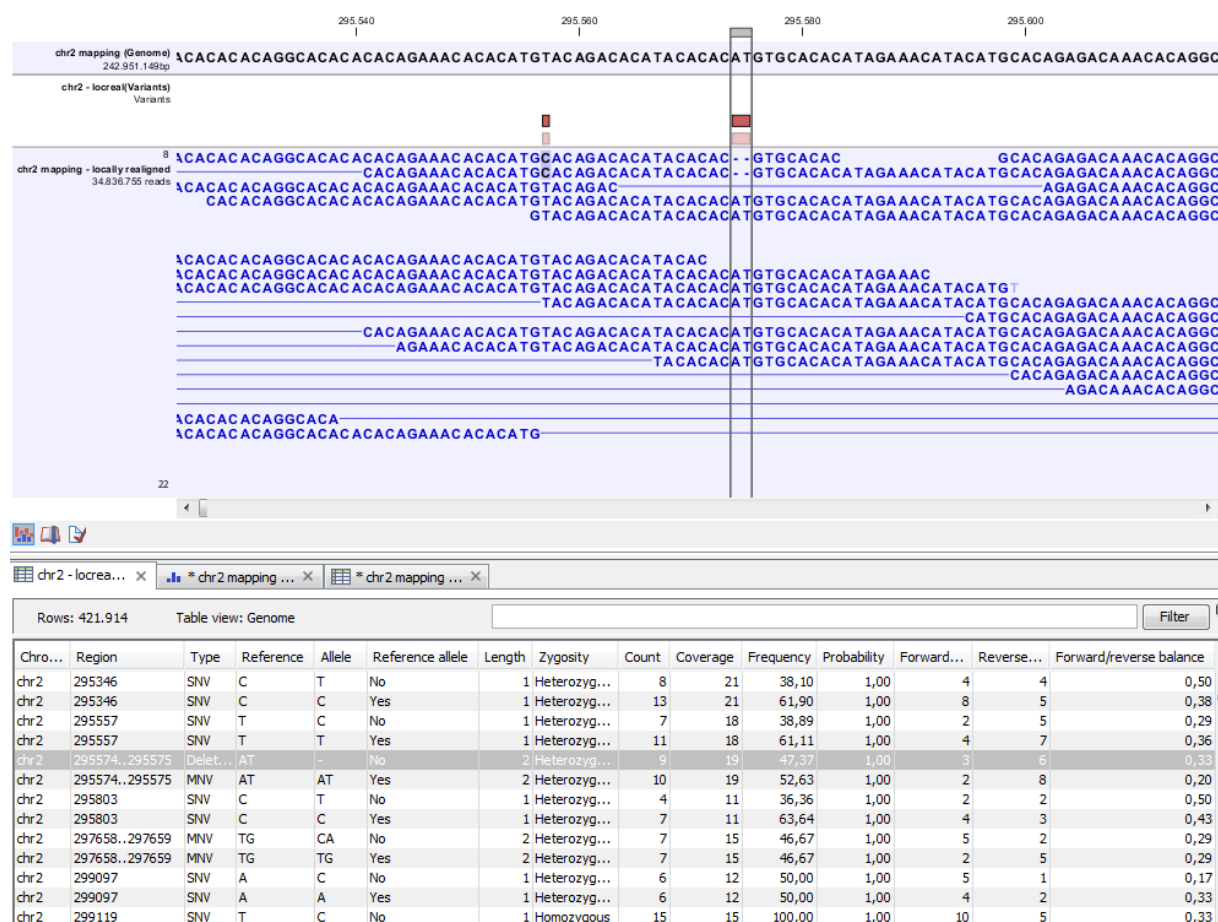


Figure 31.16: Variant track. The figure shows a track list (top), consisting of a reference sequence track, a variant track and a read mapping. The variant track has been opened in a separate table view by double-clicking on it in the track list. By selecting a row in the variant track table, the track list view is centered on the corresponding variant.

Chromosome The name of the reference sequence on which the variant is located.

Region The region on the reference sequence at which the variant is located. The region may be either a 'single position', a 'region' or a 'between position region'. Examples are given in figure 31.17.

Type Variants are classified into five different types:

- **SNV.** A single nucleotide variant. This means that one base is replaced by one other base. This is also often referred to as a SNP. SNV is preferred over SNP because the latter includes an extra layer of interpretation about variants in a population. This means that an SNV could potentially be a SNP but this cannot be determined at the point where the variant is detected in a single sample.
- **MNV.** This type represents two or more SNVs in succession.
- **Insertion.** This refers to the event where one or more bases are inserted in the experimental data compared to the reference.
- **Deletion.** This refers to the event where one or more bases are deleted from the experimental data compared to the reference.

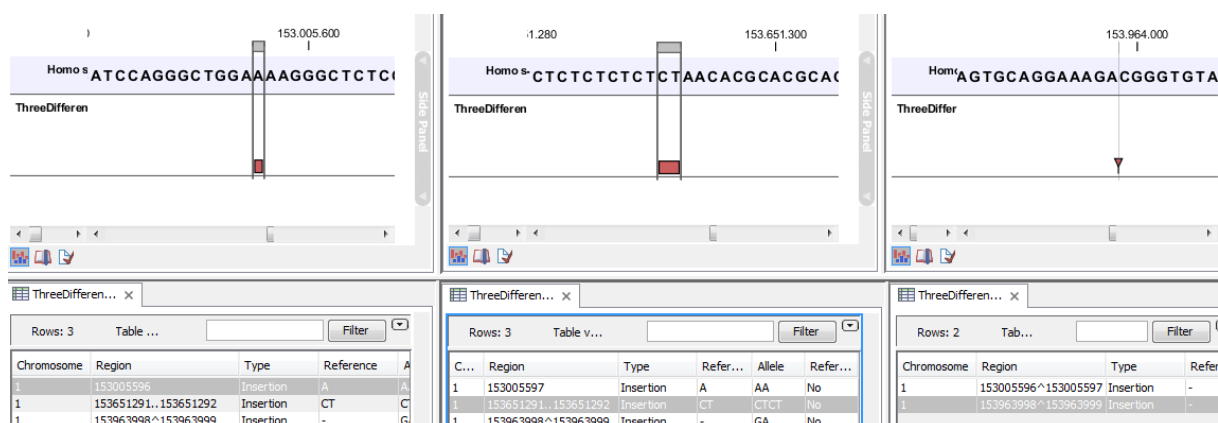


Figure 31.17: Examples of variants with different types of 'Region' column contents. The left-most variant has a 'single position' region, the middle variant has a 'region' region and the right-most has a 'between positions' region.

- **Replacement.** This is a more complex event where one or more bases have been replaced by one or more bases, where the identified allele has a length different from the reference (i.e., involving an insertion or deletion). Basically, this type represents variants that cannot be represented in the other four categories. An example could be `AAA->CC`. This cannot be resolved into a SNV or an MNV because the number of bases is different between the experimental data and the reference, it is not an insertion because something is also deleted from the reference, and it is not a deletion because something is also inserted.

Note about overlapping variants: If two different types of variants occur in the same location, these are reported separately in the output table. This is particularly important when SNVs occur in the same position as an MNV. Usually, multiple SNVs occurring alongside each other would simply be reported as one MNV, but if one SNV of the MNV is found in additional case samples by itself, it will be reported separately. For example, if an MNV of `AAT -> GCA` at position 1 occurs in five of the case samples, and the SNV at position 1 of `A -> G` occurs in an additional 3 samples (so 8 samples in total), the output table will list the MNV and SNV information separately. However, the SNV will be shown as being present in only 3 samples, as this is the number in which it appears "alone".

Reference The reference sequence at the position of the variant.

Allele The allele sequence of the variant.

Reference allele Describes whether the variant is identical to the reference. This will be the case one of the alleles for most, but not all, detected heterozygous variants (e.g. the variant detection tool might detect two variants, A and G, at a given position in which the reference is 'A'. In this case the variant corresponding to allele 'A' will have 'Yes' in the 'reference allele' column entry, and the variant corresponding to allele 'G' would have 'No'. Had the variant detection tool called the two variants 'C' and 'G' at the position, both would have had 'No' in the 'Reference allele' column).

Length The length of the variant. The length is 1 for SNVs, and for MNVs it is the number of allele or reference bases (which will always be the same). For deletions, it is the length of the deleted sequence, and for insertions it is the length of the inserted sequence. For

replacements, both the length of the replaced reference sequence and the length of the inserted sequence are considered, and the longest of those two is reported.

Zygosity The zygosity of the variant called, as determined by the variant detection tool. This will be either 'Homozygous', where there is only one variant called at that position or 'Heterozygous' where more than one variant was called at that position.

Count The number of 'countable' reads supporting the allele. The 'countable' reads are those that are used by the variant detection tool when calling the variant. Which reads are 'countable' depends on the user settings when the variant calling is performed - if e.g. the user has chosen 'Ignore broken pairs', reads belonging to broken pairs are not 'countable'. Note that, although overlapping paired reads have two reads in their overlap region, they only represent one fragment, and are counted only as one. (Please see the column 'Read count' below for a column that reports the value for 'reads' rather than for 'fragments'). Note also that the count value reported in the table may differ from the one accessible from the track's tooltip, as the 'count' value in the table is generated taking into account quality score and frequency of sequencing errors.

Coverage The fragment coverage at this position. Only 'countable' fragments are considered (see under 'Count' above for an explanation of 'countable' fragments). Note that, although overlapping paired reads have two reads in their overlap region, they only represent one fragment, and overlapping paired reads contribute only 1 to the coverage. (Please see the column 'Read coverage' below for a column that reports the value for 'reads' rather than for 'fragments'). Also see overlapping pairs in section 31.1.3 for how overlapping paired reads are treated.)

Frequency The number of 'countable' reads supporting the allele divided by the number of 'countable' reads covering the position of the variant ('see under 'Count' above for an explanation of 'countable' reads). Please see section 32.1.2 for a description of how to remove low frequency variants.

Forward and **Reverse read count** The number of 'countable' forward or reverse reads supporting the allele (see under 'Count' above for an explanation of 'countable' reads). Also see more information about overlapping pairs in section 31.1.3.

Forward and **Reverse read coverage** Coverage for forward or reverse reads supporting the allele.

Forward/reverse balance The minimum of the fraction of 'countable' forward reads and 'countable' reverse reads carrying the variant among all 'countable' reads carrying the variant (see under 'Count' above for an explanation of 'countable' reads). Some systematic sequencing errors can be triggered by a certain combination of bases. This means that sequencing one strand may lead to sequencing errors that are not seen when sequencing the other strand. In order to evaluate whether the distribution of forward and reverse reads is approximately random, this value is calculated as the minimum of the number of forward reads divided by the total number of reads and the number of reverse reads divided by the total number of reads supporting the variant. An equal distribution of forward and reverse reads for a given allele would give a value of 0.5. (See also more information about overlapping pairs in section 31.1.3.)

Average quality The average base quality score of the bases supporting a variant. The average quality score is calculated by adding the Q scores of the nucleotides supporting the variant,

and dividing this sum by the number of nucleotides supporting the variant. In the case of a deletion, the quality score reported is the lowest average quality of the two bases neighboring the deleted one. Similarly for insertions, the quality in reads where the insertion is absent is inferred from the lowest average of the two bases on either side of the position.

In rare cases, it can be possible that the quality score reported in this column for a deletion or insertion is below the threshold set for 'Minimum central quality', because this parameter is not applied to any quality value calculated from positions *outside* of the central variant. To remove low quality variants from the output, use the **Remove Marginal Variants** tool (see section 32.1.2).

If there are no values in this column, it is probably because the sequencing data was imported without quality scores (learn more about importing quality scores from different sequencing platforms in section 7.3).

Probability The contents of the Probability column (for Low Frequency and Fixed Ploidy Variant Detection tool only) depend on the variant detection tool that produced and the type of variant:

- In the Fixed Ploidy Variant Detection Tool, the probability in the resulting variant track's 'Probability' column is NOT the probability referred to in the wizard. The probability referred to in the wizard is the required minimum (posterior) probability that the site is NOT homozygous for the reference. The probability in the variant track 'Probability' column is the posterior probability of the particular site-type called. The fixed ploidy tool calculates the probability of the different possible configurations at each site. So using this tool, for single site variants the probability column just contains this quantity (for variants that span multiple positions see below).
- The Low Frequency Variant Detection tool makes statistical tests for the various possible explanations for each site. This means that the probability for the called variant must be estimated separately since it is not part of the actual variant calling. This is done by assigning prior probabilities to the various explanations for a site in a way that makes the probability for two explanations equal in exactly the situation where the statistical test shifts from preferring one explanation to the other. For a given single site variant, the probability is then calculated as the sum of probabilities for all the explanations containing that variant. So if a G variant is called, the reported probability is the sum of probabilities for these configurations: G, A/G, C/G, G/T, A/C/G, A/G/T, C/G/T, and A/C/G/T (and also all the configurations containing deletions together with G).

For multi position variants, an estimate is made of the probability of observing the same read data if the variant did not exist and all observations of the variant were due to sequencing errors. This is possible since a sequencing error model is found for both the fixed ploidy and rare variant tools. The probability column contains one minus this estimated probability. If this value is less than 50%, the variant might as well just be the result of sequencing errors and it is not reported at all.

Read count The number of 'countable' reads supporting the allele. Only 'countable' reads are considered (see under 'Count' above for an explanation of 'countable' reads). Note that each read in an overlapping pair contribute 1. To view the reads in pairs in a reads track as single reads, check the 'Show strands of paired reads' option in the side-panel of the reads track. (Please see the column 'Count' above for a column that reports the value for 'fragments' rather than for 'reads').

Read coverage The read coverage at this position. Only 'countable' reads are considered (see under 'Count' above for an explanation of 'countable' reads). Note that each read in an overlapping pair contribute 1. To view the reads in pairs in a reads track as single reads, check the 'Show strands of paired reads' option in the side-panel of the reads track. (Please see the column 'Coverage' above for a column that reports the value for 'fragments' rather than for 'reads').

Unique start positions The number of unique start positions for 'countable' fragments that support the variant. This value can be important to look at in cases with low coverage. If all reads supporting the variant have the same start position, you could suspect that it is a result of an amplification error. # Unique start positions is not provided for variants called from a mapping consisting of PacBio or Oxford Nanopore long reads.

Unique end positions The number of unique end positions for 'countable' fragments that support the variant. This value can be important to look at in cases with low coverage. If all reads supporting the variant have the same end position, you could suspect that it is a result of an amplification error. # Unique end positions is not provided for variants called from a mapping consisting of PacBio or Oxford Nanopore long reads.

BaseQRankSum The BaseQRankSum column contains an evaluation of the quality scores in the reads that have a called variant compared with the quality scores of the reference allele. Reference alleles and variants for which no corresponding reference allele is called do not have a BaseQRankSum value. The score is a z-score derived using the Mann-Whitney U test, so a value of -2.0 indicates that the observed qualities for the variant are two standard deviations below what would be expected if they were drawn from the same distribution as the reference allele qualities. A negative BaseQRankSum indicates a variant with lower quality than the reference variant, and a positive z-score indicates higher quality than the reference. BaseQRankSum is not provided for variants called from a mapping consisting of PacBio or Oxford Nanopore long reads.

Read position test probability The test probability for the test of whether the distribution of the read positions variant in the variant carrying reads is different from that of all the reads covering the variant position.

Read direction test probability Tests whether the distribution among forward and reverse reads of the variant carrying reads is different from that of all the reads covering the variant position. This value reflects a balanced presence of the variant in forward and reverse reads (1: well-balanced, 0: un-balanced). This p-value is based on a statistic that we assume follows a Chi-square(df=2) distribution under the null hypothesis of the variant having equal frequency on reads from both direction. Note that GATK uses a Fisher's exact test for the same purpose. The difference between both approaches lead to a potential overestimation of p-values output by the workbench's variant detection tools.

Hyper-allelic Basic and Fixed Ploidy Variant detectors only: Contains "yes", if the site contains more variants than the user-specified ploidy predicts, "no" if not.

Genotype Fixed Ploidy only: Contains the most probable genotype for the site.

Homopolymer The column contains "Yes" if the variant is likely to be a homopolymer error and "No" if not. This is assessed by inspecting all variants in homopolymeric regions longer than 2. A variant will get the mark "yes" if it is a homopolymeric length variation of the

reference allele, or a length variation of another variant that is a homopolymeric variation of the reference allele. When several overlapping homopolymeric variants are identified, all except the most frequent are marked as being homopolymer. However, if one of the overlapping, homopolymeric variants is the reference allele, then all of them are marked as homopolymer.

QUAL Measure of the significance of a variant, i.e., a quantification of the evidence (read count) supporting the variant, relative to the coverage and what could be expected to be seen by chance, given the error rates in the data.

The mathematical derivation of the value depends on the set of probabilities of generating the nucleotide pattern observed at the variant site (1) by sequencing errors alone and (2) under the different allele models of the variant caller allows. QUAL is calculated as $-10\log_{10}(1-p)$, p being the probability that a particular variant exists in the sample. QUAL is capped at 200 for $p=1$, with 200: highly significant, 0: insignificant. In rare cases, the QUAL value cannot be calculated for specific variant and as a result the QUAL field will be empty. A QUAL value of 10 indicates a 1 in 10 chance that the called variant is an error, while a QUAL of 100 indicates a 1 in 10^{10} chance that the called variant is an error.

Interpretation of logarithmically linked values

Average quality tells if the reads supporting a variant are likely to have the correct base call, while QUAL tells the confidence of the variant being present in the sample.

Both Average Quality and QUAL are logarithmically linked to error probabilities, so the interpretation of the values is similar.

Average Quality		Probability of incorrect base calls in the reads supporting the variant	Average base call accuracy in the reads supporting the variant
QUAL		Probability of incorrect variant call	Variant call accuracy
10	10	1 in 10	90%
20	20	1 in 100	99%
30	30	1 in 1,000	99.9%
60	60	1 in 1,000,000	99.9999%
-	100	1 in 10^{10}	99.99999999%
-	200	at least 1 in 10^{20}	at least 99.9999999999999999%

Please note that the variants in the variant track can be enriched with information using the annotation tools in section 32.2.

A variant track can be imported and exported in VCF or GVF formats. An example of the gvf-file giving rise to the variants shown in figure 31.17 is given in figure 31.18.


```
##gff-version 3
##gvf-version 1.06
##file-date 2013-09-23
##file-encoding windows-1252
1   CLC insertion  153005596  153005596  0   .   .   ID=CLC_1;Variant_seq=AA;Reference_seq=A;
1   CLC insertion  153651291  153651292  0   .   .   ID=CLC_2;Variant_seq=CTCT;Reference_seq=CT;
1   CLC insertion  153963999  153963998  0   .   .   ID=CLC_3;Variant_seq=GA;Reference_seq=-;
```

Figure 31.18: A gvf file giving rise to the variants in the figure above.

31.6.2 The annotated variant table

While the track table contains reference alleles and non-reference alleles, the annotated table lists only non-reference alleles. It include for each allele a subset of the columns of the variant track table and three additional columns (see figure 31.19).

Reference...	Type	Reference	Allele	Overlapping annotations	Coding region change	Amino acid change
3574524	SNV	T	C			
3574532	SNV	T	C			
3574536	SNV	T	C			
3575808	SNV	A	T	Gene: TEP1, mRNA: TEP1		
3655632	SNV	C	A	Gene: OSGEP, CDS: OSGEP, mRNA: OSGEP	NP_060277.1:c.681G>T	NP_060277.1:p.Glu227Asp
3655679	Deletion	A	-	Gene: OSGEP	NP_060277.1:c.637-3delT	
3655684	SNV	T	G	Gene: OSGEP	NP_060277.1:c.637-8A>C	
3656277	SNV	C	T	Gene: OSGEP, CDS: OSGEP, mRNA: OSGEP	NP_060277.1:c.597G>A	
3656304	SNV	T	C	Gene: OSGEP, CDS: OSGEP, mRNA: OSGEP	NP_060277.1:c.570A>G	

Figure 31.19: An example of an annotated variant table.

When the variant calling is performed on a read mapping in which gene and cds annotations are present on the reference sequence, the three columns will contain the following information:

Overlapping annotation This shows if the variant is covered by an annotation. The annotation's type and name will displayed. For annotated reference sequences, this information can be used to tell if the variant is found in a coding or non-coding region of the genome. **Note** that annotations of type `Variation` and `Source` are not reported.

Coding region change For variants that fall within a coding region of a gene, the change is reported according to the standard conventions as outlined in <https://hgvs-nomenclature.org/stable/>.

Amino acid change If the reference sequence of the mapping is annotated with ORF or CDS annotations, the variant detection tool will also report whether the variant is synonymous or non-synonymous. If the variant changes the amino acid in the protein translation, the new amino acid will be reported. The nomenclature used for reporting is taken from <https://hgvs-nomenclature.org/stable/>.

If the reference sequence has no gene and cds annotations these columns will have the entry "NA".

Try using a stand-alone reference with a stand-alone read mapping to avoid this situation. Also, the variant track may be enriched with information similar to that contained in the above three annotated variant table columns by using the track-based annotation tools in section 32.2).

The table can be **Exported** (📄) as a CSV file (comma-separated values) and imported into e.g. Excel. Note that the CSV export includes all the information in the table, regardless of filtering and what has been chosen in the **Side Panel**. If you only want to use a subset of the information, simply select and **Copy** (📋) the information.

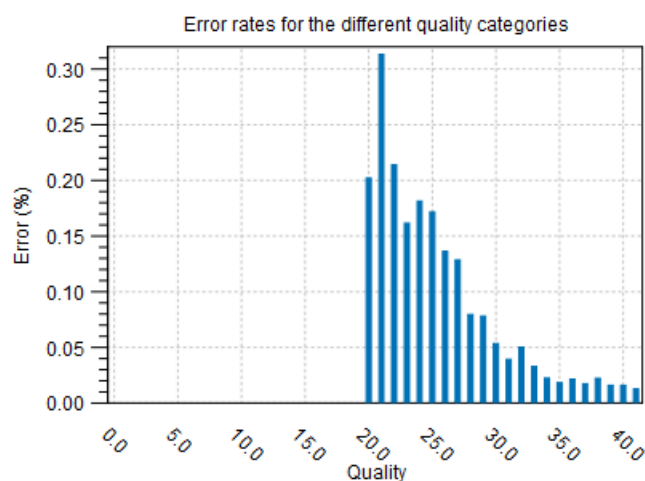
Note that if you make a split view of the table and the mapping (see section 2.1.4), you will be able to browse through the variants by clicking in the table. This will cause the view to jump to the position of the variant.

This table view is not well-suited for downstream analysis, in which case we recommend working with tracks instead (see section 31.6.1).

31.6.3 The variant detection report

In addition to the estimated error rates of the different types of errors shown in figure 31.21, the report contains information on the total error rates for each quality score as well as a distribution of the qualities of the individual bases in the reads in the read mapping, at the sites that were examined for variants (see figure 31.20).

1.1 Error rates for quality categories



1.2 Qualities of examined sites

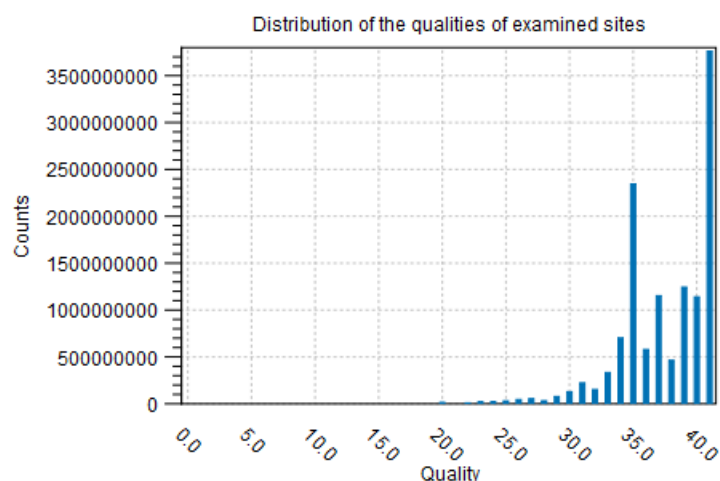


Figure 31.20: Part of the contents of the report on the variant calling.

31.7 Fixed Ploidy and Low Frequency Detection tools: detailed descriptions

This section provides a detailed description of the models, methods and estimation procedures behind the Fixed Ploidy and Low Frequency Variant Detection tools. For less detailed descriptions please see sections 31.2 and 31.3.

31.7.1 Variant Detection - error model estimation

The Fixed Ploidy and Low Frequency Variant Detection tools both rely on statistical models for the sequencing error rates. An error model is assumed and estimated for each quality score. Typically low quality read nucleotides will have a higher error rate than high quality nucleotides. In the error models, different types of errors have their own parameter, so if A's for example more often tend to result in erroneous G's than other nucleotides, that is also recognized by the error models. The parameters are all estimated from the data set being analyzed, so will adapt to the sequencing technology used and the characteristics of the particular sequencing runs. If there is not enough data to estimate a statistical model for a quality score number, the error model instead uses quality score error probabilities for all nucleotides. Information on the estimated error rates can be found in the Reports (see section 31.6 and figure 31.21).

1.5 Estimated frequencies of actual to called bases (quality scores: 20-29)

Called (across): Actual (below):	A	C	G	T	N	-
A	99.828	0.015	0.086	0.023	0.044	0.005
C	0.050	99.854	0.034	0.043	0.017	0.002
G	0.043	0.011	99.897	0.029	0.017	0.003
T	0.026	0.050	0.032	99.868	0.021	0.003
-	0.000	0.000	0.000	0.000	0.000	100.000

Number of sequenced bases with quality scores 20-29: 382,854,867

1.6 Estimated frequencies of actual to called bases (quality scores: 30-39)

Called (across): Actual (below):	A	C	G	T	N	-
A	99.979	0.001	0.008	0.003	0.008	0.001
C	0.010	99.976	0.002	0.008	0.002	0.001
G	0.008	0.001	99.974	0.012	0.004	0.001
T	0.003	0.008	0.002	99.983	0.003	0.001
-	0.000	0.000	0.000	0.000	0.000	100.000

Number of sequenced bases with quality scores 30-39: 7,400,088,878

Figure 31.21: Example of estimated error rates estimated from a whole exome sequencing Illumina data set.

The figure shows average estimated error rates across bases in the given quality score intervals (20-29 and 30-39, respectively). As expected, the estimated error rates (that is, the off-diagonal elements in the matrices in the figure) are higher for bases with lower quality scores. Note also that although the matrices in the figure show error rates of bases within *ranges* of quality scores, a separate matrix is estimated for each quality score in the error model estimation.

31.7.2 The Fixed Ploidy Variant Detection tool: Models and methods

This section describes the model, method and estimation procedure behind the Fixed Ploidy Variant Detection tool. The Fixed Ploidy Variant Detection tool is designed for detecting variants in samples for which the ploidy is known. As the Fixed Ploidy Variant Detection tool assumes, and hence can exploit, information about underlying possible allele type sites, this variant caller has particularly high *specificity* for samples for which the ploidy assumption is valid.

The Fixed Ploidy Variant Detection tool

The purpose of the Fixed Ploidy Variant Detection tool is to call variants in samples with known ploidy from read mapping data. It can detect variants in samples from haploid (e.g. bacteria), diploid (e.g. human) and polyploid (upto tetraploid) organisms (e.g. higher plants). It detects Single Nucleotide Variants (SNVs), MNVs (Multiple Nucleotide Variants), insertions, deletions as well as replacements (combinations of neighboring insertions, deletions and SNVs for which the positions are ambiguous).

The algorithm behind the Fixed Ploidy Variant Detection tool combines a Bayesian model with a Maximum Likelihood approach. Variants are called by examining the posterior probabilities from the Bayesian model: at any given site a variant is called if the sum of the posterior probabilities of the site types that are different from the homozygous reference allele site type is larger than the user-specified 'probability' cut-off value. The variant called is the variant that corresponds to the site type with the highest posterior probability. When evaluating the posterior probabilities in the Bayesian model, maximum likelihood estimates for the parameters of the model are used. These are obtained from the likelihood function using an Expectation Maximization (EM) approach.

The model for the Fixed Ploidy Variant Detection tool

The statistical model for the Fixed Ploidy Variant Detection tool consists of a model for the **possible site types**, S , and their prior probabilities, $f_s, s \in S$, and for **the sequencing errors**, e .

Prior site type probabilities: The set of possible site types is determined entirely by the assumed ploidy, and consists of the set of possible underlying nucleotide allele combinations that can exist within an individual with the specified number of alleles. E.g. if the specified ploidy is 2, the individual has two alleles, and the nucleotide at each allele can either be an A , a C , a G , a T or a $-$. The set of possible types for the diploid individual's sites is thus:

$$S = \{A/A, A/C, A/G, A/T, A/-, C/C, C/G, C/T, C/-, G/G, G/T, G/-, T/T, T/-, -/-\}.$$

Note that, as we cannot distinguish the alleles from each other there are not $5 \times 5 = 25$ possible site types, but only 15 (that is, the allele combination A/C is indistinguishable from the allele combination C/A).

We let f_s denote the prior probabilities of the site types $s \in S$. The prior probabilities of the site types are the frequencies of the true site types in the mapping. The values of these are unknown, and need to be estimated from the data.

Error probabilities: The model for the sequencing errors describes the probabilities with which the sequencing machine produces the nucleotide M , when what it should have produced was the nucleotide N , (M and $N \in \{A, C, G, T, -\}$). When quality values are available for the nucleotides in the reads, we let each quality value have its own error model; if

not, a single model is assumed for all nucleotides. Each error model has the following 25 parameters:

$$\{e(N \rightarrow M) | N, M \in \{A, C, G, T, -\}\}.$$

The values of these parameters are also unknown, and hence also need to be estimated from the data.

Deriving the posterior probabilities of the site types

We will call a variant at a site if the sum of the posterior probabilities of the non-homozygous reference site types is larger than the user-specified cut-off value. For this we need to be able to calculate the posterior site type probabilities. We here derive the formula for these.

Using the Bayesian approach we can write the posterior probability of a site type, t , as follows:

$$\begin{aligned} P(t|data) &= \frac{P(data|t)P(t)}{P(data)} \\ &= \frac{P(data|t)P(t)}{\sum_{s \in S} P(data|s)P(s)}, \end{aligned} \quad (31.1)$$

where $P(t)$ is the prior probability of site type t (that is, $f_s, s \in S$, from above) and $P(data|t)$ is the likelihood of the data, given the site type t . The data consists of all the nucleotides in all the reads in the mapping. For a particular site, assume that we have k reads that cover this site, and let i be an index over the nucleotides observed, n_i , in the reads at this site. We thus have:

$$P(data|t) = P(n_1, \dots, n_k|t).$$

To derive the likelihood of the data, $P(n_1, \dots, n_k|t)$, we first need some notation: For a given site type, t , let $P_t(N)$ be the probability that an allele from this site type has the nucleotide N . The $P_t(N)$ probabilities are known and are determined by the ploidy: For a diploid organism, $P_t(N) = 1$ if t is a homozygous site and N is one of the alleles in t , whereas it is 0.5 if t is a heterozygous and N is one of the alleles in t , and it is 0, if N is not one of the alleles in t . For a triploid organism, the $P_t(N)$ will be either 0, 1/3, 2/3 or 1.

With this definition, we can write the likelihood of the data n_1, \dots, n_k in a site t as:

$$P(n_1, \dots, n_k|t) = \prod_{i=1}^k \sum_{N \in \{A, C, G, T, -\}} P_t(N) \times e_q(N \rightarrow n_i). \quad (31.2)$$

Inserting this expression for the likelihood, and the prior site type frequencies f_s and f_t for $P(s)$ and $P(t)$, in the expression for the posterior probability (31.1), we thus have the following equation for the posterior probabilities of the site types:

$$\begin{aligned} P(t|n_1, \dots, n_k) &= \frac{P(n_1, \dots, n_k|t)f_t}{\sum_{s \in S} P(n_1, \dots, n_k|s)f_s} \\ &= \frac{\prod_{i=1}^k \sum_{N \in \{A, C, G, T, -\}} P_t(N) \times e_q(N \rightarrow n_i)f_t}{\sum_{s \in S} \prod_{i=1}^k \sum_{N \in \{A, C, G, T, -\}} P_s(N) \times e_q(N \rightarrow n_i)f_s} \end{aligned} \quad (31.3)$$

The unknowns in this equation are the prior site type probabilities, $f_s, s \in S$, and the error rates $\{e(N \rightarrow M)|N, M \in \{A, C, G, T, -\}\}$. Once these have been estimated, we can calculate the posterior site type probabilities using the equation 31.3 for each site type, and hence, for each site, evaluate whether the sum of the posterior probabilities of the non-homozygous reference site types is larger than the cut-off. If so, we will set out current estimated site type to be that with the highest posterior probability.

Estimating the parameters in the model for the Fixed Ploidy Variant Detection tool

The Fixed Ploidy Variant Detection tool uses the Expectation Maximization (EM) procedure for estimating the unknown parameters in the model, that is, the prior site type probabilities, $f_s, s \in S$ and the error rates $\{e(N \rightarrow M)|N, M \in \{A, C, G, T, -\}\}$. The EM procedure is an iterative procedure: it starts with a set of initial prior site type frequencies, $f_s^0, s \in S$ and a set of initial error probabilities, $\{e_q^0(N \rightarrow M)|N, M \in \{A, C, G, T, -\}\}$. It then iteratively updates first the prior site type frequencies (to get $f_s^1, s \in S$), then the error probabilities (to get $\{e_q^1(N \rightarrow M)|N, M \in \{A, C, G, T, -\}\}$), then the site type frequencies again, etc. (a total of four rounds), in such a manner that the observed nucleotide patterns at the sites in the alignment become increasingly likely. To give an example of the forces at play in this iteration: as you increase the error rates you will decrease the likelihood of observing 'clean' patterns (e.g. patterns of only As and Cs at site types A/C) and increase the likelihood of observing 'noisy' patterns (e.g. patterns of other than only As, and C at site types A/C). If, on the other hand, you decrease the error rates, you will increase the likelihood of observing 'clean' patterns and decrease the likelihood of observing 'noisy' patterns. The EM procedure ensures that the balance between these two is optimized relative to the data observed (assuming, of course, that the ploidy assumption is valid).

Updating equations for the prior site type probabilities

We first derive the updating equations for the prior site type probabilities $f_s, s \in S$. The probability that the site is of type t given that we observe the nucleotides n_1, \dots, n_k in the reads at the site is:

$$\begin{aligned} P(t|n_1, \dots, n_k) &= \frac{P(t, n_1, \dots, n_k)}{\sum_{s \in S} P(s, n_1, \dots, n_k)} \\ &= \frac{P(t)P(n_1, \dots, n_k|t)}{\sum_{s \in S} P(s)P(n_1, \dots, n_k|s)} \end{aligned} \quad (31.4)$$

Now, for $P(t)$ we use our current value for f_t , and if we further insert the expression for $P(n_1, \dots, n_k|t)$ (31.2) we get:

$$P(t|n_1, \dots, n_k) = \frac{f_t \prod_{i=1}^k \sum_{N \in \{A, C, G, T, -\}} P_t(N) \times e_q(N \rightarrow n_i)}{\sum_{s \in S} f_s \prod_{i=1}^k \sum_{N \in \{A, C, G, T, -\}} P_s(N) \times e_q(N \rightarrow n_i)} \quad (31.5)$$

We get the updating equation for the prior site type probabilities, $f_t, t \in S$, from equation 31.5: Let h index the sites in the alignment ($h = 1, \dots, H$). Given the current values for the set of site frequencies, $f_t, t \in S$, and the current values for the set of error probabilities, we obtain updated

values for the site frequencies, $f_t^*, t \in S$, by summing the site type probabilities given the data (as given by equation 31.5) across all sites in the alignment:

$$f_t^* = \frac{\sum_{h=1}^H \frac{f_t \prod_{i=1}^k \sum_{N \in \{A,C,G,T,-\}} P_t(N) \times e_q(N \rightarrow n_i^h)}{\sum_{s \in S} f_s \prod_{i=1}^k \sum_{N \in \{A,C,G,T,-\}} P_s(N) \times e_q(N \rightarrow n_i^h)}}{H} \quad (31.6)$$

Updating equations for the error rates

For the updating equations for the error probabilities, we consider a read, i , at a given site, h . The joint probability of the true nucleotide in the read, r_i^h , at the site being N and the data $n_1^h, \dots, n_{k_h}^h$ is:

$$\begin{aligned} P(r_i^h = N, n_1^h, \dots, n_{k_h}^h) &= \sum_{s \in S} f_s P(r_i^h = N, n_1^h, \dots, n_{k_h}^h | s) \\ &= \sum_{s \in S} f_s \prod_i P(r_i^h = N, n_i^h | s) \\ &= \sum_{s \in S} f_s P(r_i^h = N, n_i^h | s) \prod_{j \neq i} P(n_j^h | s) \\ &= \sum_{s \in S} f_s (P_s(N) \times e_{q_{ih}}(N \rightarrow n_i^h)) \prod_{j \neq i} \sum_{N' \in \{A,C,G,T,-\}} P_s(N') \times e_{q_{jh}}(N' \rightarrow n_j^h) \end{aligned} \quad (31.7)$$

Using Bayes formula again, as we did above in 31.4, we get:

$$\begin{aligned} P(r_i^h = N | n_1^h, \dots, n_{k_h}^h) &= \frac{P(r_i^h = N, n_1^h, \dots, n_{k_h}^h)}{P(n_1^h, \dots, n_{k_h}^h)} \\ &= \frac{P(r_i^h = N, n_1^h, \dots, n_{k_h}^h)}{\sum_{N' \in \{A,C,G,T,-\}} P(r_i^h = N', n_1^h, \dots, n_{k_h}^h)} \end{aligned} \quad (31.8)$$

and inserting the expression from equation 31.7:

$$\begin{aligned} P(r_i^h = N | n_1^h, \dots, n_{k_h}^h) &= \frac{\sum_{s \in S} f(s) (P_s(N) \times e_{q_{ih}}(N \rightarrow n_i^h) \prod_{j \neq i} \sum_{N' \in \{A,C,G,T,-\}} P_s(N') \times e_{q_{jh}}(N' \rightarrow n_j^h))}{\sum_{N' \in \{A,C,G,T,-\}} (\sum_{s \in S} f(s) (P_s(N') \times e_{q_{ih}}(N' \rightarrow n_i^h) \prod_{j \neq i} \sum_{N' \in \{A,C,G,T,-\}} P_s(N') \times e_{q_{jh}}(N' \rightarrow n_j^h)))} \\ &= \frac{\sum_{s \in S} f(s) (P_s(N) e_{q_{ih}}(N \rightarrow n_i^h) \prod_{j \neq i} \sum_{N' \in \{A,C,G,T,-\}} P_s(N') \times e_{q_{jh}}(N' \rightarrow n_j^h))}{(\sum_{s \in S} f(s) \prod_j \sum_{N' \in \{A,C,G,T,-\}} P_s(N') \times e_{q_{jh}}(N' \rightarrow n_j^h))} \end{aligned} \quad (31.9)$$

The equation 31.9 gives us the probabilities for a given read, i , and site, h , given the data $n_1^h, \dots, n_{k_h}^h$, that the true nucleotide is N , $N \in \{A, C, G, T, -\}$, given our current values of the error

rates and site probabilities. Since we know the sequenced nucleotide in each read at each site, we can get new updated values for the error rate of producing an M nucleotide when the true nucleotide is N , $e_q^*(N \rightarrow M)$, for $N, M \in \{A, C, G, T, -\}$ by summing the probabilities of the true nucleotide being N for all reads across all sites for which the sequenced nucleotide is M , and dividing by the sum of all probabilities of the true nucleotide being a N across all reads and all sites:

$$e_q^*(N \rightarrow M) = \frac{\sum_h \sum_{i=1, \dots, k_h: n_i^h = M} P(r_i^k = N | n_1^h, \dots, n_{k_h}^h)}{\sum_h \sum_{i=1, \dots, k_h} P(r_i^k = N | n_1^h, \dots, n_{k_h}^h)}$$

31.7.3 The Low Frequency Variant Detection tool: Models and methods

This section describes the model, method and estimation procedure behind the Low Frequency Variant Detection tool. The Low Frequency Variant Detection tool is designed to detect variants in a sample for which the ploidy is unknown. The Low Frequency Variant Detection tool has a particularly high *sensitivity* for detecting variants that are present at any, and in particularly at low, allele frequencies.

The model for the Low Frequency Variant Detection tool

The purpose of the Low Frequency Variant Detection tool is to call variants in samples with unknown ploidy (e.g., samples of limited tumor purity) using read mapping data. It will detect germline as well as somatic variants, and may also be used on samples from other high-ploidy organisms, or pooled samples. Like the Fixed Ploidy Variant Detection tool it detects Single Nucleotide Variants (SNVs), MNVs (Multiple Nucleotide Variants), insertions, deletions as well as replacements (combinations of neighboring insertions, deletions and SNVs for which the positions are ambiguous).

The Low Frequency Variant Detection algorithm relies on multinomial models to determine the presence of different alleles at a given site and an error model to account for sequencing error. The error model employed here is the same as is used in the Fixed Ploidy Variant Detection tool, described in section 31.7.2.

The multinomial models are of the kind "there are q different nucleotide alleles present at the site with frequencies f_i , $i = 1, \dots, q$, $\sum_{i=1}^q f_i = 1$ ", where the number of alleles, q , differ. The models that are evaluated at each site are given in Table 31.1.

In words, model M_x can be described as: "There is really only the X nucleotide allele present at the site, *all other nucleotides are due to errors*" and model $M_{x,y,z}$ as: "There are really only the nucleotide alleles X , Y and Z present at the site, *all other nucleotides are due to errors*". The hypotheses where a single nucleotide is present in the sample have no free parameters (there is just one frequency parameter and it must be 1). A hypothesis stating that a site is a mixture of two different nucleotides, e.g. [A/G] has one free parameter since there are frequencies for two nucleotides but they have to sum to one.

Parameter estimation relies on the Maximum Likelihood principle, and, as the Fixed Ploidy Variant Detection tool, the EM algorithm is used for estimating the parameters of the model. Given an initial set of parameter values for the error rates, the different multinomial models are evaluated at each site by finding the maximum likelihood estimates of the frequency parameters for each model. The model that offers the best explanation of the data (while taking care to adjust for

Model	Alleles present at the site	Description	Free parameters*
M_x	x	the only allele present at the site is x .	none
$M_{x,y}$	x and y	x is present at frequency $1 - f$, y at frequency f	f
$M_{x,y,z}$	x , y and z	x is present at frequency $1 - (f_1 + f_2)$, y at frequency f_1 and z at frequency f_2	f_1 and f_2
$M_{x,y,z,w}$	x , y , z and w	x is present at frequency $1 - (f_1 + f_2 + f_3)$, y at frequency f_1 , z at frequency f_2 and w at frequency f_3	f_1 , f_2 and f_3
$M_{x,y,z,w,v}$	x , y , z , w and v	x is present at frequency $1 - (f_1 + f_2 + f_3)$, y at frequency f_1 , z at frequency f_2 , w at frequency f_3 and v at frequency f_4	f_1 , f_2 , f_3 and f_4

Table 31.1: The multinomial models evaluated at each site. X, Y, Z, W and V each take on one of the values A, C, G, T , or $-$ ($X \neq Y \neq Z \neq W \neq V$). Free parameters*: the parameters that are free in each of the multinomial models of the Low Frequency Variant Detection tool.

the numbers of parameters in the multinomial model, using a criterion adopted from the Akaike Information criterion) is chosen as the current guess of the true allelic situation at that site, and given that, the error rates are re-estimated. Given the new error estimates, the maximum log likelihoods for all possible multinomial models are again evaluated and updated frequencies are produced. This procedure is performed a total of four times. After the final round of estimation the multinomial model that offers the best explanation of the data is chosen as the winning model, and variants are called according to that model.

Below we describe in detail how we choose among competing models and derive the updating equations for the EM estimation of the frequency and error rate parameters.

Updating the choice of favored multinomial model for each site

Given a set of error rates, and, for each site, a set of Maximum Likelihood estimates of the nucleotide allele frequencies for each multinomial model, we can calculate the Maximum loglikelihood values for each of the models at each site. Since the hypotheses with fewer free parameters are special cases of hypotheses with more free parameters, the hypotheses with the most free parameters will necessarily have the highest likelihoods. We wish to only favor a model with more parameters over one with fewer, if it offers a significantly better explanation of the data at the site. In the simple case where we have nested models (e.g. a hypothesis, H_0 , with no free parameters and an alternative hypothesis, H_1 , which has one free parameter and contains H_0 as a special case) it is a well known result that twice the loglikelihood ratio is approximately χ^2 distributed with a parameter that is equal to the difference between the number of free parameters in the hypotheses, n :

$$2 \log \frac{L(H_1)}{L(H_0)} \sim \chi^2(n).$$

If we write $c_n(p)$ for the inverse cumulative probability density function for a $\chi^2(n)$ distribution evaluated at $1 - p$, we get a cutoff value for when we prefer H_1 over H_0 at the significance level given by p .

We wish to compare all models together and some of these will not be nested. We therefore generalize this approach to apply to any set of multinomial model hypothesis H_m , $m = 1, \dots, M$. For each model we calculate the value:

$$v_m = 2 \log L(H_m) - c_{df_m}(p) \quad (31.10)$$

where df_m is the number of free parameters in hypothesis H_m . We now prefer the hypothesis with the highest value of v (note that when comparing a hypothesis with zero free parameters to another hypothesis, we get exactly the same results as with the log likelihood ratio approach). If this hypothesis is that only the reference allele is present at the site, no variants are called. For other models, the variants corresponding to the alleles hypothesized by the models are called.

The approach applied is an extension of the Akaike approach: In the Akaike approach you subtract twice the number of parameters, whereas in the approach that we apply, we subtract $c_{df_x}(p)$, whereby additional parameters are punished relative to the significance level, p , rather than in a fashion that is linear in the number of parameters. This means that the more parameters that are present, the more reluctant we will be to add even more.

Updating equations for the multinomial model frequency parameters

Consider a site, h , and let n_i^h be the nucleotide observed in read i at this site, $i = 1, \dots, k$. For each of the multinomial models that may explain the data at the site we have a number of frequency parameters. For simplicity, we consider the model which states that there are two alleles present at the site, the reference allele, y , and another allele x , and let f be the frequency parameter for the non-reference allele (hence the frequency of the reference allele, f_y , is $1 - f$). Models with more alleles are treated in a similar manner.

We want to estimate the parameter for the frequency of the x allele at the site h , f , by the fraction of true nucleotides that are x at this site, given the observed data:

$$f^* = \frac{\sum_{i=1}^k P(r_i^h = x | n_i^h)}{k}. \quad (31.11)$$

To calculate this we use Bayes Theorem on the numerator:

$$P(r_i^h = x | n_i^h) = \frac{P(r_i^h = x, n_i^h)}{P(n_i^h)} \quad (31.12)$$

$$= \frac{P(x) \times e(x \rightarrow n_i^h)}{P(x) \times e(x \rightarrow n_i^h) + P(y) \times e(y \rightarrow n_i^h)} \quad (31.13)$$

$$= \frac{f \times e(x \rightarrow n_i^h)}{f \times e(x \rightarrow n_i^h) + (1 - f) \times e(y \rightarrow n_i^h)} \quad (31.14)$$

Inserting our current values for the frequency parameter f under the model, and the error rates $e(x \rightarrow n_i^h)$ and $e(y \rightarrow n_i^h)$, in 31.12, and further inserting the obtained values in 31.11 gives us updated values for the frequency parameter f .

Updating equations for the error rates

Consider a site h and a read i . The joint probability of the true nucleotide in the read, r_i^h , at the site being N and the observed nucleotide at the site n_i^h is:

$$P(r_i^h = N, n_i^h) = P_h(N) e_{q_i^h}(N \rightarrow n_i^h). \quad (31.15)$$

Using Bayes Theorem, the probability of the true nucleotide in the read, r_i^h , at the site being N , given that we observe n_i^h is:

$$P(r_i^h = N | n_i^h) = \frac{P(r_i^h = N, n_i^h)}{\sum_{N' \in A, C, G, T, -} P(r_i^h = N', n_i^h)}. \quad (31.16)$$

Inserting 31.15 in 31.16 we get:

$$P(r_i^h = N | n_i^h) = \frac{P_h(N) e_{q_i^h}(N \rightarrow n_i^h)}{\sum_{N' \in A, C, G, T, -} P_h(N') e_{q_i^h}(N' \rightarrow n_i^h)}. \quad (31.17)$$

The equation 31.17 gives us the probabilities for a given read, i , and site, h , given the observed nucleotide n_i^h , that the true nucleotide is N , $N \in \{A, C, G, T, -\}$, given our current values for the frequency f (inserted for $P_h(N)$) and error rates. Since we know the sequenced nucleotide in each read at each site, we can get new updated values for the error rate of producing an M nucleotide when the true nucleotide is N , $e_q^*(N \rightarrow M)$, for $N, M \in \{A, C, G, T, -\}$ by summing the probabilities of the true nucleotide being N for all reads across all sites for which the sequenced nucleotide is M , and dividing by the sum of all probabilities of the true nucleotide being a N across all reads and all sites:

$$e_q^*(N \rightarrow M) = \frac{\sum_h \sum_{i=1, \dots, k_h: n_i^h = M} P(r_i^h = N | n_i^h)}{\sum_h \sum_{i=1, \dots, k_h} P(r_i^h = N | n_i^h)}.$$

Probability of the variant

A probability value is reported for each variant call in the variant track table column with the header "Probability". For the Fixed Ploidy Variant Detection tool this value is the posterior probability of the site being different from the homozygous reference site. This is possible to calculate because the model behind the Fixed Ploidy Variant Detection tool is Bayesian. As the model for the Low Frequency Variant Detection tool is not a Bayesian model, such a value strictly speaking does not exist. However, as a proxy for such a value we report the support for the winning model m relative to the total support of all models:

$$P_m = \frac{e^{v_m}}{\sum_{m'} e^{v_{m'}}$$

31.8 Copy Number Variant Detection

The Copy Number Variant Detection (Targeted) tool is designed to detect copy number variations (CNVs) from targeted resequencing experiments.

The tool takes read mappings and target regions as input, and produces amplification and deletion annotations. The annotations are generated by a 'depth-of-coverage' method, where the target-level coverages of the case and the controls are compared in a statistical framework using a model based on 'selected' targets. Note that to be 'selected', a target has to have a coverage higher than the specified coverage cutoff AND must be found on a chromosome that was not identified as a coverage outlier in the chromosomal analysis step. If fewer than 50 'selected' targets are found suitable for setting up the statistical models, the CNV tool will terminate prematurely.

The algorithm implemented in the Copy Number Variant Detection (Targeted) tool is inspired by the following papers:

- **Li et al.**, *CONTRA: copy number analysis for targeted resequencing*, Bioinformatics. 2012, 28(10):1307-1313 [[Li et al., 2012](#)].
- **Niu and Zhang**, *The screening and ranking algorithm to detect DNA copy number variations*, Ann Appl Stat. 2012, 6(3): 1306-1326 [[Niu and Zhang, 2012](#)].

For more information, you can also read our whitepaper: https://digitalinsights.qiagen.com/files/whitepapers/Biomedical_Genomics_Workbench_CNV_White_Paper.pdf.

The Copy Number Variant Detection (Targeted) tool identifies CNVs regions where the normalized coverage is statistically significantly different from the controls.

The algorithm carries out the analysis in several steps.

1. Base-level coverages are analyzed for all samples, and a robust coverage baseline is generated using the control samples.
2. Chromosome-level coverage analysis is carried out on the case sample, and any chromosomes with unexpectedly high or low coverages are identified.
3. Sample coverages are normalized, and a global, target-level statistical model is set up for the variation in fold-change as a function of coverage in the baseline.
4. Each chromosome is segmented into regions of similar fold-changes.
5. The expected fold-change variation in region is determined using the statistical model for target-level coverages. Region-level CNVs are identified as the regions with fold-changes significantly different from 1.0.
6. If chosen in the parameter steps, gene-level CNV calls are also produced.

31.8.1 The Copy Number Variant Detection (Targeted) tool

To run the Copy Number Variant Detection (Targeted) tool, go to:

Tools | Resequencing Analysis () | **Variant Detection** () | **Copy Number Variant Detection (Targeted)** ()

Select the case read mapping and click **Next**.

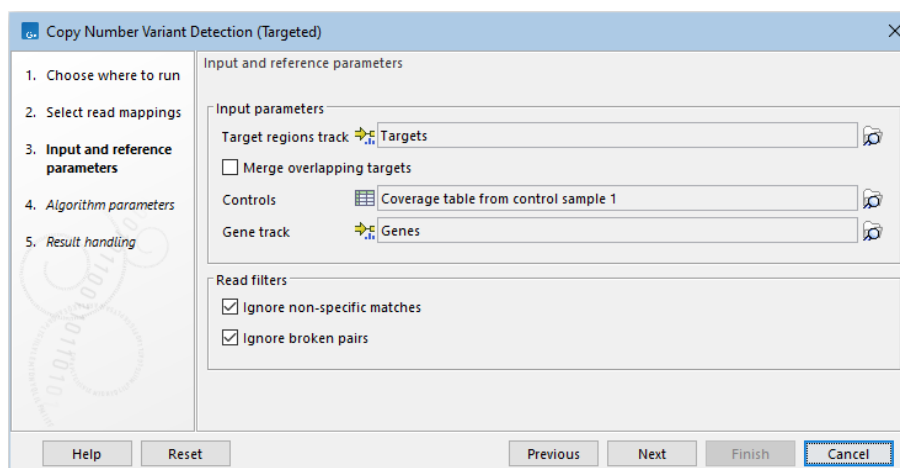


Figure 31.22: The first step of the CNV detection tool.

You are now presented with choices regarding the data to use in the CNV prediction method, as shown in figure 31.22.

- **Target regions track** An annotation track containing the regions targeted in the experiment must be chosen. This track must not contain overlapping regions, or regions made up of several intervals, because the algorithm is designed to operate on simple genomic regions.
- **Merge overlapping targets** When enabled, overlapping target regions will be merged into one larger target region by expanding the first region to include all the bases of the overlapping targets, regardless of their strandedness. CNV calls are made on this larger region of merged amplicons, considered to be of undefined strand if it originated from both + and - stranded targets.
- **Controls** You must specify at least one read mapping or coverage table. The controls will be used to create a baseline by the algorithm. Coverage tables can be generated using the **QC for Targeted Sequencing** tool, see section 29.1. When using coverage tables, it is important to use the same target region and settings for handling non-specific matches and broken pairs in this tool and in **QC for Targeted Sequencing**. For the best results, the controls should be matched with respect to the most important experimental parameters, such as gender and technology. If using non-matched controls, the CNVs reported by the algorithm may be less accurate.
- **Gene track** Optional: If you wish, you can provide a gene track, which will be used to produce gene-level output as well as CNV-level output.
- **Ignore non-specific matches** If checked, the algorithm will ignore any non-specifically mapped reads when counting the coverage in the targeted positions. Note: If you are interested in predicting CNVs in repetitive regions, this box should be unchecked.
- **Ignore broken pairs** If checked, the algorithm will ignore any broken paired reads when counting the coverage in the targeted positions.

Click **Next** to set the parameters related to the target-level and region-level CNV detection, as shown in as shown in figure 31.23.

The screenshot shows a software window titled "Copy Number Variant Detection (Targeted)". On the left is a sidebar with five steps: 1. Choose where to run, 2. Select read mappings, 3. Input and reference parameters, 4. Algorithm parameters (selected), and 5. Result handling. The main area is titled "Algorithm parameters" and contains two sections. The "Statistics" section has four input fields: "Threshold for significance" (0.05), "Minimum fold change for amplification" (1.4), "Minimum fold change for deletion" (1.4), and "Low coverage cutoff" (30). The "Graining level" section has three radio buttons: "Coarse" (selected), "Intermediate", and "Fine", plus a checkbox for "Enhance single-target sensitivity" which is unchecked. At the bottom are buttons for "Help", "Reset", "Previous", "Next" (highlighted with a blue border), "Finish", and "Cancel".

Figure 31.23: The second step of the CNV detection tool

- **Threshold for significance** P-values lower than the threshold for significance will be considered "significant". The higher you set this value, the more CNVs will be predicted.
- **Minimum fold change for amplification** and **Minimum fold change for deletion** You must specify the minimum fold changes for a CNV call for amplification and deletion. If the absolute value of the fold change of a CNV is less than the value specified in this parameter, then the CNV will be filtered from the results, even if it is otherwise statistically significant. For example, if a minimum fold-change of 1.5 is chosen for amplification, then the adjusted coverage of the CNV in the case sample must be 1.5 times higher than the coverage in the baseline for it to pass the filtering step. Similarly, if a minimum fold-change of 1.5 is chosen for deletion, then the adjusted coverage of the CNV in the case sample must be 1.5 times lower than the coverage in the baseline.

If you do not want to filter on the fold-change, enter 0.0 in these fields. Also, if your sample purity is less than 100%, it is necessary to take that into account when adjusting the fold-change cutoff. This is described in more detail in section 31.8.1. Note: This value is used to filter the Region-level CNV track. The Target-level CNV track will always include full information for all targets.

- **Low coverage cutoff** If the average coverage of a target is below this value in the control read mappings, it will be considered "low coverage" and it will not be used to set up the statistical models, and p-values will not be calculated for it in the target-level CNV prediction.

Note: Targets with low control coverage are included when targets are binned to identify region level copy numbers. Hence the number of targets supporting a region-level CNV can be very low if some targets have low control coverage and having many targets with low control coverage should be avoided. This can be achieved by setting an appropriate low coverage cutoff or by removing targets from the target regions file that are known to have low coverage.

- **Graining level** The graining level is used for the region-level CNV prediction. Coarser graining levels produce longer CNV calls and less noise, and the algorithm will run faster. However, smaller CNVs consisting of only a few targets may be missed at a coarser graining level.
 - Coarse: prefers CNVs consisting of many targets. The algorithm is most sensitive to CNVs spanning over 10 targets. This is the recommended setting if you expect

large-scale deletions or insertions, and want a minimal false positive rate.

- Intermediate: prefers CNVs consisting of an intermediate number of targets. The algorithm is most sensitive to CNVs spanning 5 or more targets. This is the recommended setting if you expect CNVs of intermediate size.
- Fine: prefers CNVs consisting of fewer targets. The algorithm is most sensitive to CNVs spanning 3 or more targets. This is the recommended setting if you want to detect CNVs that span just a few targets, but the false positive rate may be increased.

Note: The CNV sizes listed above are meant as general guidelines, and are not to be interpreted as hard rules. Finer graining levels will produce larger CNVs when the signals for this are sufficiently clear in the data. Similarly, the coarser graining levels will also be able to predict shorter CNVs under some circumstances, although with a lower sensitivity.

- **Enhance single-target sensitivity** All of the graining levels assume that a CNV spans more than one target. If you are also interested in very small CNVs that affect down to a single target in your data, check the 'Enhance single-target sensitivity' box. This will increase the sensitivity of detection of very small CNVs, and has the greatest effect in the case of the coarser graining levels. Note however that these small CNV calls are much more likely to be false positives. If this box is unchecked, only larger CNVs supported by several targets will be reported, and the false positive rate will be lower.

Clicking **Next**, you are presented with options about the results (see figure 31.24). In this step, you can choose to create an algorithm report by checking the **Create algorithm report** box. Furthermore, you can choose to output results for every target in your input, by checking the **Create target-level CNV track** box.

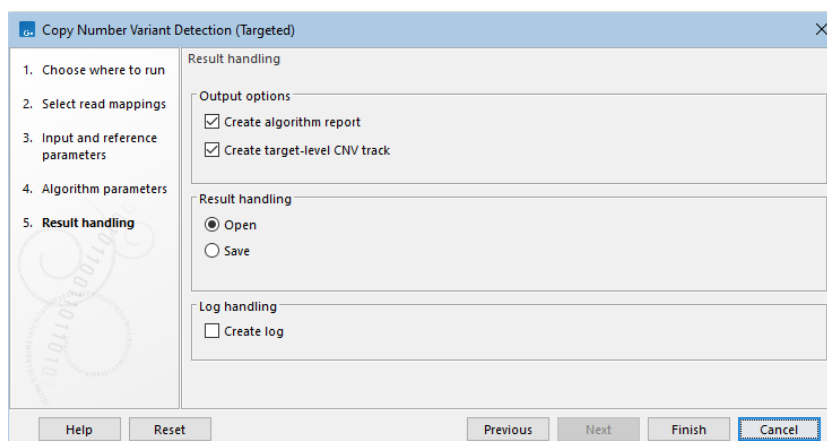


Figure 31.24: Specifying whether an algorithm report and a target-level CNV track should be created.

When finished with the settings, click **Next** to start the algorithm.

Copy number and fold change

When configuring the minimum fold change thresholds for calling CNVs, it can be useful to understand the difference between copy number and fold change and the relationship between tumor fold change, sample fold change and sample purity.

The copy number (CN) gives the number of copies of a gene. For a normal diploid sample the copy number, or ploidy, of a gene is 2.

The fold change is a measure of how much the copy number of a case sample differs from that of a normal sample. When the copy number for both the case sample and the normal sample is 2, this corresponds to a fold change of 1 (or -1).

The sample fold change can be calculated from the normal copy number and sample copy number. The formula differs for amplifications and deletions:

$$\text{Fold change, amplifications (CN(sample) > CN(normal))} = \frac{\text{CN(sample)}}{\text{CN(normal)}}$$

$$\text{Fold change, deletions (CN(sample) < CN(normal))} = -\frac{\text{CN(normal)}}{\text{CN(sample)}}$$

Fold change values for amplifications and deletions are asymmetric in that a 50% increase in copy number from 2 to 3 (heterozygote amplification) converts to a fold change of 1.5, whereas a 50% decrease in copy number from 2 to 1 (heterozygous deletion), gives a fold change of -2.0. The difference is even more pronounced if we consider what could be interpreted as a homozygote duplication (copy number 4) and a homozygote deletion (copy number 0). Here, the calculated fold changes land at 2 and $-\infty$, respectively.

The fact that the same percent-wise change in coverage (copy number) leads to a higher fold change for deletions than for amplifications means that given the same amplification and deletion fold change cutoff there is a higher risk of calling false positive deletions than amplifications - it takes less coverage fluctuation to pass the fold change cutoff for deletions.

	Copy number	Fold change
Amplifications		
	2	1
	3	1.5
	4	2
	6	3
	8	4
Deletions		
	2	-1
	1	-2
	0.5	-4
	0.2	-10
	0.1	-20
	0	$-\infty$

Table 31.2: The relationship between copy number and fold change for amplifications and deletions.

How to set the fold-change cutoff when the sample purity is not 100%

Given a sample purity of $X\%$, and a desired detection level (absolute value of fold-change in 100% pure sample) of T , the following formula gives the required fold-change cutoff for an amplification:

$$\text{cutoff} = \frac{X\%}{100\%} \times T + \left(1 - \frac{X\%}{100\%}\right).$$

For example, if the sample purity is 40%, and you want to detect 6-fold amplifications (e.g. 12 copies instead of 2), then the cutoff should be:

$$\text{cutoff} = \frac{40\%}{100\%} \times 6 + \left(1 - \frac{40\%}{100\%}\right) = 3.0.$$

The following formula gives the required fold-change cutoff for a deletion:

$$\text{cutoff} = \frac{1}{\frac{X\%}{100\%} \times \frac{1}{T} + \left(1 - \frac{X\%}{100\%}\right)}.$$

For example, if the sample purity is 40%, and you want to detect a 2-fold deletions (e.g. 1 copy instead of 2), then the cutoff should be:

$$\text{cutoff} = \frac{1}{\frac{40\%}{100\%} \times \frac{1}{2} + \left(1 - \frac{40\%}{100\%}\right)} = 1.25.$$

Figure 31.25 and Figure 31.26 shows the required fold-change cutoffs in order to detect a particular degree of amplification or deletion respectively at different sample purities.

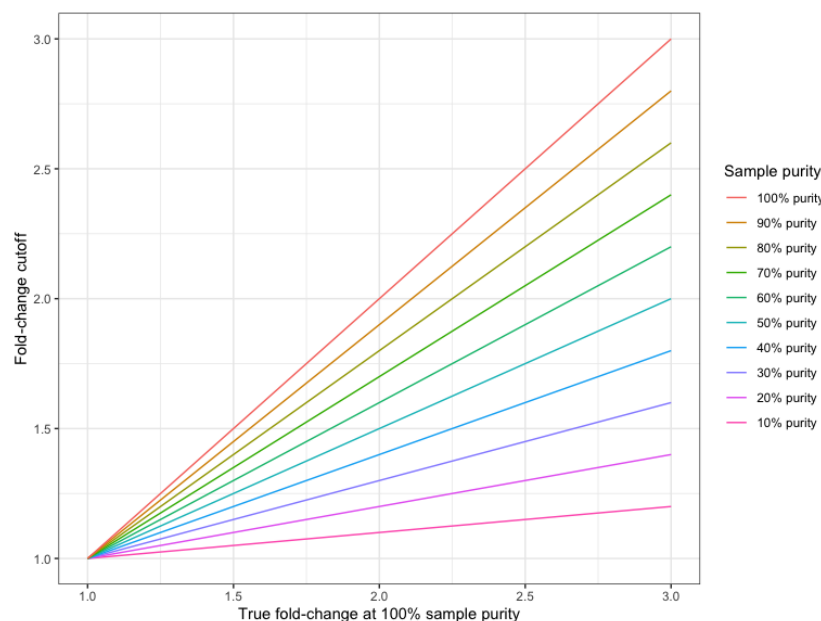


Figure 31.25: The required fold-change cutoff to detect amplifications of different magnitudes as a function of sample purity.

The Copy Number Variant Detection (Targeted) tool calls CNVs that are both global outliers on the target-level, and locally consistent on the region-level. The tool produces several outputs, which are described below.

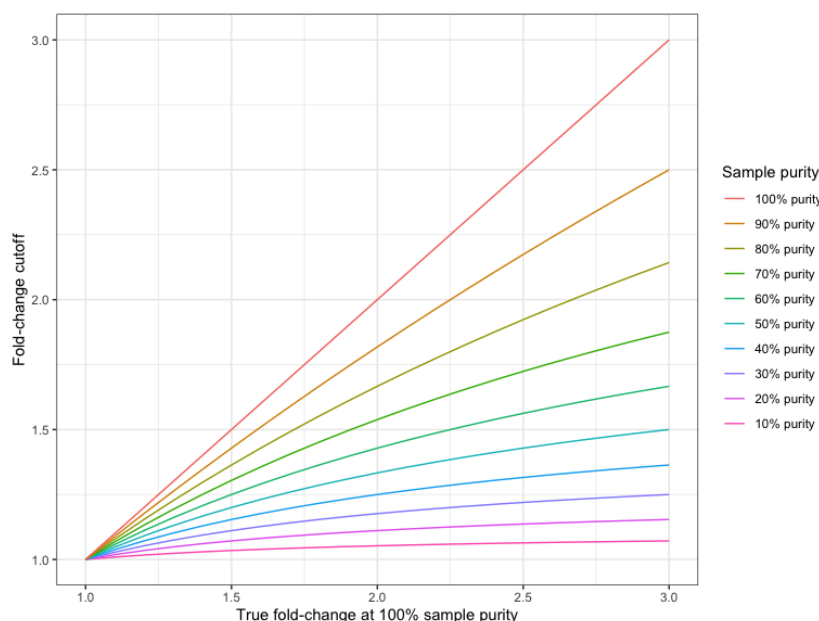


Figure 31.26: The required fold-change cutoff to detect deletions of different magnitudes as a function of sample purity.

31.8.2 Region-level CNV track (Region CNVs)

The algorithm will produce a region-level annotation track, which contains the CNV regions detected by the algorithm. Every annotation in this track joins one or more targets from the input target track, to produce contiguous CNVs. Each CNV in the region-level tracks is characterized in terms of the following properties:

Minimum CNV length: The minimum CNV length is the length of the region-level CNV annotation. This number should be interpreted as the lowest bound for the size of the CNV. The "true" CNV can extend into the adjacent genomic regions that have not been targeted.

P-value: The p-value corresponds to the probability that an observation identical to the CNV, or even more of an outlier, would occur by chance under the null hypothesis. The null hypothesis is that of no CNVs in the data. The p-value for a CNV region is calculated by combining the p-values of its constituent targets (discarding any low-coverage targets) using Fisher's method.

Fold-change (adjusted): The fold-change of the *adjusted* case coverage compared to the baseline. Negative fold-changes indicate deletions, and positive fold-changes indicate amplifications. A fold-change of 1.0 (or -1.0) represents identical coverages. The fold-changes are adjusted for statistical differences between targets with different sequencing depths. The fold-change for a CNV region is the median of the adjusted fold-changes of its constituent targets (discarding any low-coverage targets). Note: if your sample purity is less than 100%, you need to take that into account when interpreting the fold-change values. This is described in more detail in section 31.8.5.

Consequence: The consequence classifies statistically significant CNVs as "Gain" or "Loss".

Number of targets: The total number of targets forming the (minimal) CNV region.

Targets: A list of the names of the targets forming the (minimal) CNV region. Note however that the list is truncated to 100 characters. If you want to see all the targets that constitute the CNV region, you can use the target-level output (section 31.8.3).

Comments: The comments can include useful information for interpreting individual CNV calls. The possible comments are:

1. Small region: If a region only consists of 1 target, it is classified as a 'small region'. The p-value of this region is therefore based on evidence from just one target, and may be less accurate than p-values for larger regions.
2. Disproportionate chromosome coverage: If a region is found on a chromosome that was determined to have disproportionate coverage, this will be noted in the comments. This means that the targets constituting this region were not used to set up the statistical models. Furthermore, the size and fold-change value of this CNV region may explain why the chromosome was detected to have disproportionate coverage.
3. Low coverage: If all targets inside a region had low-coverage, then the region will be classified as a 'low-coverage' region, and will be given a p-value of 1.0. You will only see these regions in the results if you set the significance cutoff to 1.0.

These properties can be found in separate columns when viewing the tracks in table view.

Note: The region-level calls do not guarantee that a single, larger CNV will always be called in just one CNV region. This is because adjacent region-level CNV calls are not joined into a single region if their average fold-changes are sufficiently different. For example, if a 2-fold gain is detected in a region and a 3-fold gain is detected in an immediately adjacent region of equal size, then these may appear in the results as two separate CNVs, or one single CNV with a 2.5-fold gain, depending on your chosen graining level, and the fold-changes observed in the rest of the data.

31.8.3 Target-level CNV track (Target CNVs)

The algorithm will produce a target-level CNV track, if you have chosen to create one when running the algorithm. The target-level CNV track is an annotation track, containing one annotation for every target in the input data. Inspection of the target-level CNV track can provide additional information about identified region-level CNVs as well as regions where no CNVs were detected. The sample is not considered covered enough for statistical purposes if you have fewer than 49 statistically useful targets.

Each target is annotated with the following information:

Target number: Targets are numbered in the order in which they occur in the genome. This information is used by the results report (see section 31.8.6).

Case coverage: The normalized coverage of the target in the case sample.

Baseline coverage: The normalized coverage of the target in the baseline.

Length: The length of the target region.

P-value: The p-value corresponds to the probability that an observation identical to the CNV, or even more of an outlier, would occur by chance under the null hypothesis. The null

hypothesis is that of no CNVs in the data. The p-value in the target-level output reflects the global evidence for a CNV at that particular target. The target-level p-values are combined to produce the region-level p-values in the region-level CNV output.

FDR-corrected p-value: The FDR-corrected p-values correct for false positives arising from carrying out a very high number of statistical tests. The FDR-corrected p-value will, therefore, always be larger than the uncorrected p-value.

Fold-change (raw): The fold-change of the normalized case coverage compared to the normalized baseline coverage. The normalization corrects for the effects of different library sizes between the different samples. Negative fold-changes indicate deletions, and positive fold-changes indicate amplifications. A fold-change of 1.0 represents identical coverages.

Fold-change (adjusted): As observed by Li et al (2012, [Li et al., 2012]), the fold-changes (raw) depend on the coverage. Therefore, the fold-changes have to be adjusted for statistical differences between targets with different sequencing depths, before the statistical tests are carried out. The results of this adjustment are found in the "Fold-change (adjusted)" column. Note that sometimes, this will mean that a change that appears to be an amplification in the "raw" fold-change column may appear to be a deletion in the "adjusted" fold-change column, or vice versa. This is simply because for a given coverage level, the raw fold-changes were skewed towards amplifications (or deletions), and this effect was corrected in the adjustment. Note: if your sample purity is less than 100%, you need to take that into account when interpreting the fold-change values. This is described in more detail in section 31.8.5.

Standard deviation: The estimated standard deviation of log2 fold-change used to compute the p-value.

Statistically useful: A target that has a coverage higher than the specified coverage cutoff, AND is found on a chromosome that was not identified as a coverage outlier in the chromosomal analysis step.

Region (joined targets): The region to which this target was classified to belong. The region may or may not have been predicted to be a CNV.

Regional fold-change: The adjusted fold-change of the region to which this target belongs. This fold-change value is computed from all targets constituting the region.

Regional p-value: The p-value of the region to which this target belongs. This is the p-value calculated from combining the p-values of the individual targets inside the region.

Regional consequence: If the target is included in a CNV region, this column will show "Gain" or "Loss", depending on the direction of change detected for the region. Note, however, that the change detected for the region may be inconsistent with the fold-change for a single target in the region. The reason for this is typically statistical noise at the single target. Regional consequence column is only filled in the target-level output when the region is both significant (determined by the p-value) AND has a "Strong" effect size (determined by the fold-change).

Regional effect size: The effect size of a target-level CNV reflects the magnitude of the observed fold-change of the CNV region in which the target was found. The effect size of a CNV is classified into the following categories: "Strong" or "Weak". The effect size is "Strong"

if the fold-change exceeds the fold-change cutoff specified in the parameter steps. A "Weak" CNV calls will be filtered from the region-level output. Regional effect size column is only filled in the target-level output when the region is both significant (determined by the p-value) AND has a "Strong" effect size (determined by the fold-change).

Comments: The comments can include useful information for interpreting the CNV calls. Possible comments in the target-level output are:

1. Low coverage target: If the target had a coverage under the specified coverage cutoff, it will be classified as low-coverage. Low-coverage targets were not used in calculating the statistical models, and will not have p-values.
2. Disproportionate chromosome coverage: If the target occurred on a chromosome that was detected to have disproportionate coverage. In this case, the target was not used to set up the statistical models.
3. Atypical fold-change in region: If there is a discrepancy between the direction of fold-change detected for the target and the direction of fold-change detected for the region, then the fold-change of the target is "atypical" compared to the region. This is usually due to statistical noise, and the regional fold-change is likely to be more accurate in the interpretation, especially for large regions.

31.8.4 Gene-level annotation track (Gene CNVs)

If you have specified a gene track in the input parameters, you will get a gene-level CNV track as well. The gene-level CNV track is an annotation track, which is obtained by intersecting the region-level CNV track with the gene track in the input (ignoring any genes that do not overlap with the targets). Note that a single CNV may be reported several times in different genes, and a single gene may also be reported several times, if it is affected by more than one CNV. In addition to the annotations on the gene track supplied in the input parameters, the gene-level CNV track contains the following annotation columns:

Region length: The length of the actual annotation. That is, the length of the CNV region intersected with the gene.

CNV region: The *entire* CNV region affecting this gene (and possibly other genes).

CNV region length: The length of the *entire* CNV region affecting this gene (and possibly other genes).

Consequence: The consequence classifies statistically significant CNVs as "Gain" or "Loss".

Fold-change (adjusted): The adjusted fold-change of the *entire* CNV region affecting this gene (and possibly other genes).

P-value: The p-value of the *entire* CNV region affecting this gene (and possibly other genes).

Number of targets: The total number of targets forming the *entire* CNV region affecting this gene (and possibly other genes).

Comments: If the CNV region affecting this gene had any comments (as described in section 31.8.2, this will be present in the gene-level results as well.

Targets: A list of the names of the targets forming the (minimal) CNV region forming the *entire* CNV region affecting this gene (and possibly other genes). Note however that the list is truncated to 100 characters. If you want to know the full list of targets inside the CNV region, you can use the target-level output track.

31.8.5 How to interpret fold-changes when the sample purity is not 100%

If your sample purity is less than 100%, it is necessary to take that into account when interpreting the fold-change values. Given a sample purity of $X\%$, and an amplification with an observed fold-change of F , the following formula gives the actual fold-change that would be seen if the sample were 100% pure:

$$\text{fold-change in 100\% pure sample} = \frac{F - 1}{X/100\%} + 1$$

For example, if the sample purity is 40%, and you have observed an amplification with a fold-change of 3, then the fold-change in the 100% pure sample would have been:

$$\text{fold-change in 100\% pure sample} = \frac{3.0 - 1}{40\%/100\%} + 1 = 6.0.$$

For a deletion the formula for converting an observed (absolute) fold-change to the actual (absolute) fold change is:

$$\text{fold-change in 100\% pure sample} = \frac{F \times X/100\%}{1 - F \times (1 - X/100\%)}$$

For example, if the sample purity is 40%, and you have a deletion with an absolute fold-change of 1.25, then the absolute fold-change in the 100% pure sample would have been:

$$\text{fold-change in 100\% pure sample} = \frac{1.25 \times 40/100\%}{1 - 1.25 \times (1 - 40/100\%)} = 2.0.$$

Figures 31.27 and 31.28 shows the 'true' fold changes for different observed fold-changes at different sample purities.

31.8.6 CNV results report

The report contains information about the results of the Copy Number Variant Detection (Targeted) tool.

Normalization The Normalization section gives information about the sample-level and chromosome-level coverages. Any chromosomes with disproportionate coverages are noted; targets on these chromosomes were ignored when setting up the statistical models.

Target-level log2-ratios The target-level coverage log2-ratios and predicted regional gains and losses are presented as a graph. Double click on it to see it in full view and access data points in a table.

An example is shown in figure 31.29. The following elements are included in the plots:

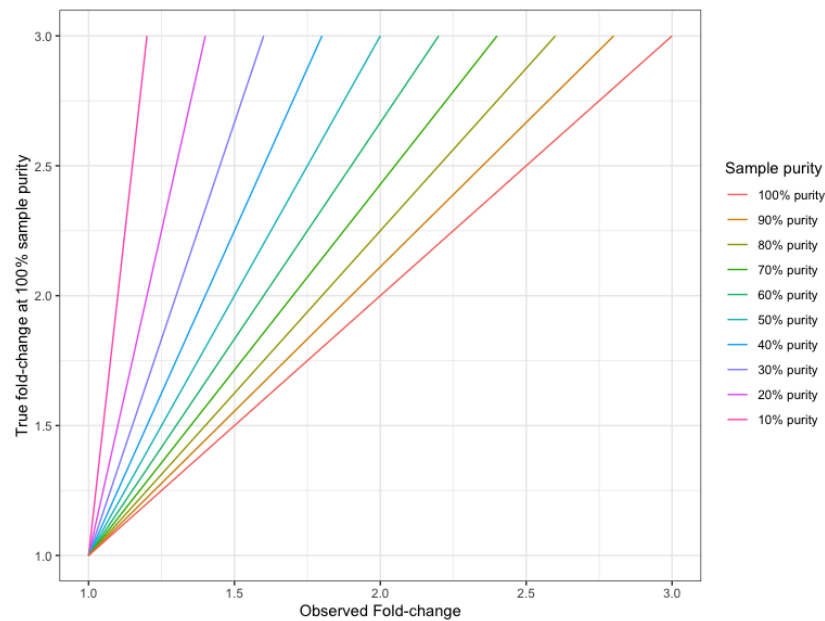


Figure 31.27: The true amplification fold-change in the 100% pure sample, for different observed fold-changes, as a function of sample purity.

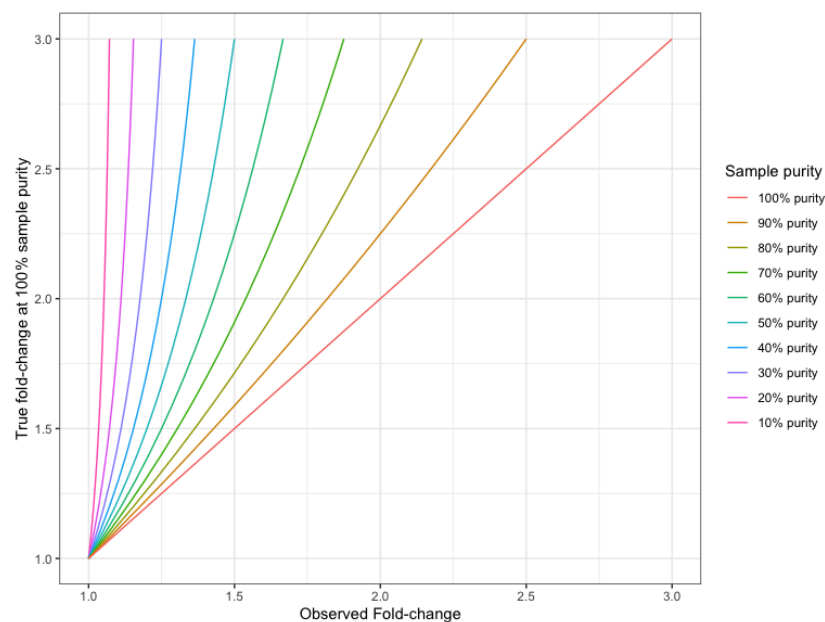


Figure 31.28: The true deletion fold-change in the 100% pure sample, for different observed fold-changes, as a function of sample purity.

- Targets are plotted by genomic position (horizontal axis) and observed mean adjusted coverage log₂-ratio (vertical axis). Targets are colored by their position relative to regional gains and losses. Red: The target is inside a predicted gain. Blue: The target is inside a predicted loss. Grey: The target is in a region with no predicted regional copy number change.
- Regional gains (red) and losses (blue) are plotted by genomic position (horizontal axis) and

mean coverage log2-ratio (vertical axis).

- 95 % confidence intervals of the expected mean adjusted log2-ratios of coverages based on the statistical model are plotted as grey dotted lines.

2.1 All chromosomes

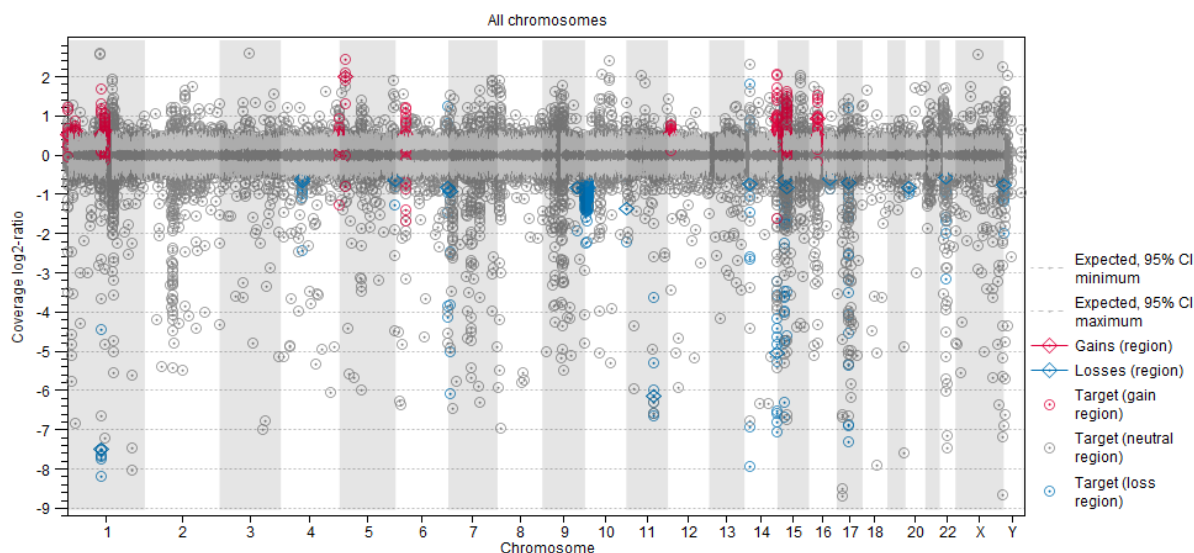


Figure 31.29: An example graph showing the mean adjusted log2-ratios of coverages in the report produced by the Copy Number Variant Detection (Targeted) tool. Targets located in regions where gains are predicted are red. Targets located in regions where losses are predicted are blue.

CNV statistics The last section in the report provides some information about the number of CNVs called in the region-level prediction results. If a gene track is provided as input, it also shows the number of genes affected by CNVs. It is possible for a gene to be affected by both a deletion and an amplification, if the gene overlap two different regions from the Region-level CNV results track. The number of uncalled or filtered regions are also shown.

31.8.7 CNV algorithm report

If you have chosen to produce an algorithm report in the output handling step of the wizard, an algorithm report will also be produced. This contains information about the statistical models of the algorithm, and can be used to evaluate how well the assumptions of the model were fulfilled. We will now present the different sections of this report.

Normalization and chromosome analysis

This section of the report is related to the first step of the Copy Number Variant Detection (Targeted) tool, where the chromosome-level coverages are analyzed to detect any outliers. The total coverages of the case chromosomes are plotted against the total coverages of the baseline, and the detected outliers are indicated. Chromosome coverages identified as disproportionate are marked with red crosses (see figure 31.30).

The graph is followed by a table, where the detailed chromosome coverages are shown after normalization. Chromosomes with disproportionate coverage and chromosomes without any

1.1 Chromosome coverage regression model

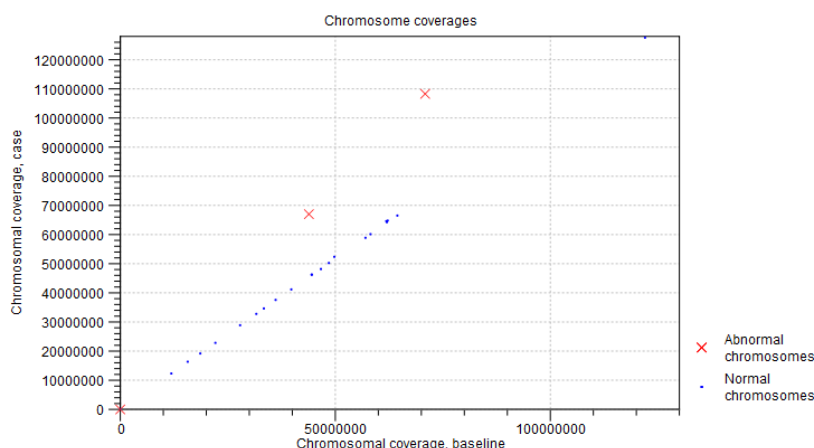


Figure 31.30: An example graph showing the coverages of the chromosomes in the case versus the baseline. In this example, three chromosomes are marked as abnormal. Two of these chromosomes are significantly amplified, and log-ratios of coverages of many targets on these chromosome are significantly higher than for targets on other chromosomes. The third outlier chromosome had zero coverage in both the case and the baseline.

targets are marked in the 'Comment' column. These chromosomes are the ones marked with red crosses in the graph in section 1.1. of the algorithm report, and these chromosomes were not used in the coverage normalization step.

Prediction of target-level CNVs

This section of the algorithm report gives information about the statistical models used to predict target-level CNVs.

Adjustment of log2-ratios The first two graphs in this section are related to the adjustment of the log-ratios of coverages as a function of log-coverage. The log-ratio of coverages for targets depends on the level of coverage of the target, as observed by Li et al. (Bioinformatics, 2012), who also proposed that a linear correction should be applied [Li et al., 2012]. In the first of the two graphs, the non-adjusted log-ratios of target coverages are plotted against the log-coverage of the targets. In the second graph, the mean log-ratios are plotted after adjustment (figure 31.31). If the model fits the data, we expect to see that the adjusted mean log-ratios are centered around 0 for all log-coverages, and the variation decreases with increasing log-coverage.

Statistical model for adjusted log2-ratios In this section of the algorithm report, you can see how well the algorithm was able to model the statistical variation in the log-ratios of coverages. An example is shown in figure 31.32). A good fit of the model to the data points indicates that the variance has been modeled accurately.

To make the points more visible, double-click the figure, to open it in a separate editor. Here, you can select how to visualize the data points and the fitted model. For example, you can choose to highlight the data points in the sidepanel:

MA Plot Settings | Dot properties | Dot type | "Dot"

Distribution of adjusted log2-ratios in bins One of the assumptions of the statistical model used

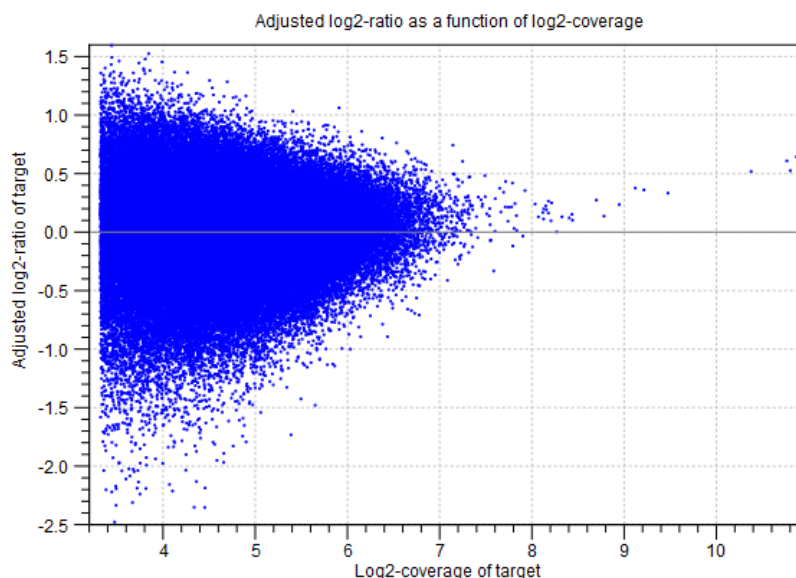


Figure 31.31: An example graph showing the mean adjusted log-ratios of coverages plotted against the log-coverages of targets, in the algorithm report of the Copy Number Variation Detection tool. Here, the adjusted mean log-ratios are centered around 0.0 for most coverages, and the variation decreases with increasing log-coverage. This indicates a good fit of the model. However, at very high coverages, the adjusted log-ratios are centered higher than 0.0, which indicates that for these coverages, the model is not a perfect fit. But only very few targets are affected by this, as the points are very sparse at these high coverage levels.

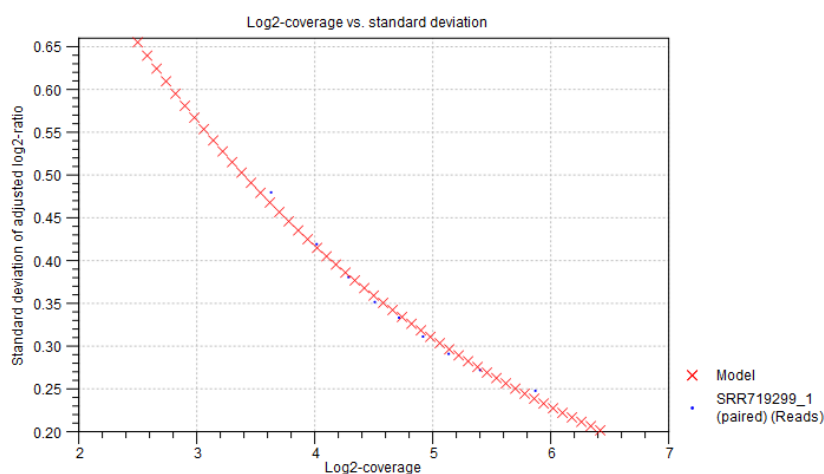


Figure 31.32: An example graph showing how the variance in the target-level mean log-ratios was modeled in the algorithm report of the Copy Number Variation Detection tool. Here, the data points are very close to the fitted model, indicating a good fit of the model to the data.

by the CNV detection tool is that the coverage log-ratios of targets are normally distributed with a mean of zero, and the variance only depends on the log-coverage of each target in the baseline. The bar charts in this section of the algorithm report show how well this assumption of the model fits the data. An example is shown in figure 31.33). A good fit of the model to the data points indicates that the variance has been modeled accurately.

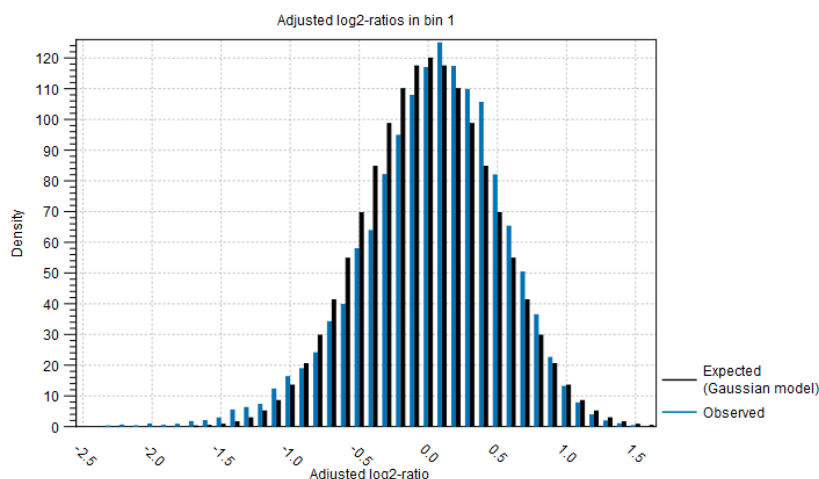


Figure 31.33: An example bar chart from the algorithm report of the Copy Number Variation Detection tool, showing how well the normal distribution assumption was fulfilled by the adjusted coverage log-ratios. Here, there is a good correspondence between the expected distribution and the observations.

Prediction of region-level CNVs

The final section of the algorithm report is related to the region-level CNV prediction. In this part of the algorithm, the chromosomes are segmented into regions of similar adjusted mean log-ratios. More segments lead to a reduced variance per segment; in the extreme, where every target forms its own segment, the variance is zero. However, more segments also mean that the model contains more free parameters, and is therefore potentially over-fitted. A value known as the Bayesian Information Criterion (BIC) gives an indication of the balance of these two effects, for any potential segmentation of a chromosome. The segmentation process aims to minimize the BIC, producing the best balance of accuracy and overfitting in the final segments.

The segmentation begins by identifying a set of potential breakpoints, known as local maximizers. The number of potential breakpoints at the start of the segmentation is shown in the "# local maximizers at start" column, and the corresponding BIC score is indicated in the "Start BIC" column. Breakpoints are removed strategically one-by-one, and the BIC score is calculated after each removal. When enough breakpoints have been removed for the BIC score to reach its minimum, the final number of breakpoints is shown in the "# local maximizers at end" column, and the corresponding BIC score is indicated in the "End BIC" column. A large reduction in the number of local maximizers indicates that it was possible to join many smaller CNV regions into larger ones.

Note: The segmentation process only produces regions of similar adjusted coverage log-ratios. Each segment is tested afterwards, to identify if it represents a CNV. Therefore, the number of segments shown in this table does not correspond to the number of CNVs actually predicted by the algorithm.

31.9 Identify Known Mutations from Sample Mappings

The **Identify Known Mutations from Sample Mappings** tool can be used to look up known genomic variants in read mappings. This can be done in one or more samples by comparing a

track of known variants with the read mappings of interest in order to test for the presence or absence of relevant variants in samples for example.

The **Identify Known Mutations from Sample Mappings** tool does not perform any kind of variant calling, which means that this tool cannot be used to find de novo variants. Rather, the tool is intended for identification of variants that have already been reported.

You need two types of input for the **Identify Known Mutations from Sample Mappings** tool:

- A variant track that holds the specific variants that you wish to test for.
- The read mapping(s) that you wish to check for the presence (or absence) of specific variants.

The **Identify Known Mutations from Sample Mappings** tool has two kinds of outputs:

- An overview track with information about:
 - whether the variant could be detected or not
 - whether the coverage was sufficient at the given position
 - the frequency of the variant in each sample.
- Individual output tracks for each sample that show the observed frequency, average base quality, forward/reverse read balance, zygosity and observed allele count.

31.9.1 Run the Identify Known Mutations from Sample Mappings tool

To run **Identify Known Mutations from Sample Mappings**, go to:

Tools | Resequencing Analysis  | **Variant Detection**  | **Identify Known Mutations from Sample Mappings** 

This opens the wizard shown where you can specify the read mapping(s) to analyze. Click **Next** to get the following options (figure 31.34):

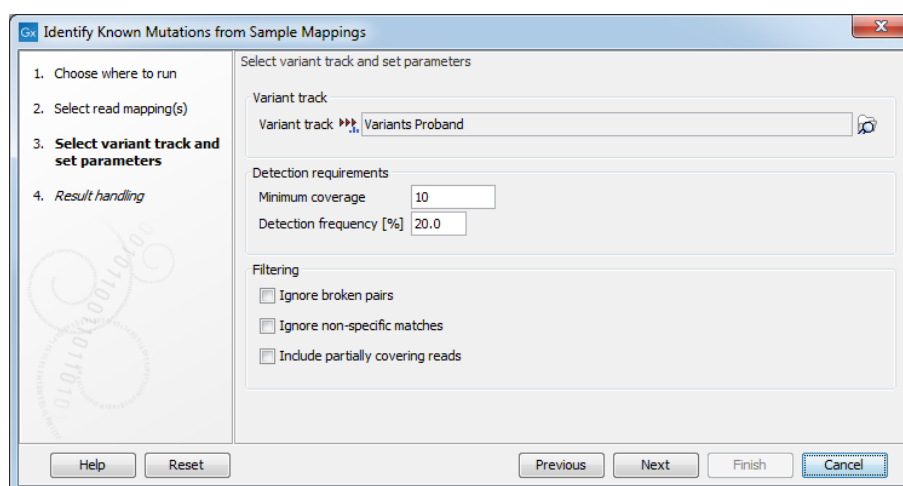


Figure 31.34: Select the variant track with the variants that you wish to use for variant testing.

Variant track

- **Variant track** Select the variant track that contains the specific variants that you wish to test for in your read mapping. **Note!** You can only select one variant track at the time. If you wish to compare with more than one variant track, you must run the analysis with each individual variant track at the time.

Detection requirements

- **Minimum coverage** The minimum number of reads that covers the position of the variant, which is required to set "Sufficient Coverage" to YES.
- **Detection frequency** The minimum allele frequency that is required to annotate a variant as being present in the sample. The same threshold will also be used to determine if a variant is homozygous or heterozygous. In case the most frequent alternative allele at the position of the considered variant has a frequency of less than this value, the zygosity of the considered variant will be reported as being homozygous.

Filtering

- **Ignore broken pairs** When ticked, reads from broken pairs are ignored. Broken pairs may arise for a number of reasons, one being erroneous mapping of the reads. In general, variants based on broken pair reads are likely to be less reliable, so ignoring them may reduce the number of spurious variants called. However, broken pairs may also arise for biological reasons (e.g. due to structural variants) and if they are ignored some true variants may go undetected.
- **Ignore non-specific matches** Reads that have an equally good match elsewhere on the reference genome (these reads are colored yellow in the mapping view) can be ignored in the analysis. Whether you include these reads or not will be a tradeoff between sensitivity and specificity. Including them may lead to the prediction of transcripts that are not correct, whereas excluding them may mean that you will lose some true transcripts.
- **Include partially covering reads** Reads that partially overlap variants (see the blue box below for a definition) will be considered to enable the detection of variants that are longer than the reads. When the "Include partially covering reads" option is disabled, only fully covering reads will be counted for all annotations. Enabling the "Include partially covering reads" option means that all fully covering reads will be counted for all annotations, and that additionally, partially covering reads will be included in relevant annotations including Coverage. Thus, if a partial read is compatible with multiple variants in the same region, the sum of all Counts for that region may be greater than the Coverage, and the sum of all Frequencies for that region may be higher than 100%.

A fully covering read is described as such:

- for SNV, MNV and Deletion: the read must cover all reference positions in the variant region.
- for Insertion and Replacement: the read must overlap adjacent reference positions on both sides of the variant region.

A partially covering read is read that is not fully covering the variant region, but overlaps with at least one residue.

Click **Next** to go to the next wizard step (figure 31.35). At this step the output options can be adjusted.

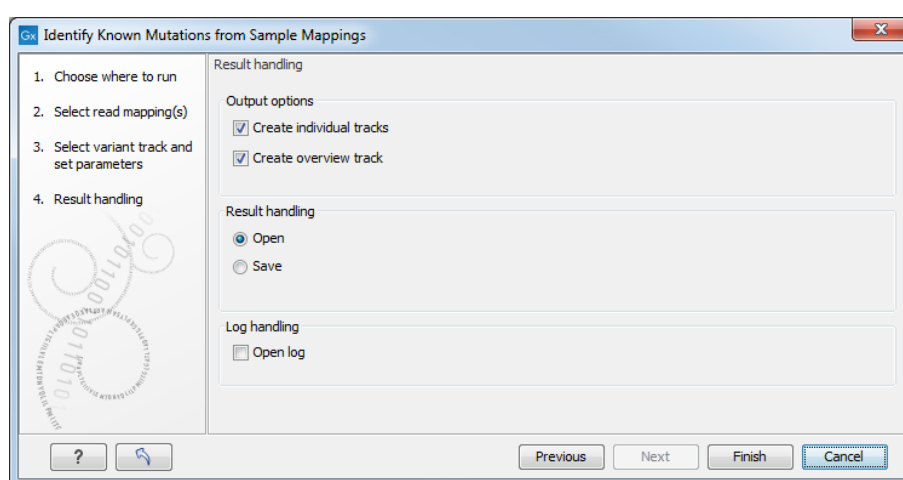


Figure 31.35: Select the desired output format(s). If using the default settings, two types of output will be generated; individual tracks and overview tracks.

The output options are:

- **Create individual track** For each read mapping an individual track is created with the observed frequency, average base quality, forward/reverse read balance, zygosity and observed allele count.
- **Create overview track** The overview track is a summary for all samples with information about whether the coverage is sufficient at a given variant position and if the variant has been detected; the frequency of the variant.

Specify where to save the results and click on the button labeled **Finish**.

31.9.2 Output from the Identify Known Mutations from Sample Mappings tool

One individual sample output track will be created for each read mapping analyzed, while one overview track will be created per analysis (figure 31.36).

At the bottom of the window it is possible to switch to a table view that lists all the mutations from the variant track that were found in your sample mapping.

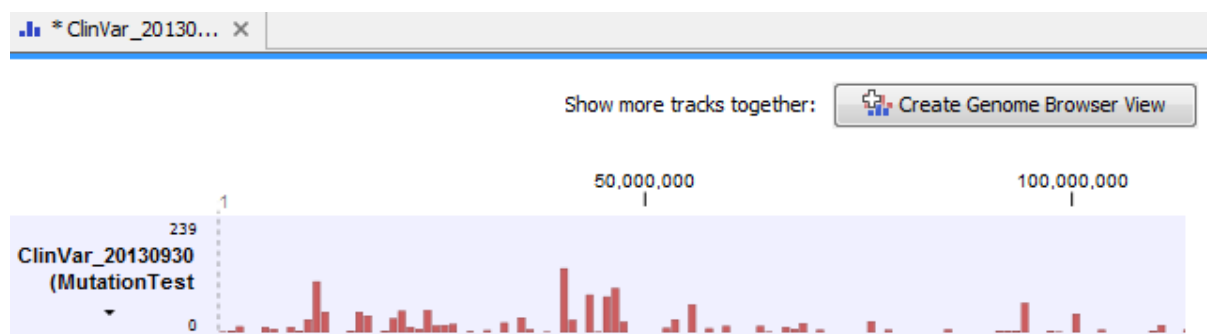


Figure 31.36: Overview track of read mappings tested against a Clinvar variant track.

In the individual track, the variant has been annotated with most classical variant track annotations (see section 31.6.1), as well as:

- **Most frequent alternative allele (MFAA)**
- **MFAA count** The Most Frequent Alternative Allele count (MFAA count) is the count of reads supporting the most frequent alternative allele at the position of the variant
- **MFAA frequency** Frequency of reads supporting the most frequent alternative allele at the position of the variant
- **MFAA forward read count** forward reads supporting the most frequent alternative allele at the position of the variant
- **MFAA reverse read count** reverse reads supporting the most frequent alternative allele at the position of the variant
- **MFAA forward/reverse balance** forward/reverse balance of the most frequent alternative allele at the position of the variant
- **MFAA average quality** average quality of the most frequent alternative allele at the position of the variant

In the overview track the variant has been annotated with the following information:

- **("Sample name") coverage** Either Yes or No, depending on whether the coverage at the position of the variant was higher or lower than the user given threshold for minimum coverage.
- **("Sample name") detection** Either Yes or No, depending on the minimum frequency settings chosen by the user.
- **("Sample name") frequency** The variant frequency observed in this sample.
- **("Sample name") zygosity** The zygosity observed in the sample. This setting is based on the minimum frequency setting made by the user. If this variant has been detected and the most frequent alternative allele at this position is also over the cutoff, the value is heterozygote.

An example of the individual and overview tables can be seen in figure 31.37.

Figure 31.37: Table views of the individual track (left) and overview track (right).

31.10 InDels and Structural Variants

The InDels and Structural Variants tool is designed to identify structural variants such as insertions, deletions, inversions and tandem duplications in read mappings and is primarily developed for short read technologies (such as Illumina reads). The tool relies *exclusively* on information derived from unaligned ends (also called 'soft clippings') of the reads in the mappings. This means that:

- The tool will detect NO structural variants if there are NO reads with unaligned ends in the read mapping.
- Read mappings made with the Map Reads to Reference tool with the '*global*' option switched on will have NO unaligned ends and the InDels and Structural Variants tool will thus find NO structural variants on these. (The '*global*' option means that reads are aligned in their entirety - irrespective of whether that introduces mismatches towards the ends of the reads. In the '*local*' option such reads will be mapped with unaligned ends).
- Read mappings based on really short reads (say, below 35 bp) are not likely to produce many reads with unaligned ends of any useful length, and the tool is thus not likely to produce many structural variant predictions for these read mappings.
- Read mappings generated with the Large Gap Read Mapping tool of the Transcript Discovery plugin are NOT optimal for the detection of structural variants with this tool. This is because this tool will map some reads with (large) gaps that would be mapped with unaligned ends with standard read mappers. This results in a weaker unaligned end signal in these mappings for the InDels and Structural Variants tool to work with.

In its current version the InDels and Structural Variants tool has the following known limitations:

- It will only detect intra-chromosomal structural variants.
- It can only process reads that are shorter than 5000 bp, reads that are longer are discarded.

31.10.1 Run the InDels and Structural Variants tool

Go to:

Tools | Resequencing Analysis | **Variant Detection** | **InDels and Structural Variants**

This will open up a dialog. Select the read mapping of interest as shown in figure 31.38 and click on the button labeled **Next**.

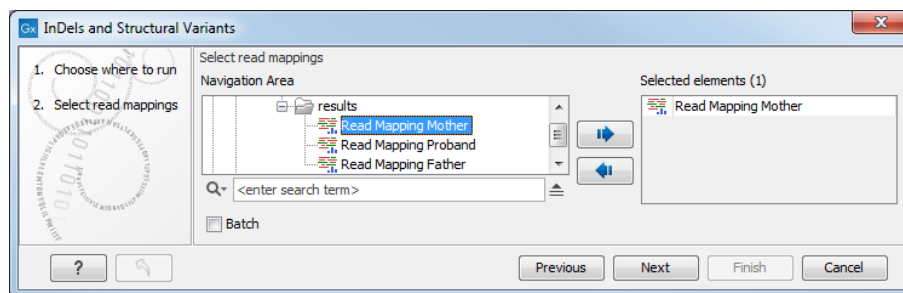


Figure 31.38: Select the read mapping of interest.

The next wizard step (Figure 31.39) is concerned with specifying parameters related to the algorithm used for calling structural variants. The algorithm first identifies positions in the mapping(s) with an excess of reads with left (or right) unaligned ends. Once these positions and the consensus sequences of the unaligned ends are determined, the algorithm maps the determined consensus sequences to the reference sequence around other positions with unaligned ends. If mappings are found that are in accordance with a 'signature' of a structural variant, a structural variant is called.

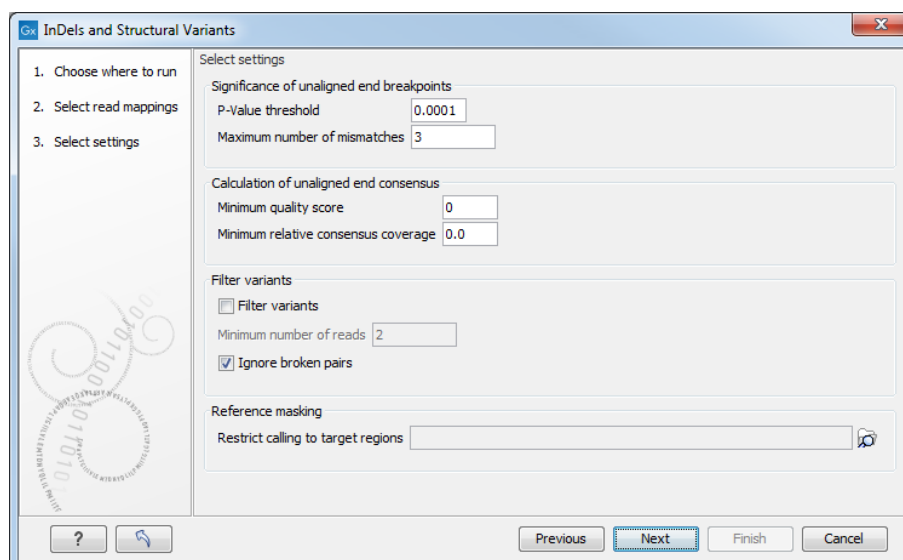


Figure 31.39: Select the relevant settings.

The '**Significance of unaligned end breakpoints**' parameters are concerned with when a position with unaligned ends should be considered by the algorithm, and when it should be ignored:

- **P-value threshold:** Only positions in which the fraction of reads with unaligned ends is sufficiently high will be considered. The 'P-value threshold' determines the cut-off value in a Binomial Distribution for this fraction. The higher the P-value threshold is set, the more unaligned breakpoints will be identified.
- **Maximum number of mismatches:** The 'Maximum number of mismatches' parameter determines which reads should be considered when inferring unaligned end breakpoints. Poorly map reads tend to have many mis-matches and unaligned ends, and it may be preferable to let the algorithm ignore reads with too many mis-matches in order to avoid

false positives and reduce computational time. On the other hand, if the allowed number of mis-matches is set too low, unaligned end breakpoints in proximities of other variants (e.g. SNVs) may be lost. Again, the higher the number of mis-matches allowed, the more unaligned breakpoints will be identified.

The **Calculation of unaligned end consensus** parameters can improve the calculation of the unaligned end consensus by removing bases according to:

- **Minimum quality score:** quality score under which bases should be ignored.
- **Minimum relative consensus coverage:** consensus coverage threshold under which bases should be ignored. The relative consensus coverage is calculated by taking the coverage at the current nucleotide position and dividing by the maximum coverage obtained along the unaligned ends upstream from this position. When the value thus calculated falls below the specified threshold, consensus generation stops. The idea behind the "Minimum relative consensus coverage" option is to stop consensus generation when dramatic drops in coverage are observed. For example, a drop from 1000 coverage to 10 coverage would give a relative consensus coverage of $10/1000 = 0.01$.

The **'Filter variants'** parameters are concerned with the amount of evidence for each structural variant required for it to be called:

- **Filter variants:** When the **Filter variants** box is checked, only variants that are inferred by breakpoints that *together* are supported by at least the specified **Minimum number of reads** will be called.
- **Ignore broken pairs:** This option is checked by default, but it can be unchecked to include variants located in broken pairs.

'Reference masking' allows specification of target regions:

- **Restrict calling to target regions:** When specifying a target region track only reads that overlap with at least one of the targets will be examined when the unaligned end breakpoints are identified. Hence only breakpoints that fall within, or in close proximity of, the targets will be identified (a read may overlap a target, but have an unaligned end outside the target - these are also identified and therefore breakpoints outside, but in the proximity of the target). The runtime will be decreased when you specify a target track as compared to when you do not.

Note! As the set of identified unaligned end breakpoints differs between runs where a target region track has been specified and where it has not, the set of predicted indels and structural variants is also likely to differ. This is because the indels and structural variants are predicted from the mapping patterns of the unaligned ends at the set of identified breakpoints. This is also the case even if you restrict the comparison to only involve the indels and structural variants detected within the target regions. You cannot expect these to be exactly the same but you can expect a large overlap.

Specify these settings and click **Next**. The "Results handling" dialog (Figure 31.40) will be opened. The Indels and Structural Variants tool has the following output options:

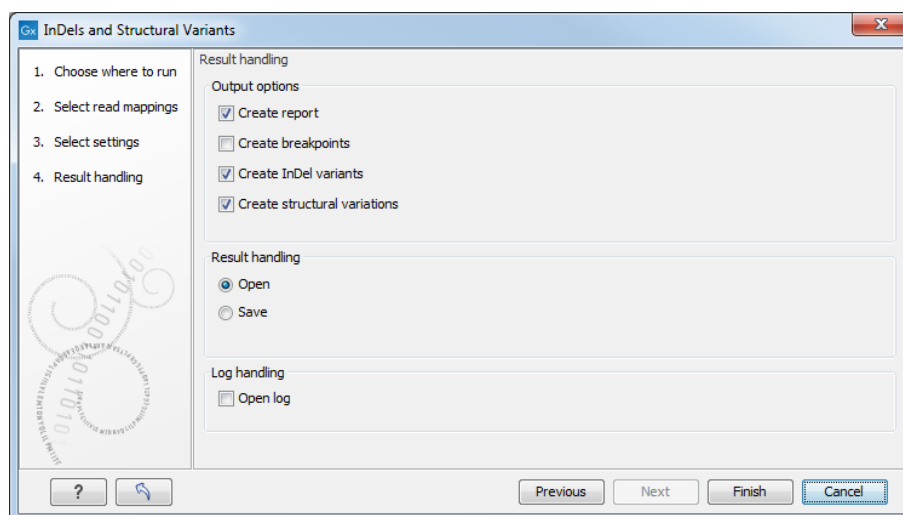


Figure 31.40: Select output formats.

- **Create report** When ticked, a report that summarizes information about the inferred breakpoints and variants is created.
- **Create breakpoints** When ticked, a track containing the detected breakpoints is created.
- **Create InDel variants** When ticked, a variant track containing the detected indels that fulfill the requirements for being 'variants' is created. These include:
 - the detected insertions for which the allele sequence is inferred, but not those for which it is not known, or only partly known. As the algorithm relies on mapping two unaligned ends against each other for detecting insertions *with inferred allele sequence*, the maximum length of these that can potentially be detected depends on (1) the read length and (2) the "length fraction" parameter of the read mapper. With current read lengths and default settings you are unlikely to get insertions *with inferred allele sequence* larger than a couple of hundred, and hence will not see insertions in this track larger than that.
 - medium sized deletions (those between six and 405 bp). All other deletions are put in the "Structural variants" track. The reason for not including all detected deletions in the indel track is that the main intended use of this track is to guide re-alignment. In our experience, the re-alignment algorithm performs best when only including the medium sized events. Notice that, in contrast to insertions, there is no upper limit on the length of deletions *with inferred allele sequence* that the algorithm can detect. This is because the allele sequence is trivial for deletions, whereas for insertions it must be inferred from the nucleotides in the unaligned ends.

See section 31.6.1 for a definition of the requirements for 'variants'. Note that insertions and deletions that are not included in the InDel track, will be present in the 'Structural Variant track' (described below).

- **Create structural variations** When ticked, a track containing the detected structural variants is created, including the insertions *with unknown allele sequence* and the deletions that are not included in the "InDel" track.

An example of the output from the InDels and Structural Variant tool is shown in figure 31.41. The output is described in detail in section 31.10.2.

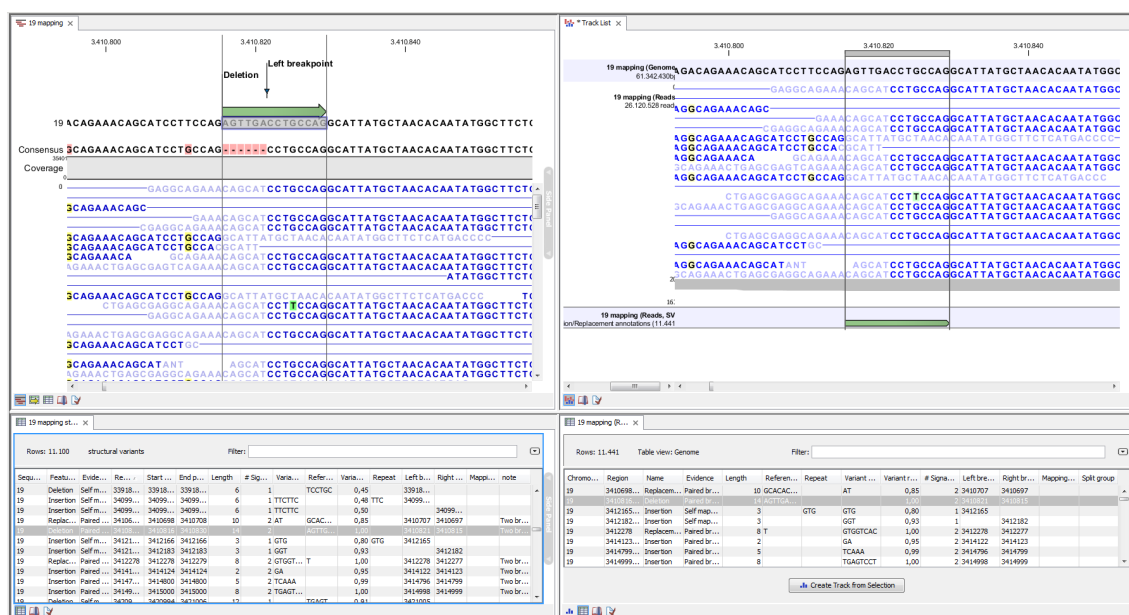


Figure 31.41: Example of the result of an analysis on a standalone read mapping (to the left) and on a reads track (to the right).

31.10.2 The Structural Variants and InDels output

The Structural Variants and InDels report The report gives an overview of the numbers and types of structural variants found in the sample. It contains

- A table listing the total number of reads in the read mapping and the number of reads that were discarded based on length.
- A table with a row for each reference sequence, and information on the number of breakpoint signatures and structural variants found.
- A table giving the total number of left and right unaligned end breakpoint signatures found, and the total number of reads supporting them. Note that paired-end reads are counted once.
- A distribution of the logarithm of the sequence complexity of the unaligned ends of the left and right breakpoint signatures (see section 31.10.5 for how the complexity is calculated).
- A distribution of the length of the unaligned ends of the left and right breakpoint signatures.
- A table giving the total number of the different types of structural variants found.
- Plots depicting the distribution of the lengths of structural variants identified.

The Breakpoint track (BP) The breakpoint track contains a row for each called breakpoint with the following information:

- **Chromosome** The chromosome on which the breakpoint is located.
- **Region** The location on the chromosome of the breakpoint.

- **Name** The type of the breakpoint ('left breakpoint' or 'right breakpoint').
- **p-value** The p-value (in the Binomial distribution) of the unaligned end breakpoint.
- **Unaligned** The consensus sequence of the unaligned ends at the breakpoint.
- **Unaligned length** The length of the consensus sequence of the unaligned ends at the breakpoint.
- **Mapped to self** If the unaligned end sequence at the breakpoint was found to map back to the reference in the vicinity of the breakpoint itself, a 'Deletion' or 'Insertion' based on 'self-mapping' evidence is called. This column will contain 'Deletion' or 'Insertion' if that is the case, or be empty if the unaligned end did not map back to the reference in the vicinity of the breakpoint itself.
- **Perfect mapped** The number of 'perfect mapped' reads (paired-end reads count as one). This number is intended as a proxy for the number of reads that fit with the reference sequence. When calculating this number we consider all reads that extend across the breakpoint. We ignore reads that are non-specifically mapped, in a broken pair, or has more than the **maximum number of mismatches**. A read is perfectly mapped if (1) it has no insertions or deletions (mismatches are allowed) and (2) it has no unaligned end.
- **Not perfect mapped** The number of 'not perfect mapped' reads (paired-end reads count as one). This number is intended as a proxy for the number of reads that fit with the predicted indel. When calculating this number we consider all reads that extend across the breakpoint or that has an unaligned end starting at the breakpoint. We ignore reads that are non-specifically mapped, in a broken pair, or has more than the **maximum number of mismatches**. A read is not perfect mapped if (1) it has an insertion or deletion or (2) it has an unaligned end.
- **Fraction non-perfectly mapped** the 'Non perfect mapped' divided by the 'Non perfect mapped' + 'Perfect mapped'.
- **Sequence complexity** The sequence complexity of the unaligned end of the breakpoint (see section 31.10.5 for how the sequence complexity is calculated).
- **Reads** The number of reads supporting the breakpoint (paired-end reads count as one).

Note that typically, breakpoints will be found for which it is not possible to infer a structural variant. There may be a number of reasons for that: (1) the unaligned ends from which the breakpoint signature was derived might not be caused by an underlying structural variant, but merely be due to read mapping issues or noise, or (2) the breakpoint(s) which the detected breakpoint should have been matched to was/were not detected, and therefore no matching breakpoint(s) were found. Breakpoints may go un-detected either because of lack of coverage in the breakpoint region or because they are located within regions with exclusively non-uniquely mapped reads (only unaligned ends of uniquely mapping reads are used).

The InDel variant track (InDel) The InDel variant track contains a row for each of the called insertions or deletions. These are the small to medium sized insertions, as well as deletions up to 405 bp in length, for which the algorithm was able to identify the allele sequence, i.e., the exact inserted or deleted sequence.

For insertions, the full allele sequence is found from the unaligned ends of mapped reads. For some insertions, the length and allele sequence cannot be determined and as these do not fulfill the requirements of a 'variant', they do not qualify for representation in the InDel Variant track but instead appear in the Structural Variant track (see below).

The information provided for each of the indels in the InDel Variant track is the 'Chromosome', 'Region', 'Type', 'Reference', 'Allele', 'Reference Allele', 'Length' and 'Zygosity' columns that are provided for all variants (see section 31.6.1). Note that the Zygosity field is set to 'Homozygous' if the 'Variant ratio' is 0.80 or above, and 'Heterozygous' otherwise.

In addition, the track provides the following information, primarily to assess the degree of evidence supporting each predicted indel:

- **Evidence** The mapping evidence on which the call of the indel was based. This may be either 'Self mapped', 'Paired breakpoint', 'Cross mapped breakpoint' or 'Tandem duplication' depending on the mapping signature of the unaligned ends of the breakpoint(s) from which the indel was inferred.
- **Repeat** The algorithm attempts to identify if the variant sequence contains perfect repeats. This is done by searching the region around the structural variant for perfect repeat sequences. The region searched is 3 times the length of variant around the insertion/deletion point. The maximum repeat length searched for is 10. If a repeat sequence is found, the repeated sequence is given in this column. If not, the column is empty.
- **Variant ratio** This column contains the sum of the 'Non perfect mapped' reads for the breakpoints used to infer the indel, divided by the sum of the 'Non perfect mapped' and 'Perfect mapped' reads for the breakpoints used to infer the indel (see section the description above of the breakpoint track). This fraction is intended to give a hint towards the zygosity of the indel. The closer the value to 1, the higher the likelihood that the variant is homozygous.
- **# Reads** The total number of reads supporting the breakpoints from which the indel was constructed (paired-end reads count as one).
- **Sequence complexity** The sequence complexity of the unaligned end of the breakpoint (see section 31.10.5). Indels with higher complexity are typically more reliable than those with low complexity.

The Structural Variant track (SV) The Structural Variant track contains a row for each of the called structural variants that are not already reported in the InDel track. It contains the following information:

- **Chromosome** The chromosome on which the structural variant is located.
- **Region** The location on the chromosome of the structural variant.
- **Name** The type of the structural variant ('deletion', 'insertion', 'inversion', 'replacement' or 'complex').
- **Evidence** The breakpoint mapping evidence, i.e., the 'unaligned end' signature on which the call of the structural variant was based. This may be either 'Self mapped', 'Paired

breakpoint', 'Cross mapped breakpoints', 'Cross mapped breakpoints (invalid orientation)', 'Close breakpoints', 'Multiple breakpoints' or 'Tandem duplication', depending on which type of signature that was found.

- **Length** the length of the allele sequence of the structural variant. Note that the length of variants for which the allele sequence could not be determined is reported as 0 (e.g insertions inferred from 'Close breakpoints').
- **Reference sequence** The sequence of the reference in the region of the structural variant.
- **Variant sequence** The allele sequence of the structural variant if it is known. If not, the column will be empty.
- **Repeat** The same as in the InDel track.
- **Variant ratio** The same as in the InDel track.
- **Signatures** The number of unaligned breakpoints involved in the signature of the structural variant. In most cases these will be pairs of breakpoints, and the value is 2, however some structural variants that have signatures involving more than two breakpoint (see section 31.10.4). Typically structural variants of type 'complex' will be inferred from more than 2 breakpoint signatures.
- **Left breakpoints** The positions of the 'Left breakpoints' involved in the signature of the structural variant.
- **Right breakpoints** The positions of the 'Right breakpoints' involved in the signature of the structural variant.
- **Mapping scores fraction** The mapping scores of the unaligned ends for each of the breakpoints. These are the similarity values between the unaligned end and the region of the reference to which it was mapped. The values lie between 0 and 1. The closer the value is to 1, the better the match, suggesting better reliability of the inferred variant.
- **Reads** The total number of reads supporting the breakpoints from which the indels was constructed.
- **Sequence complexity** The sequence complexity of the unaligned end of the breakpoint (see section 31.10.5).
- **Split group** Some structural variants extend over a very large a region. For these visualization is challenging, and instead of reporting them in a single row we split them in multiple rows - one for each 'end' of the variant. To allow the user to see which of these 'split features' belong together, we give features that belong to the same structural variant a common 'split group' identifier. If the column is empty the structural variant is not split, but contained within a single row.

31.10.3 The InDels and Structural Variants detection algorithm

The Indels and Structural Variants detection algorithm has two steps:

1. Identify 'breakpoint signatures': First, the algorithm identifies positions in the mapping(s) with an excess of reads with left (or right) unaligned ends. For each of these, it creates a Left breakpoint (LB) or Right breakpoint (RB) signature.

2. Identify 'structural variant signatures': Secondly, the algorithm creates structural variant signatures from the identified breakpoint signatures. This is done by mapping the consensus unaligned ends of the identified LB and RB signatures to selected areas of the references as well as to each other. The mapping patterns of the consensus unaligned ends are examined and structural variant annotations consistent with the mapping patterns are created.

Step 1: Creating Left- and Right breakpoint signatures

In the first step of the InDels and Structural Variants detection algorithm points in the read mapping are identified which have a significant proportion of reads mapped with unaligned ends. There are typically numerous reads with unaligned ends in read mappings – some are due to structural variants in the sample relative to the reference, others are due to poorly mapped, or poor quality reads. An example is given in figure 31.42. In order to make reliable predictions, attempts must be made to distinguish the unaligned ends caused by noisy read(mappings) from those caused by structural variants, so that the signal from the structural variants comes through as clearly as possible – both in terms of where the 'significant' unaligned ends are and in terms of what they look like.



Figure 31.42: Example of a read mapping containing unaligned ends with three unaligned end signatures.

To identify positions with a 'significant' portion of 'consistent' unaligned end reads we first estimate 'null-distributions' of the fractions of left and right unaligned end reads at each position in the read mapping, and subsequently use these distributions to identify positions with an 'excess' of unaligned end reads. In these positions we create a Left (LB) or Right (RB) breakpoint signature. To estimate the null-distributions we:

1. Calculate the coverage, c_i , in each position, i of all uniquely mapped reads (Non-specifically mapped reads are ignored. Furthermore, for paired read data sets, only intact paired reads pairs are considered – broken paired reads are ignored).
2. Calculate the coverage in each position of 'valid' reads with a starting left unaligned end, l_i (of minimum consensus length 3bp).
3. Calculate the coverage in each position of 'valid' reads with a starting right unaligned end, r_i (of minimum consensus length 3bp).

We then use the observed fractions of 'Left unaligned ends' ($\sum_i l_i / \sum_i c_i$) and 'Right unaligned ends' ($\sum_i r_i / \sum_i c_i$) as frequencies in binomial distributions of 'Left unaligned end' and 'Right unaligned end' read fractions. We go through each position in the read mapping and examine it for an excess of left (or right) unaligned end reads: if the probability of obtaining the observed number of left (or right) unaligned ends in a position with the observed coverage, is 'small', a Left breakpoint signature (LB), respectively Right breakpoint signature (RB), is created.

The two user-specified settings 'The P-value threshold' and the 'Maximum number of mismatches' determine which breakpoint signatures the algorithm will detect (see section 31.10.1 and figure 31.39). The p-value is used as a cutoff in the binomial distributions estimated above: if the probability of obtaining the observed number of left (or right) unaligned ends in a position with the observed coverage, is smaller than the user-specified cut-off, a Left breakpoint signature (LB), respectively Right breakpoint signature (RB), is created. The 'Maximum number of mis-matches' parameter is used to determine which reads are considered 'valid' unaligned end reads. Only reads that have at most this number of mis-matches in their aligned parts are counted. The higher these two values are set, the more breakpoints will be called. The more breakpoints are called, the larger the search space for the Structural variation detection algorithm, and thus the longer the computation time.

In figure 31.42, three unaligned end signatures are shown. The left-most LB signature is called only when the p-value cut-off is chosen high (0.01 as opposed to 0.0001).

Step 2: Creating Structural variant signatures

In the second step of the InDels and Structural Variants detection algorithm the unaligned end 'breakpoint signatures' (identified in step 1) are used to derive 'structural variant signatures'. This is done by:

1. Generating a consensus sequence of the reads with unaligned ends at each identified breakpoint.
2. Mapping the generated consensus sequences against the reference sequence in the regions around *other* identified breakpoints ('cross-mapping').
3. Mapping the generated consensus sequences of breakpoints that are *near each other against each other* ('aligning').
4. Mapping the generated consensus sequences against the reference sequence in the *region around the breakpoint itself* ('self-mapping').
5. Considering the breakpoints whose unaligned end consensus sequences are found to cross map against each other together, and compare their mapping patterns to the set of theoretically expected 'structural variants signatures' (see section 31.10.4).
6. Creating a 'structural variant signature' for each of the groups of breakpoints whose mapping patterns were in accordance with one of the expected 'structural variants signatures'.

A structural variant is called for each of the created 'structural variant signatures'. For each of the groups of breakpoints whose mapping patterns were NOT in accordance with one of the expected 'structural variants signatures', we call a structural variant of type 'complex'.

The steps above require a number of decisions to be made regarding (1) When is the consensus sequence reliable enough to work with?, and (2) When does an unaligned end map well enough that we will call it a match? The algorithm uses a number of hard-coded values when making those decisions. The values are described below.

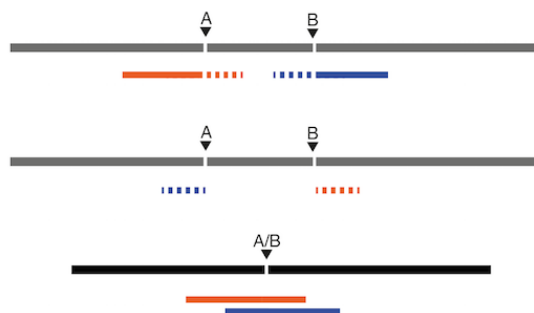
Algorithmic details

- **Generating a consensus:** The consensus of the unaligned ends is calculated by simple alignment without gaps. Having created the consensus, we exclude the unaligned ends which differ by more than 20% from the consensus, and recalculate the consensus. This prevents 'spuriously' unaligned ends that extend longer than other unaligned ends from impacting the tail of the consensus unaligned end.
- **Mapping of the consensus:**
 - **'Cross mapping':** When mapping the consensus sequences against the reference sequence around other breakpoints we require that:
 - * The consensus is at least 16 bp long.
 - * The score of the alignment is at least 70% of the maximal possible score of the alignment.
 - **'Aligning':** When aligning the consensus sequences two closely located breakpoints against each other we require that:
 - * The breakpoints are within a 100 bp distance of each other.
 - * The overlap in the alignment of the consensus sequences is least 4 nucleotides long.
 - **'Self-mapping':** When mapping the consensus sequences of breakpoints against the reference sequence in a region around the breakpoint itself we require that:
 - * The consensus is at least 9 bp long.
 - * A match is found within 400 bp window of the breakpoint.
 - * The score of the alignment is at least 90% of the maximal possible score of the alignment of the part of the consensus sequence that does not include the variant allele part.

31.10.4 Theoretically expected structural variant signatures

Different types of structural variants will leave different 'signatures' in terms of the mapping patterns of the unaligned ends. The 'structural variant signatures' of the set of structural variants that are considered by the Indels and Structural variant tool are drawn in figures 31.43 to 31.50.

Deletion - cross mapped/paired breakpoints evidence



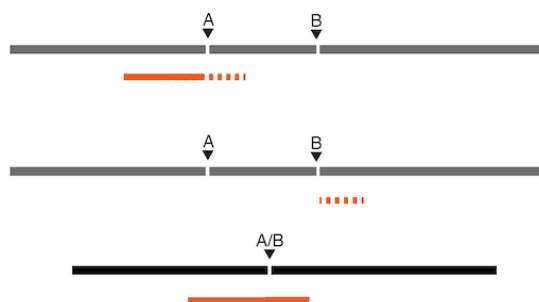
Sample reads map to reference genome.
Unaligned ends detected between points A and B.

Unaligned ends map to reference at the opposite
sides of points A and B.

Conclusion: Sample genome contains a deletion
between A and B.

Figure 31.43: A deletion with cross-mapping breakpoint evidence.

Deletion - selfmapped evidence



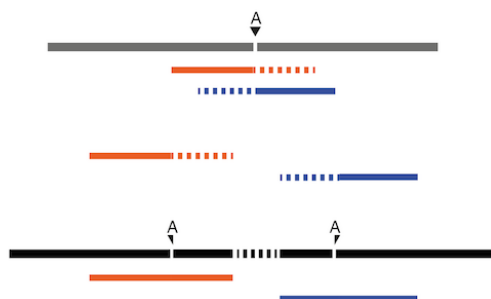
Sample read maps to reference genome before
point A.
Unaligned end detected between points A and B.

Unaligned end maps to reference after point B.

Conclusion: Sample genome contains a deletion
between A and B.

Figure 31.44: A deletion with selfmapping breakpoint evidence.

Insertion - close breakpoints evidence

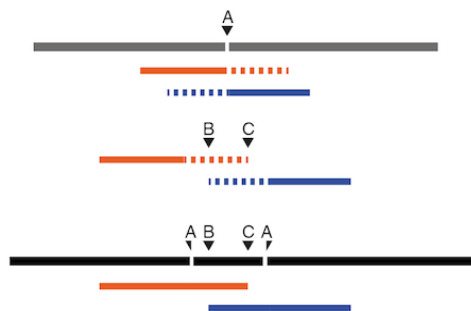


Sample reads map to reference genome at point A.
Unaligned end detected at either side of point A.

Unaligned ends do not map to each other.

Conclusion: Sample genome contains an insertion.
The inserted sequence cannot be fully deduced.

Figure 31.45: An insertion with close breakpoint evidence.

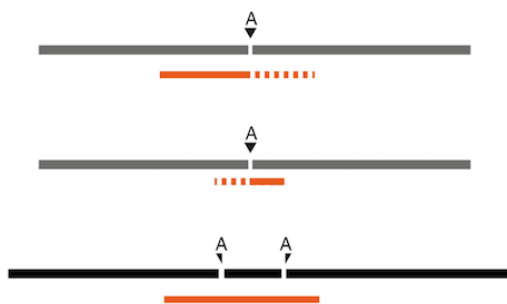
Insertion - crossedmapped evidence

Sample reads map to reference genome at point A.
Unaligned end detected at either side of point A

Unaligned ends map to each other between points B and C

Conclusion: Sample genome contains an insertion.
The inserted sequence can be deduced

Figure 31.46: An insertion with cross-mapped breakpoints evidence.

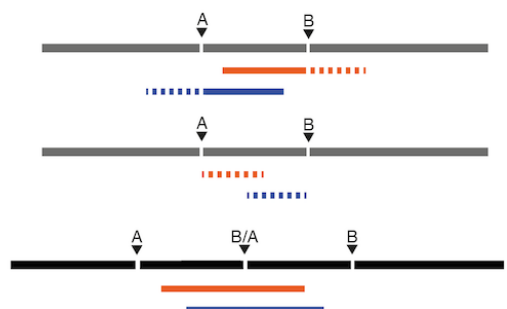
Insertion - selfmapped evidence

Sample reads map to reference genome before point A.
Unaligned end detected after point A.

Terminal part of the unaligned end maps at other side of A.

Conclusion: Sample genome contains an insertion.
The inserted sequence can be deduced.

Figure 31.47: An insertion with selfmapped breakpoint evidence.

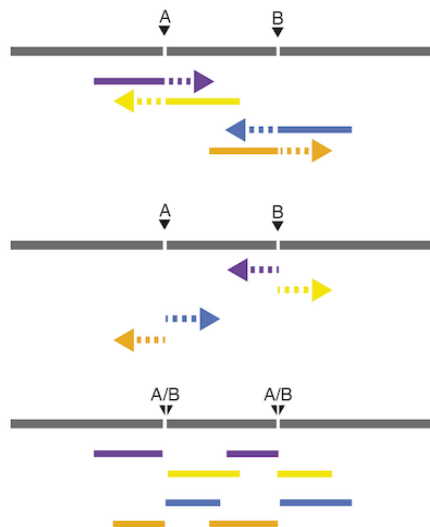
Insertion - tandem duplication

Sample reads map to the reference genome between points A and B.
Unaligned ends are detected before A and after B.

Unaligned ends map before B and after A.

Conclusion: Sample genome contains a tandem duplication insertion.

Figure 31.48: An insertion with breakpoint mapping evidence corresponding to a 'Tandem duplication'.

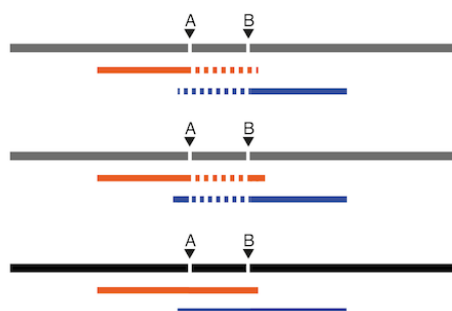
Inversion - crossmapped/paired breakpoints

Unaligned ends are detected at two positions.

Unaligned ends from point A map at B reverse translated and vice versa.

Conclusion: The sample genome contains an inversion of the region between A and B.

Figure 31.49: *The unaligned end mapping pattern of an inversion.*

Replacement

Unaligned ends are detected between A and B.

Unaligned ends map to each other.
The consensus alignment of the reads map to the reference with a replaced internal sequence.

Conclusion: Sample genome contains a replacement between points A and B.

Figure 31.50: *The unaligned end mapping pattern of a replacement.*

31.10.5 How sequence complexity is calculated

The sequence complexity of an unaligned end is calculated as the product of 'the observed vocabulary-usages' divided by 'the maximal possible vocabulary-usages', for word sizes from one to seven. When multiple breakpoints are used to construct a structural variant, the complexity is calculated as the product of the individual sequence complexities of the breakpoints constituting the structural variant.

The observed vocabulary usage for word size, k , for a given sequence is the number of different "words" of size k that exist in that sequence. The maximal possible vocabulary usage for word size k for a given sequence is the maximal number of different words of size k that can possibly be observed in a sequence of a given length. For DNA sequences, the set of all possible letters in such words is four, that is, there are four letters that represent the possible nucleotides: A, C, G and T. The calculation is most easily described using an example.

Consider the sequence CAGTACAG. In this sequence we observe:

- 4 different words of size 1 ('A', 'C', 'G' and 'T').
- 5 different words of size 2 ('CA', 'AG', 'GT', 'TA' and 'AC') Note that 'CA' and 'AG' are found twice in this sequence.
- 5 different words of size 3 ('CAG', 'AGT', 'GTA', 'TAC' and 'ACA') Note that 'CAG' is found twice in this sequence.
- 5 different words of size 4 ('CAGT', 'AGTA', 'GTAC', 'TACA' and 'ACAG')
- 4 different words of size 5 ('CAGTA', 'AGTAC', 'GTACA' and 'TACAG')
- 3 different words of size 6 ('CAGTAC', 'AGTACA' and 'GTACAG')
- 2 different words of size 7 ('CAGTACA' and 'AGTACAG')

Note that we only do the calculations for word sizes up to 7, even when the unaligned end is longer than this.

Now we consider the maximal possible number of words we could observe in a DNA sequence of this length, again restricting our considerations to word of size of 7.

- Word size of 1: The maximum number of different letters possible here is 4, the single characters, A, G, C and T. There are 8 positions in our example sequence, but there are only 4 possible unique nucleotides.
- Word size of 2: The maximum number of different words possible here is 7. For DNA generally, there is a total of 16 different dinucleotides (4×4). For a sequence of length 8, we can have a total of 7 dinucleotides, so with 16 possibilities, the dinucleotides at each of our 7 positions could be unique.
- Word size of 3: The maximum number of different words possible here is 6. For DNA generally, there is a total of 64 different trinucleotides ($4 \times 4 \times 4$). For a sequence of length 8, we can have a total of 6 trinucleotides, so with 64 possibilities, the trinucleotides at each of our 6 positions could be unique.

- **Word size of 4:** The maximum number of different words possible here is 5. For DNA generally, there is a total of 256 different dinucleotides ($4*4*4*4$). For a sequence of length 8, we can have a total of 5 quaternucleotides, so with 256 possibilities, the quaternucleotides at each of our 5 positions could be unique.

We then continue, using the logic above, to calculate a maximum possible number of words for a word size of 5 being 4, a maximum possible number of words for a word size of 6 being 3, and a maximum possible number of words for a word size of 7 being 2.

Now we can compute the complexity for this 7 nucleotide sequence by taking the number of different words we observe for each word size from 1 to 7 nucleotides and dividing them by the maximum possible number of words for each word size from 1 to 7. Here that gives us:

$$(4/4)(5/7)(5/6)(5/5)(4/4)(3/3)(2/2) = 0.595$$

As an extreme example of a sequence of low complexity, consider the 7 base sequence AAAAAAA. Here, we would get the complexity:

$$(1/4)(1/6)(1/5)(1/4)(1/3)(1/2)(1/1) = 0.000347$$

31.11 Structural Variant Caller for Long Reads

The Structural Variant Caller for Long Reads tool calls structural variants of length ≥ 35 from long reads. The input read mapping must have been produced by the Map Long Reads to Reference tool using *CLC Genomics Workbench* 24 or later.

The tool does not have specific handling for circular chromosomes. Circular chromosomes are treated as if they were linear. This means that variants near the origin may go undetected because the part of the read wrapping around the origin is ignored.

31.11.1 Structural Variant Caller for Long Reads settings

To run the Structural Variant Caller for Long Reads tool:

Tools | Resequencing Analysis  | Variant Detection  | Structural Variant Caller for Long Reads 

The tool accepts a single read mapping as input.

There are two calling modes, Germline and Somatic, which can be used in two applications, Whole genome sequencing and Targeted sequencing, as shown in figure 31.51.

The calling modes are:

- **Germline.** Recommended for diploid samples.
- **Somatic.** Recommended for calling on samples where variants may appear at low frequencies. When this is enabled:
 - Filtering of variants based on coverage and read mapping orientation may differ from germline mode depending on the application - see below for details.
 - Strict filtering of deletions and duplications that are longer than 50,000bp is disabled.

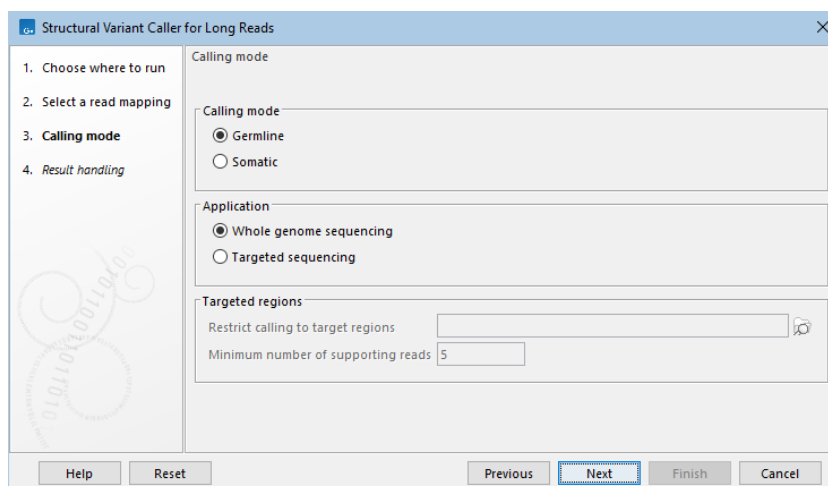


Figure 31.51: The calling mode and application options for Structural Variant Caller for Long Reads.

- Inversions and duplications are only reported if they are at least 500bp long.
- Variants with frequencies between 5% and 30% are filtered away if they are supported by alignments with many mismatches and indels compared to the reference.
- Variants with frequencies <5% are not reported.

The Targeted sequencing application requires two additional options to be set:

- **Restrict calling to target regions.** Enables selection of a feature track to limit calling to regions where reads are expected to map with high coverage. Variants will only be reported if they start or end within a target region, or at most 15bp away from a target region. In the case of breakend variants, only one breakend in a pair need be within the target region for both to be reported. This makes this setting suitable for the detection of unknown fusion partners using the FUDGE technique of [Stangl et al., 2020](#).

Several filters are applied to variants that depend on their coverage at upstream and downstream positions. If such a position lies outside a target region, then it is assumed that no coverage is expected at this position, and the filter is not applied.

In rare cases, the coverage of the variant itself cannot be determined, because the locations used to calculate it lie outside the target region. Such variants are reported as heterozygous with coverage '0'.

- **Minimum supporting reads.** Variants are filtered away if they have fewer countable reads than this. The 'countable' reads are those that are used by the variant detection tool when calling the variant and may be fewer in number than seen in the read mapping.

Targeted sequencing differs from Whole genome sequencing by how variants are filtered based on coverage. This is because coverage may vary along the target length and/or targets may have very different average coverages from each other:

- **Germline - Whole genome sequencing.** For each variant, the minimum number of supporting reads required for it to be reported is calculated based on its type (deletion, insertion etc.), the local coverage near the variant, and the average chromosome coverage.

- **Somatic - Whole genome sequencing.** Variants whose coverage changes by more than 10% when moving from upstream of the variant to the start position, or from the start position to the center of the variant, or from the center of the variant to the end position, or from the end position to downstream of the variant are filtered away.
- **Germline - Targeted sequencing** and **Somatic - Targeted sequencing.** Variants are filtered away if they have fewer supporting reads than specified by the 'Minimum supporting reads' parameter.

Targeted sequencing also differs from Whole genome sequencing by how variants are filtered based on read mapping orientation. This is because targeted protocols may generate reads in a particular orientation for each target.

- **Germline - Whole genome sequencing.** Breakend variants are filtered away if evidence for them is observed on reads mapping in only one orientation.
- **Somatic - Whole genome sequencing.** All variants other than long insertions are filtered away if evidence for them is observed on reads mapping in only one orientation.
- **Germline - Targeted sequencing** and **Somatic - Targeted sequencing.** Variants are never filtered away based on read mapping orientation.

31.11.2 Structural Variant Caller for Long Reads output

The tool has the following output options, as shown in figure 31.52:

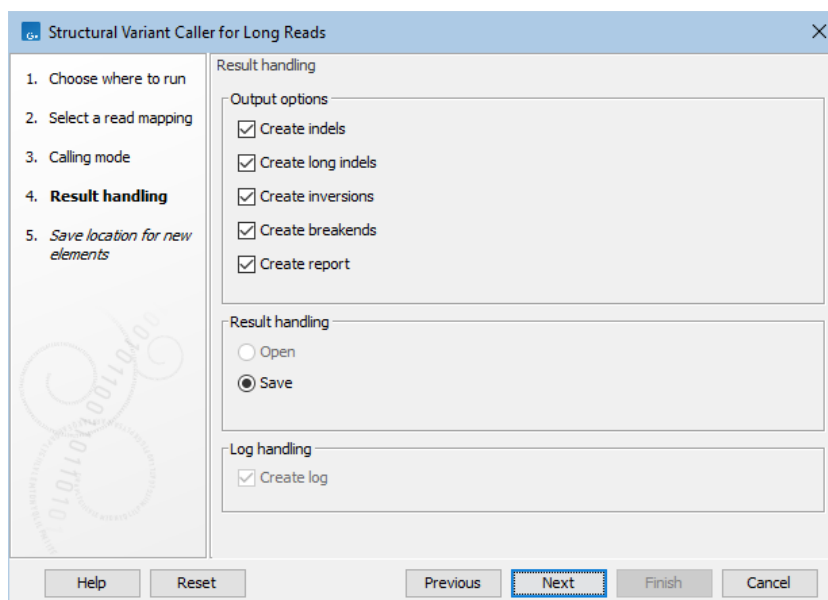






Figure 31.52: The output options for Structural Variant Caller for Long Reads.

- **Indels (Indels)** (PBB). A variant track with indels (deletions and insertions - including tandem duplications) that have lengths up to 100,000 bp.

- **Long indels (Indels long)** . An annotation track with long indels (those with lengths larger than 100,000 bp). The reason for putting indels larger than 100,000 bp in a separate annotation track, is that these variants have either long allele or reference entries in the variant track, which make them challenging to work with in the track viewer.
- **Inversions (Inv)** . An annotation track with inversions.
- **Breakends (Breakends)** . An annotation track with a row for each breakend in a translocation.
- **Report** . A report giving an overview over analyzed references and found structural variants.

All outputs (other than the Report) can be exported together to a single VCF file, see section 8.1.7. The VCF-exportable outputs contain the following annotations:

- **Chromosome.** The chromosome on which the variant is located.
- **Region.** The location of the variant.
- **Zygosity.** The zygosity of the variant called, as determined by the variant detection tool. This will be either 'Homozygous', where there is only one variant called at that position or 'Heterozygous' where more than one variant was called at that position.
- **Count.** The estimated count for the structural variant. In the case of insertions longer than 2500 bp, this is increased by an ad hoc procedure to account for reads that are not observed because they align within the insertion. If the coverage cannot be determined then this is reported as '0'. It is therefore advised to perform any desired filtering of variants on the 'Raw count' annotation rather than this annotation.
- **Coverage.** The estimated coverage for the structural variant. This is determined in different ways for different structural variant types. For insertions, this is the coverage at the location of the insertion; for duplications it is the average of the coverage at the two ends of the duplication; for inversions it is the average at positions upstream and downstream of the inversion; for deletions and breakends it is the average of the coverage at the start, end, and center of the variant. When taking averages, locations with no reads are ignored. In rare cases the coverage cannot be determined, because none of the locations used for calculating it have any reads. In these cases, the coverage will be reported as '0'.
- **Frequency.** The count divided by the coverage, reported as a percentage.
- **Average quality.** The average mapping quality of the alignments supporting the variant. The mapping quality is reported on the Phred scale, and describes the probability that the alignment is incorrect. The estimates are made by minimap2. A value of 30 means a 1 in 1000 chance of incorrect alignment. Very low average qualities are not seen, because the algorithm filters alignments with low mapping quality. Note that other variant calling tools, such as the Low Frequency Variant Detection tool, use this annotation to report the average quality scores of reads on the Phred scale rather than the average quality of alignments on the Phred scale.
- **Stdev position.** An estimate of the standard deviation of the position of the variant.

Additional annotations present on more than one output are:

- **Length.** The length of the variant. For deletions, it is the length of the deleted sequence, and for insertions and duplications it is the length of the inserted sequence. For inversions, it is the length of the inverted region.
- **Allele.** The inserted sequence. This is only reported for insertions and duplications.
- **Forward read count.** For inversions, the number of countable reads supporting the 5' side of the inversion on the reference. For other types of variant, the number of reads supporting the allele and mapping in the forward direction. The 'countable' reads are those that are used by the variant detection tool when calling the variant.
- **Reverse read count.** For inversions, the number of countable reads supporting the 3' side of the inversion on the reference. For other types of variant, the number of reads supporting the allele and mapping in the reverse direction. The 'countable' reads are those that are used by the variant detection tool when calling the variant.
- **Raw count.** The number of countable reads supporting the allele. The 'countable' reads are those that are used by the variant detection tool when calling the variant.
- **Detailed type.** This is only present for insertions and is set to "Insertion" if the variant describes a novel insertion, and "Duplication" if the insertion instead describes a duplication.
- **Repeat unit size.** Duplications are sometimes detected as insertions within a read. In these cases, their reported "Length" is estimated from these insertions. Because the reference sequence may be duplicated more than once, this can be larger than the duplicated region on the reference. This is an estimate of the size of one copy on the reference. It is equal to the "Length" if it cannot be estimated. In rare circumstances, the "Repeat unit size" may be estimated to exceed the "Length". This is usually caused when a repeat is inserted between two existing repeats of the same kind.
- **Stdev length.** An estimate of the standard deviation of the length of the variant.

Indels variant track

The indels track uses many of the standard variant annotations, see section [31.6.1](#).

Long indels variant track

This track contains insertions and deletions larger than 100,000 bp. It is often enriched for false positive calls. This is because deletions and duplications are called when a read maps to two disjoint locations. If these two locations are at either end of a chromosome, then a near-whole chromosome deletion (or duplication) will be called. In many cases, it is more likely that the read maps to two places because an insertion is present that shares homology with one of the locations.

It is sometimes possible to detect false positives. For example, if the sample is germline, and other structural variants are called within a long homozygous deletion, then it is likely that the long homozygous deletion is a false positive.

Inversions track

This track is often enriched for false positive calls. This is because an inversion may be called when a read maps to two disjoint locations on the same chromosome and in different orientations. If these two locations are at either end of a chromosome, then a near-whole chromosome inversion will be called. In many cases, it is more likely that the read maps to two places because an insertion is present that shares homology with one of the locations.

When coverage is high, it is often possible to detect false positives by requiring that there is support for both sides of the inversion. Reads supporting the 5' side of the inversion on the reference are counted as "forward" reads, and reads supporting the 3' side of the inversion on the reference are counted as "reverse" reads. Each variant reports these in "Forward read count" and "Reverse read count" annotations respectively.

Another class of false positives are inversions that start or end at the same location as an insertion. This is sometimes a signature of an inverted repeat.

Breakend track

The breakend track can be used to look for translocations and other complex rearrangements that involve more than one chromosome. The definition of a breakend that we use here closely follows that from the VCF specification. Please refer to Section 5.4 "Specifying complex rearrangements with breakends" of <https://samtools.github.io/hts-specs/VCFv4.4.pdf>. Specifically we support the cases shown in figures 1, 4, 5, and 7 of that section.

Annotations that are only present on the breakends output are:

- **Filter.** This is always PASS. It is necessary for VCF export.
- **Breakpoint type.** 5' if the breakend is on an earlier chromosome than its mate, 3' if it is on a later chromosome.
- **Type.** Either "donor" or "acceptor". Reads mapping to a "donor" breakend are mapped to the left of the breakend, disappear at the breakend location, and reappear on another chromosome. Reads mapping to an "acceptor" breakend appear at the breakend location and continue to the right of the location.
- **Fusion crossing reads.** The number of distinct reads that support the breakend. Note that this is not necessarily the number of reads that support the fusion, because the same breakend may take part in more than one fusion.
- **Fusion number.** Two breakends that share a fusion number describe one fusion. The read aligns up to one breakend on one chromosome, and then continues aligning at the other breakend on a different chromosome. If a breakend is involved in more than one fusion, it will appear more than once in the track, with a different fusion number for each fusion.

A simple reciprocal translocation involves 4 breakends: an acceptor and donor on each of the two chromosomes involved in the translocation. The 4 breakends will have different combinations of Name and Type, and two different Fusion numbers.

The easiest way to find translocations is to:

- Open the table view of the Breakend track and sort by **Fusion number** by clicking the column header.
- Look for two nearby fusion pairs in the sorted table with the same **Chromosome** and similar **Region**.
- Verify that each fusion pair consists of one donor and one acceptor type, and that each chromosome contains one donor and one acceptor type.

Note that the **Region** for a breakend is sometimes on the plus strand (e.g. 123456^123457) and sometimes on the minus strand (e.g. complement(123456^1234567)). There is no significance to the reported strand - it is used by the VCF exporter.

31.11.3 Structural Variant Caller for Long Reads algorithm

The tool is based on Sniffles2 v2.2. Results are therefore expected to be similar but not identical to those of Sniffles2 v2.2. For details on the algorithm, please refer to the Sniffles2 preprint <https://www.biorxiv.org/content/10.1101/2022.04.04.487055v2.full>.

The principal differences from Sniffles2 v2.2 are outlined below:

- Insertions contained within a primary alignment of a read are remapped to determine if they are duplications. This reduces the number of duplications that are erroneously reported as both an insertion and a duplication.
- Breakends that are close to an insertion of known length are not called when the longest read supporting the breakend is shorter than the shortest estimate of the insertion length. This is because it is likely that the inserted sequence is homologous to a region on another chromosome, and that the reads supporting the breakend are the reads that did not extend through the insertion.
- The full supplementary alignments for a read are always used, rather than the summary of the alignments provided by minimap2 in the SA SAM flag. Supplementary alignments describe the alignment of parts of a read that are not aligned in the primary alignment. This change is expected to give more precise structural variant locations and lengths in some cases.
- The consensus sequence of insertions is determined from all the reads supporting the insertion.
- Breakends are paired, and are only reported if the two locations supported by a read passing through the breakend are called. This leads to more interpretable breakends, but at the cost that rearrangements where only one breakend meets quality control cutoffs are not visible.
- A targeted mode is available.

Chapter 32

Resequencing

Contents

32.1 Variant filtering	913
32.1.1 Filter against Known Variants	913
32.1.2 Remove Marginal Variants	914
32.1.3 Remove Homozygous Reference Variants	915
32.1.4 Remove Variants Present in Control Reads	916
32.2 Variant annotation	917
32.2.1 Annotate from Known Variants	917
32.2.2 Annotate with Effect Scores	917
32.2.3 Annotate with Conservation Score	918
32.2.4 Annotate with Flanking Sequence	919
32.2.5 Annotate with Repeat and Homopolymer Information	919
32.3 Variants comparison	922
32.3.1 Identify Shared Variants	922
32.3.2 Identify Enriched Variants in Case vs Control Samples	923
32.3.3 Trio Analysis	925
32.4 Variant quality control	927
32.4.1 Create Variant Track Statistics Report	927
32.5 Functional consequences	929
32.5.1 Amino Acid Changes	929
32.5.2 Predict Splice Site Effect	936
32.5.3 GO Enrichment Analysis	937
32.5.4 Download 3D Protein Structure Database	939
32.5.5 Link Variants to 3D Protein Structure	940
32.6 Create Consensus Sequences from Variants	946

In the *CLC Genomics Workbench* *resequencing* is the overall category for applications comparing genetic variation of a sample to a reference sequence. This can be targeted resequencing of a single locus or whole genome sequencing. The overall workflow will typically involve read mapping, some sort of variant detection and interpretation of the variants.

This chapter describes the tools relevant for the resequencing workflows downstream from the actual read mapping which is described in chapter 30.

32.1 Variant filtering

In addition to the general filter for track tables, including the ability to create a new track from a selection (see section 27.3.2), there are a number of tools for general filtering of variants.

For functional filtering, see section 32.5.


32.1.1 Filter against Known Variants

The **Filter against Known Variants** tool filters experimental variants based on a known variant track to remove common variants.

Any variant track can be used as the "known variants" track. It may either be produced by the *CLC Genomics Workbench*, imported or downloaded from variant database resources like dbSNP, 1000 genomes, HapMap etc. (see section 7.2 and section 11.1).

To get started, go to:

Tools | Resequencing Analysis  | **Variant Filtering**  | **Filter against Known Variants** 

This opens a dialog where you can select a variant track  with experimental data that should be filtered.

Clicking **Next** will display the dialog shown in figure 32.1

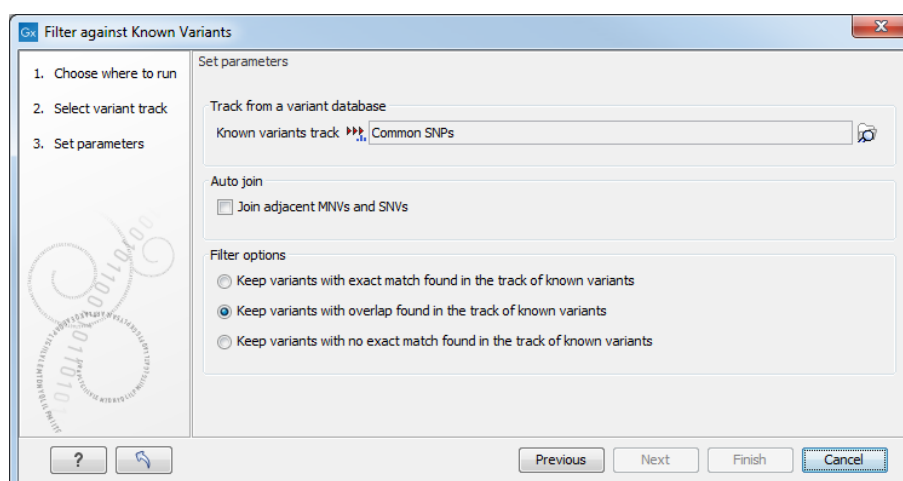



Figure 32.1: Specifying a variant track to filter against.

Select  one or more tracks of known variants to compare against. The tool will then compare each of the variants provided in the input track with the variants in the track of known variants. The output will be a variant track where the remaining variants will depend on the mode of filtering chosen:

- **Keep variants with exact match found in the track of known variants.** This will filter away all variants that are not found in the track of known variants. This mode can be useful for filtering against tracks with known disease-causing mutations, where the result will only include the variants that match the known mutations. The criteria for matching are simple: the variant position and allele both have to be identical in the input and the

known variants track (however, note the extra option for joining adjacent SNVs and MNVs described below). For each variant found, the result track will include information from the known variant. Please note that the exact match criterion can be too stringent, since the database variants need to be reported in the exact same way as in the sample. Some databases report adjacent indels and SNVs separately, even if they would be called as one replacement using the variant detection of *CLC Genomics Workbench*. In this case, we recommend using the overlap option instead and manually interpret the variants found.

- **Keep variants with overlap found in the track of known variants.** The first mode is based on exact matching of the variants. This means that if the allele is reported differently in the set of known variants, it will not be identified as a known variant. This is typically not the case with isolated SNVs, but for more complex variants it can be a problem. Instead of requiring a strict *match*, this mode will keep variants that *overlap* with a variant in the set of known variants. The result will therefore also include all variants that have an exact match in the track of known variants. This is thus a more conservative approach and will allow you to inspect the annotations on the variants instead of removing them when they do not match. For each variant, the result track will include information about overlapping or strictly matched variants to allow for more detailed exploration.
- **Keep variants with no exact match found in the track of known variants.** This mode can be used for filtering away common variants if they are not of interest. For example, you can download a variant track from 1000 genomes or dbSNP and use that for filtering away common variants. This mode is based on exact match.

Since many databases do not report a succession of SNVs as one MNV, it is not possible to directly compare variants called with *CLC Genomics Workbench* with these databases. In order to support filtering against these databases anyway, the option to **Join adjacent SNVs and MNVs** can be enabled. This means that an MNV in the experimental data will get an exact match, if a set of SNVs and MNVs in the database can be combined to provide the same allele.

Note! This assumes that SNVs and MNVs in the track of known variants represent the same allele, although there is no evidence for this in the track of known variants.

This tool will create a new track where common variants have been removed. The annotations that are left are marked in three different ways:

Exact match This means that the variant position and allele both have to be identical in the input and the known variants track (however, note the extra option for joining adjacent SNVs and MNVs described below).

Partial MNV match This applies to MNVs which can be annotated with partial matches if an SNV or a shorter MNV in the database has an allele sequence that is contained in the allele sequence of the annotated MNV.

Overlap This will report if the known variant track has an overlapping variant.

32.1.2 Remove Marginal Variants

Variant calling is always a balance between sensitivity and specificity. **Remove Marginal Variants** is designed for the removal of potential false positive variants. The tool can be configured

to remove variant calls with low frequency, a skewed forward-reverse reads balance, or those predominantly supported by low quality bases.

A new variant track is produced as output, leaving the original variant track as it was.

To run **Remove Marginal Variants**, go to:

Tools | Resequencing Analysis (📁) | **Variant Filtering** (📁) | **Remove Marginal Variants** (🔍)

This opens a dialog where you can select a variant track (📁).

Click on **Next** to move to the dialog where filtering thresholds can be set, as shown in figure 32.2.

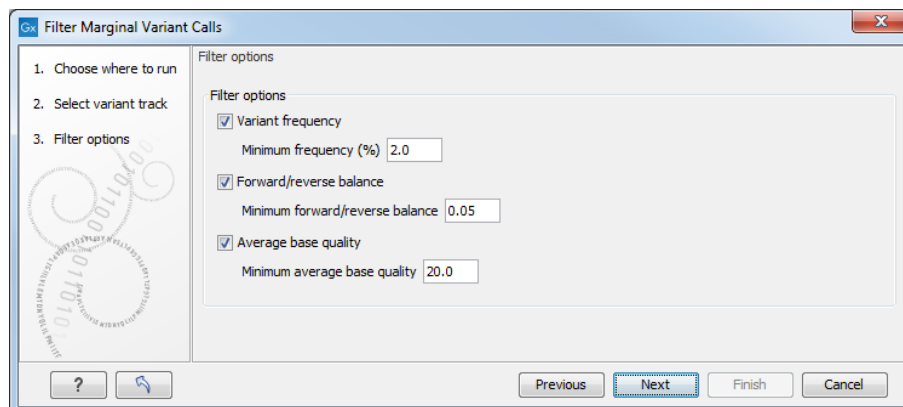


Figure 32.2: One or more thresholds can be configured, defining the basis for variant removal.

The following thresholds can be specified. All alleles are investigated separately.

- **Variant frequency.** The frequency filter will remove all variants having alleles with a frequency (= number of reads supporting the allele/number of all reads) lower than the given threshold.
- **Forward/reverse balance.** The forward/reverse balance filter will remove all variants having alleles with a forward/reverse balance of less than the given threshold.
- **Average base quality.** The average base quality filter will remove all variants having alleles with an average base quality of less than the given threshold.

If several thresholds are applied, just one needs to be fulfilled to discard the allele. For more information about how these values are calculated, please refer to section 31.6.1.

If all non-reference alleles at a position are removed, any remaining homozygous reference alleles will also be removed at that position.

A new variant track is produced by this tool, containing just the variants that exceeded the configured thresholds.

32.1.3 Remove Homozygous Reference Variants

This tool removes orphan variants from a variant track. Homozygous variants are described as lacking a variant allele, i.e., a corresponding non-reference variant with exactly the same start and end positions.

To start the tool, go to:

Tools | Resequencing Analysis (📁) | **Variant Filtering** (📁) | **Remove Homozygous Reference Variants** (🔍)

In the first dialog, select the variant track from which you would like to remove the homozygous variants, and click **Next** to decide whether to **Save** or **Open** the resulting variant track.

32.1.4 Remove Variants Present in Control Reads

Running the variant detection tool on a case and control sample separately and filtering away variants found in the control data set does not always give a satisfactory result as many variants in the control sample have not been called. This is often due to lack of read coverage in the corresponding regions or too stringent parameter settings. Therefore, instead of calling variants in the control sample, the Remove Variants Present in Control Reads tool can be used to remove variants found in both samples from the set of candidate variants identified in the case sample.

Tools | Resequencing Analysis (📁) | **Variant Filtering** (📁) | **Remove Variants Present in Control Reads** (📁)

The variant track from the case sample must be used as input. When clicking **Next**, you are asked to supply the number of reads in the control data set that should support the variant allele in order to include it as a match (see figure 32.3). All the variants where at least this number of control reads show the particular allele will be filtered away in the result track.

Please note that variants, which have no coverage in the mapped control reads will be reported too. You can identify them by looking for a 0 value in the column 'Control coverage'.

The following annotations will be added to each variant not found in the control data set:

Control count For each allele the number of reads supporting the allele.

Control coverage Read coverage in the control dataset for the position in which the allele has been identified in the case dataset.

Control frequency Percentage of reads supporting the allele in the control sample.

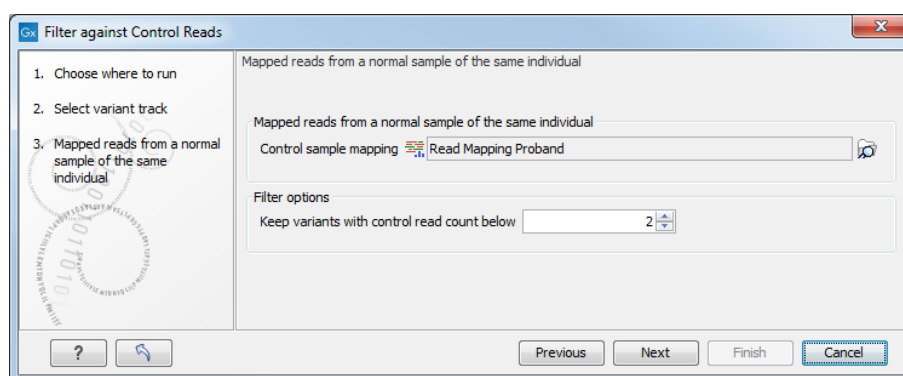


Figure 32.3: The control reads data set.

The filter option can be used to set a threshold for which variants should be kept. In the dialog shown in figure 32.3 the threshold is set at two. This means that if a variant is found in one or less of the control reads, it will be kept.

32.2 Variant annotation

Comparison with known variants from variant databases is a key concept when working with resequencing data. The *CLC Genomics Workbench* provides tools for annotating your experimental variants with information from known variants (e.g. adding information about phenotypes like cancer associated with a certain variant allele).

For functional annotation, see section 32.5.

For removing annotations from a variant track, see section 27.5.5.

32.2.1 Annotate from Known Variants

Annotate from Known Variants compares variants in an input track to known variants provided in another track. The tool outputs a new track containing the original variants with information added where matching known variants were identified.

To run the Annotate from Known Variants tool, go to:

Tools | Resequencing Analysis  | **Variant Annotation**  | **Annotate from Known Variants** 

There are three levels of matches between variants in the input track and the known variants track:

- **Exact match** Exact matches are those variants where the *position* and *allele* are identical in the input and known variants tracks. For exact matches, the output track includes information about that variant taken from the known variants track.
- **Partial MNV match** Partial MNV matches are those where the variant in the input track consists of an SNV or an MNV that has an allele sequence that is contained within the allele sequence of an MNV in the known variant track. These are not exact matches due to a difference in the variant lengths. Information from known variants are not transferred for partial matches.
- **Overlap** Where a variant in the input track overlaps a variant in the known variants track, but is not categorized as an exact match or a partial MNV match, it will be documented as an Overlap. Information from known variants are not transferred for overlap matches.

32.2.2 Annotate with Effect Scores

The Annotate with Effect Scores tool annotates SNV variants with precomputed effect scores. An effect score indicates the impact of a mutation on the gene or transcript. Synonymous mutations have low impact, whereas mutations introducing premature stop-codons or altering protein structures have high impact. Typically, scores are given in a range from 0 to 1, where 1 indicates completely neutral mutations and 0 indicates deleterious mutations, the range and interpretation does however depend on the score used.

To run the Annotate with Effect Scores tool, go to:

Tools | Resequencing Analysis  | **Variant Annotation**  | **Annotate with Effect Scores** 

Select the variant track (🔍) as input, when you click **Next** you will need to provide four tracks with effect scores (see figure 32.4). The track **Effect score A** contains the precomputed effect scores of SNVs where the new allele is A. The other three tracks are defined similarly.

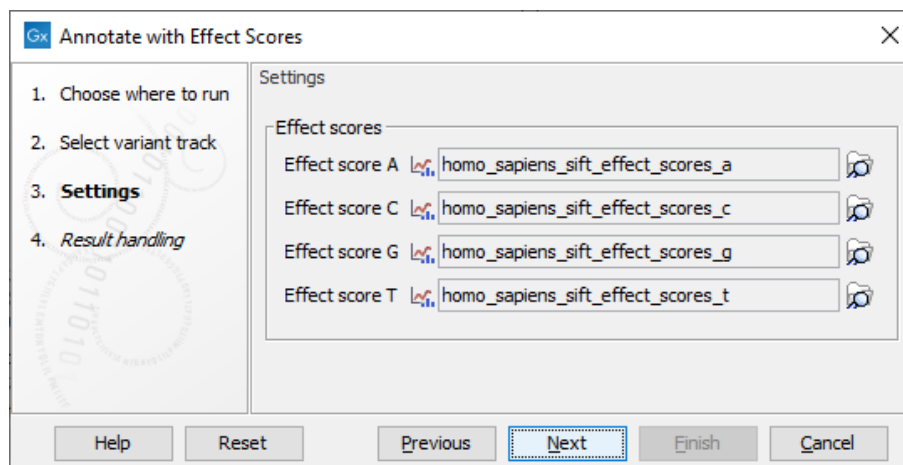


Figure 32.4: Select Effect Score tracks.

The tool outputs a variant track with an **Effect score** annotation added.

32.2.3 Annotate with Conservation Score

The possible functional consequence of a variant can be interrogated by comparing to a conservation score that tells how conserved this particular position is among a set of different species. The underlying line of thought is that conserved bases are functionally important - otherwise they would have been mutated during evolution. If a variant is found at a position that is otherwise well conserved, it is an indication that the variant is functionally important. Of course this is only a prediction, as non-conserved regions could have functional roles too.

Conservation scores can be computed by several tools e.g. PhyloP and PhastCons and can be downloaded as pre-computed scores from an whole genome alignment of different species from different sources. See how to find and import tracks with conservation scores in section 7.2.

Tools | Resequencing Analysis (🔍) | **Variant Annotation** (📁) | **Annotate with Conservation Scores** (🔍)

Select the variant track as input and when you click **Next** you will need to provide the track with conservation scores (see figure 32.5).

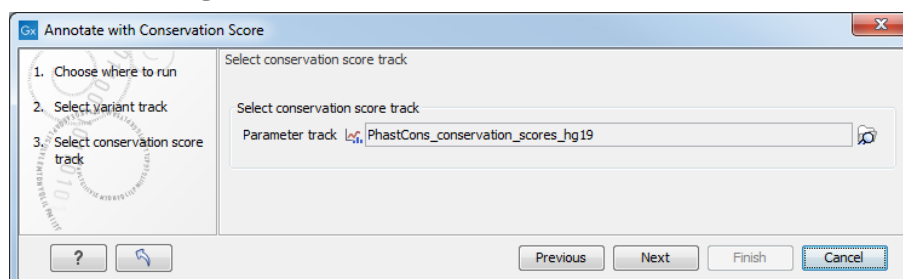


Figure 32.5: The conservation score track.

In the resulting track, all the variants will have quality scores annotated, and this can be used for sorting and filtering the track (see section 27.3.2).

32.2.4 Annotate with Flanking Sequence

In some situations, it is useful to see a variant in the context of the bases of the reference sequence. This information can be added using the **Annotate with Flanking Sequence** tool:

Tools | Resequencing Analysis (🔍) | **Variant Annotation** (📁) | **Annotate with Flanking Sequence** (T)

This opens a dialog where you can select a variant track (📊) to be annotated.

Clicking **Next** will display the dialog shown in figure 32.6

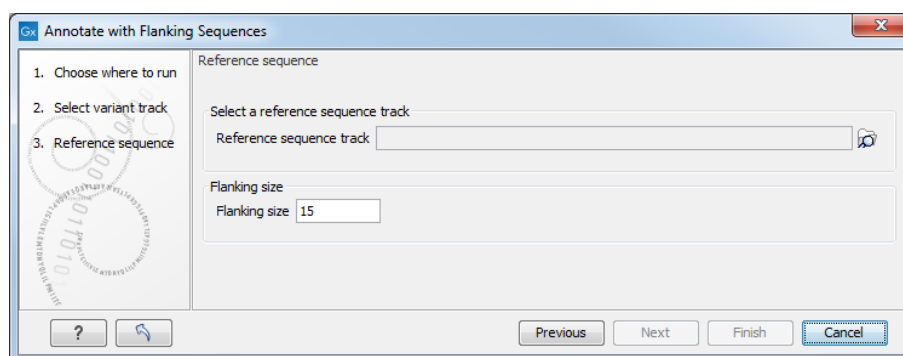


Figure 32.6: Specifying a reference sequence and the amount of flanking bases to include.

Select a sequence track that should be used for adding the flanking sequence, and specify how large the flanking region should be.

The result will be a new track with an additional column for the flanking sequence formatted like this: CGGCT [T] AGTCC with the base in square brackets being the variant allele.

32.2.5 Annotate with Repeat and Homopolymer Information

The Annotate with Repeat and Homopolymer Information tool annotates variants with repeat and homopolymer information, based on the variant itself and the genome sequence flanking it.

Homopolymers A variant is considered to be present in a homopolymer region if there are at least 4 consecutive repeats at that location, or for deletions, next to where the deletion occurred.

Repeats A variant is considered to be in a repeat region if:

- For a 2 bp variant, there are least 4 full copies at that location on the reference, or for deletions, next to where the deletion occurred.
- For a variant of 3bp or longer, there are at least 3 full copies at that location on the reference, or for deletions, next to where the deletion occurred.

The tool looks for homopolymers and repeats in both the reference and sample sequence. The sample sequence is the same as the reference sequence but also contains the variant that is being evaluated.


To determine if there is a homopolymer or repeat in a given reference and sample region, a hidden Markov model (HMM) is used. The HMM will allow for mismatches between single nucleotides if it determines that the mismatching part is still likely part of the homopolymer or

repeat. There is however a maximum number of mismatches allowed for repeating elements in the homopolymer/repeat sequence, where the comparison between repeating elements is handled differently than single nucleotides. For example, if we have the sequence TGGTGGTAA, then TGG is the repeating element and there is one mismatch between TGG and TAA. The maximum number of mismatches in repeating elements can be set when running the tool.

Note: This tool is designed for detecting shorter repeats and potential sequencing errors. Variants longer than 200 bp are therefore not evaluated and will always be marked as not being part of a homopolymer or repeat region.

To run **Annotate with Repeat and Homopolymer Information**, go to:

Tools | Resequencing Analysis  | **Variant Annotation**  | **Annotate with Repeat and Homopolymer Information** 

The tool takes variant tracks  as input.

In the first dialog select the variant track that should be annotated with repeat and homopolymer information.

In the second dialog, select the reference sequence and specify the maximum number of mismatches.

This tool outputs a report, containing a summary of the results, and a variant track with the following annotations added:

- **Homopolymer region** The value is "Yes" if the variant is in a homopolymer region in either the reference or sample sequence, and "No" if it is not.
- **Repeat region** The value is "Yes" if the variant is in a repeat region in either the reference or sample sequence, and "No" if it is not.
- **Reference homopolymer length** The homopolymer length is found by looking to the left and right of the variant position in the reference sequence. If two adjacent homopolymers are found then the longest one is chosen (see examples below).
- **Reference repeat sequence length** The repeat length is found by looking to the left and right of the variant position in the reference sequence. If two adjacent repeats are found then the longest one is chosen.
- **Sample homopolymer length** The homopolymer length is found by looking to the left and right of the variant position in the sample sequence. If two adjacent homopolymers are found then the longest one is chosen.
- **Effective sample homopolymer length** If the variant is within the homopolymer observed in the reference, then this value will be the same as the sample homopolymer length. Otherwise, this will be the length of the homopolymer that contains the variant in the sample sequence (see examples below).
- **Sample repeat sequence length** The repeat length is found by looking to the left and right of the variant position in the sample sequence. If two adjacent repeats are found then the longest one is chosen.
- **Reference repeat element** The k-mer that is repeated in the reference sequence (for example TCGA in TCGATCGATCGA).

- **Sample repeat element** The k-mer that is repeated in the sample sequence.

The five examples below illustrate how the reference, sample, and effective sample homopolymer lengths differ (the reference and sample repeat lengths follow the same principles). Note that the top sequence is the reference, below it is the sample sequence, while the variant position is marked by *:

1. Two adjacent homopolymers

```
TTTTTTTTTTTTTACCC
TTTTTTTTTTTTTCCCC
                *
```

Reference: 12

Sample: 12

Effective sample: 4

2. One adjacent homopolymer

```
TTTTTTTTTTTTTGATC
TTTTTTTTTTTTTCATC
                *
```

Reference: 12

Sample: 12

Effective sample: 1

3. Homopolymer in sample is shorter than reference

```
TTTTTTTTTTTTTGATC
TTTTTTTTTTTTTAGATC
                *
```

Reference: 12

Sample: 11

Effective sample: 11

4. Homopolymer in sample is longer than reference

```
TTTTTTTTTTTTTGCCC
TTTTTTTTTTTTTCCC
                *
```

Reference: 12

Sample: 13

Effective sample: 13

5. Variant in the middle of the sample homopolymer

```
TTTTTTTTTTTTT
TTATTTTTTTTTT
      *
```

Reference: 12

Sample: 9

Effective sample: 9

32.3 Variants comparison

The **Variant Comparison** folder in the Tools menu contains tools that can be used to compare experimental variants. The tool **Identify Shared Variants** is similar to the **Filter against Known Variants** found in the **Variant Filtering** folder. The main difference is how the tools are used. The **Filter against Known Variants** should be used when comparing experimental variants with variant databases, and the **Identify Shared Variants** when comparing experimental variants with other experimental variants.

32.3.1 Identify Shared Variants

This tool should be used if you are interested in finding common (frequent) variants in a group of samples. For example one use case could be that you have 50 unrelated individuals with the same disease and would like to identify variants that are present in at least 70% of all individuals. It can also be used to do an overall comparison between samples (a frequency threshold of 0% will report all alleles).

Tools | Resequencing Analysis (📁) | **Variant Comparison** (📁) | **Identify Shared Variants** (🔍)

This opens a dialog where you can select the variant tracks (📁) from the samples in the group.

Clicking **Next** will display the dialog shown in figure 32.7.

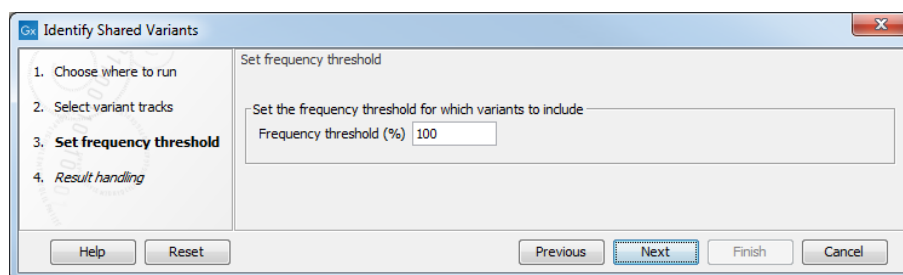


Figure 32.7: Frequency threshold.

The **Frequency threshold** is the percentage of samples that have this variant. Setting it to 70% means that at least 70% of the samples selected as input have to contain a given variant for it to be reported in the output.

The output of the analysis is a track with all the variants that passed the frequency thresholds and with additional reporting of:

- **Sample count.** Number of samples that have the variant
- **Total number of samples.** Total number of samples (this will be identical for all variants).
- **Sample frequency.** Frequency that is also used as a threshold (see figure 32.7).
- **Origin tracks.** Comma-separated list of the name of the tracks that contain the variant.
- **Homozygous frequency.** Percentage of samples passing the filter which have Zygosity annotation homozygous.
- **Heterozygous frequency.** Percentage of samples passing the filter which have zygosity annotation heterozygous.
- **Allele frequency.** Mean frequency of the allele in all input samples.

Note that this tool can be used for merging all variants from a number of variant tracks into one track by setting the frequency threshold to 0.

32.3.2 Identify Enriched Variants in Case vs Control Samples

This tool should be used if you have a case-control study. This could be individuals with a disease (case) and healthy individuals (control). The Identify Enriched Variants in Case vs Control Samples tool will identify variants that are significantly more common in the case samples than in the control samples. The Fisher exact test is applied on the number of occurrences of each allele of each variant in the case and the control data set. The alleles from each variant are considered separately, i.e. for an SNV with two alleles; a Fisher Exact test will be applied to each of the two. The test will also check whether an SNV in the case group is part of an MNV in the control group. Those with a low p-value are potential candidates for variants playing a role in the disease/phenotype. Please note that a low p-value can only be reached if the number of samples in the data set is high.

Tools | Resequencing Analysis  | **Variant Comparison**  | **Identify Enriched Variants in Case vs Control Samples** 

In the first step of the dialog, you select the case variant tracks (figure 32.8).

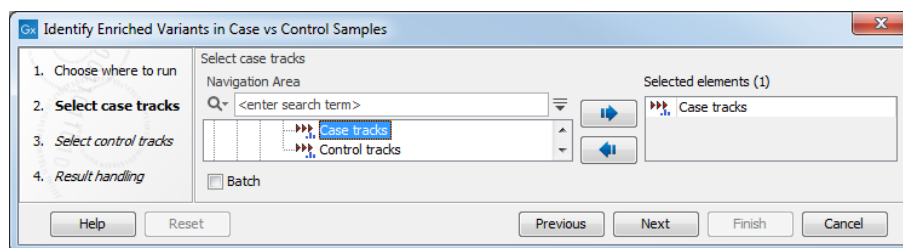


Figure 32.8: Select the case variant track.

Clicking **Next** shows the dialog in figure 32.9.

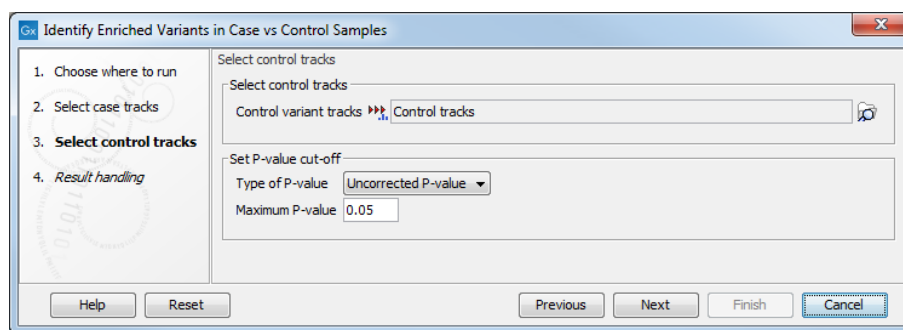


Figure 32.9: In this dialog you can select the control tracks, a p-value correction method, and specify the p-value threshold for the fisher exact test.

At the top, select the variant tracks from the control group. Furthermore, you must set a threshold for the p-value (default is 0.05); only variants having a p-value below this threshold will be reported. You can choose whether the threshold p-value refers to a corrected value for multiple tests (either Bonferroni Correction, or False Discovery Rate (FDR)), or an uncorrected p-value. A variant table is created as output (see figure 32.10), reporting only those variants with p-values lower than the threshold. All corrected and uncorrected p-values are shown here, so alternatively, variants with non-significant p-values can also be filtered out or more stringent thresholds can be applied at this stage, using the manual filtering options.

Chro...	Region	Type	Reference	Allele	Refere...	Length	Zygosity	Sample c...	Total num...	Sample fr...	P-value	Bonferroni	FDR p-val...
1	115247749-115247750	Insertion	-	A	No		1 Heterozygous	1	15	6.67	0.60	1.00	1.00
1	115247750	Deletion	A	-	No		1 Unknown	9	15	60.00	0.47	1.00	1.00
1	115248048	SNV	A	G	No		1 Heterozygous	1	15	6.67	0.85	1.00	1.00
1	115248053	SNV	C	T	No		1 Heterozygous	1	15	6.67	0.85	1.00	1.00
1	115248097	SNV	T	C	No		1 Heterozygous	1	15	6.67	0.60	1.00	1.00
1	115248251	Deletion	T	-	No		1 Heterozygous	1	10	10.00	1.00	1.00	1.00
1	115248260	SNV	G	A	No		1 Heterozygous	2	10	20.00	1.00	1.00	1.00
1	115248873	SNV	A	G	No		1 Heterozygous	4	15	26.67	0.54	1.00	1.00
3	178917005	SNV	A	G	No		1 Unknown	5	15	33.33	0.40	1.00	1.00
3	178917413	Deletion	A	-	No		1 Heterozygous	1	10	10.00	1.00	1.00	1.00

Figure 32.10: In the output table, you can view information about all significant variants, select which columns to view, and filter manually on certain criteria.

There are many other columns displaying information about the variants in the output table, such as the type, sequence, and length of the variant, its frequency and read count in case and control samples, and its overall zygosity. The zygosity information refers to **all** of the case samples; a label of 'homozygous' means the variant is homozygous in all case samples, a label of 'heterozygous' means the variant is heterozygous in all case samples, whereas a label of 'unknown' means it is heterozygous in some, and homozygous in others.

Overlapping variants: If two different types of variants occur in the same location, these are reported separately in the output table. This is particularly important, where SNPs occur in the same position as an MNV. Usually, multiple SNVs occurring alongside each other would simply be reported as one MNV, but if one SNV of the MNV is found in additional case samples by itself, it will be reported separately. For example, if an MNV of AAT -> GCA at position 1 occurs in five of the case samples, and the SNV at position 1 of A -> G, occurs in an *additional* 3 samples (so 8 samples in total), the output table will list the MNV and SNV information separately (however, the SNV will be shown as being present in only 3 samples, as this is the number in which it appears 'alone').

The test will also check whether an SNV in the case group is part of an MNV in the control group.

32.3.3 Trio Analysis

This tool should be used if you have a trio study with one child and its parents. It should be mainly used for investigating differences in the child in comparison to its parents.

To start the Trio analysis:

Tools | Resequencing Analysis (📁) | **Variant Comparison** (📁) | **Trio Analysis** (🔍)

In the first step of the dialog, select the variant track of the child. Clicking **Next** shows the dialog in figure 32.11.

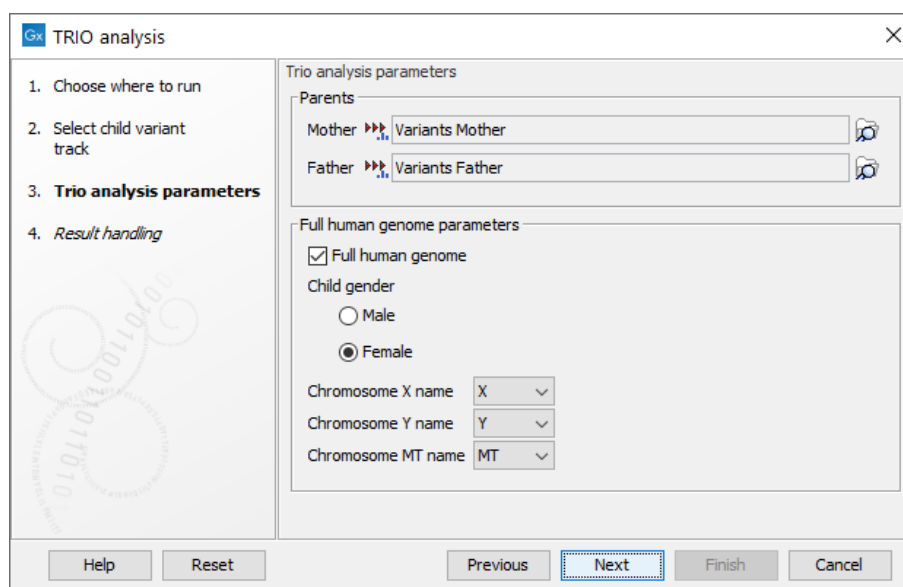


Figure 32.11: Selecting variant tracks of the parents.

Click on the folder (📁) to select the two variant tracks for the mother and the father. In case you have a human TRIO, please specify if the child is male or female and how the X, Y chromosomes as well as the mitochondrion are named in the genome track. These parameters are important in order to apply specific inheritance rules to these chromosomes.

Click **Next** and **Finish**.

The output is a variant track showing all variants detected in the child. For each variant in the child, it is reported whether the variant is inherited from the father, mother, both, either or is a de novo mutation. This information can be found in the tooltip for each variant or by switching to the table view (see the column labeled "Inheritance") (figure 32.12).

In cases where both parents are heterozygous with respect to a variant allele, and the child has the same phenotype as the parents, it is unclear which allele was inherited from which parent. Such mutations are described as 'Inherited from either parent'.

In cases where both parents are homozygous with respect to a variant allele, and the child has the same phenotype as the parents, it is also unclear which allele was inherited from which parent. Such mutations are described as 'Inherited from both'.

In cases where both parents are heterozygous and the child homozygous for the variant, the child has inherited a variant from both parents. In such cases the tool will also check for a potential recessive mutation. Recessive mutations are present in a heterozygous state in each of the

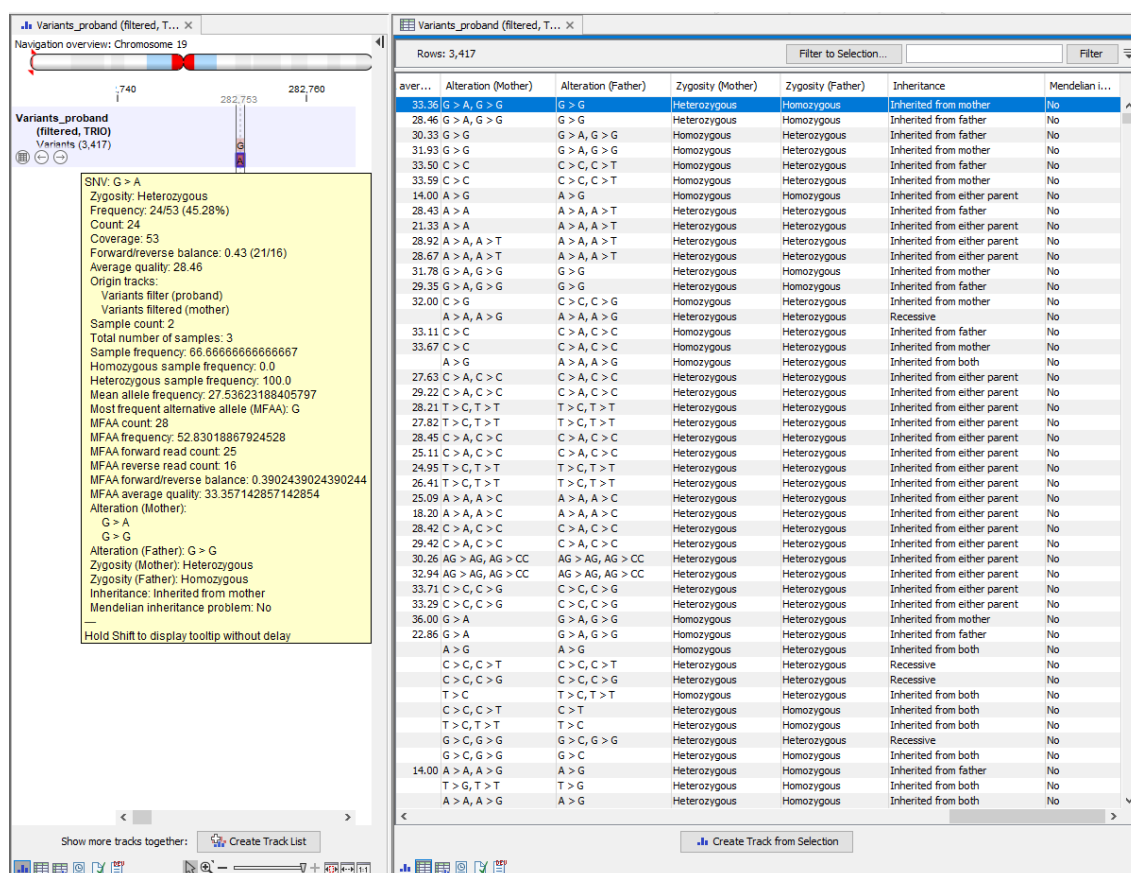


Figure 32.12: Output from Trio Analysis showing the variants found in the child in track and table format.

parents, but are homozygous in the child. To investigate potential disease relevant variants, recessive variants and de novo variants are the most interesting (in case the parents are not affected). The tool will also add information about the genotype (homozygote or heterozygote) in all samples.

If child or parent variants have a zygosity that is inconsistent with the number of alleles (e.g. heterozygous but one allele), then the mutation is described with 'Inconsistent zygosity'. Similarly, if the zygosity of child or parent variants are unknown, then the mutation will be described with 'Unknown zygosity'.

For humans, special rules apply for chromosome X (in male children) and chromosome Y, as well as the mitochondrion, as these are haploid and always inherited from the same parent. Heterozygous variants in the child that do not follow mendelian inheritance patterns will be marked in the result.

Here is an example where the trio analysis is performed with a boy:

The boy has a position on the Y chromosome that is heterozygous for C/T. The heterozygous C is not present in neither the mother or father, but the T is present in the father. In this case the inheritance result for the T variant will be: 'Inherited from the father', and for the C variant 'de novo'. However, both variants will also be marked with 'Yes' in the column 'Mendelian inheritance problem' because of this aberrant situation. In case the child is female, all variants on the Y chromosome will be marked in the same way.

The following annotations will be added to the resulting child track:

- **Zygosity.** Zygosity in the child as reported from the variant detection tool. Can be either homozygote or heterozygote.
- **Zygosity (Name of parent track 1).** Zygosity in the corresponding parent (e.g. father) as reported from the variant detection tool. Can be either homozygote or heterozygote.
- **Allele variant (Name of parent track 1).** Alleles called in the corresponding parent (e.g. father).
- **Zygosity (Name of parent track 2).** Zygosity in the corresponding parent (e.g. mother) as reported from the variant detection tool. Can be either homozygote or heterozygote.
- **Allele variant (Name of parent track 2).** Alleles called in the corresponding parent (e.g. mother).
- **Inheritance.** Inheritance status. Can be one of the following values, or a combination of these: 'De novo', 'Recessive', 'Inherited from both', 'Inherited from either', 'Inherited from (Name of parent track)', 'Inconsistent zygosity', 'Unknown zygosity'. For example, a proband homozygous for a variant that only one parent (the mother in this case) presents will result in an inheritance described as "De novo, inherited from mother".
- **Mendelian inheritance problem.** Variants not following the mendelian inheritance pattern are marked here with 'Yes'.

Note! If the variant at this position cannot be found in either of the parents, the zygosity status of the parent where the variant has not been found is unknown, and the allele variant column will be left empty.

32.4 Variant quality control

32.4.1 Create Variant Track Statistics Report

The tool Create Variant Track Statistics Report takes a variant track as input and reports statistics on variants classified by types. When the track contains the appropriate annotations, the report will also include sections on biological consequences and/or splice site effect.

To run the tool, go to:

Tools | Resequencing Analysis  | **Variant Quality Control**  | **Create Variant Track Statistics Report** 

In the first dialog, select the variant track you want to get statistics for. In the second dialog you can optionally specify a second variant track, that is a filtered version of the input track. The tool will check that all variants present in the filtered track are also present in the input track. If it is the case, the report will give statistics "before filtering" (based on the input track) and "after filtering" (based on the optional filtered track). When the compatibility check fails, the filtered track will be ignored.

Variant types section This section summarizes the count of non-reference variants of different types: SNV, MNV, Insertion, Deletion, and Replacement. Figure 32.13 shows the content of the section when only an input track is provided (left), and when both input and filtered tracks are provided (right).

1 Variant types		1 Variant types		
			Before filtering	After filtering
SNV	874,913	SNV	874,913	226,417
MNV	25,568	MNV	25,568	5,604
Insertion	79,281	Insertion	79,281	30,638
Deletion	122,334	Deletion	122,334	37,759
Replacement	5,365	Replacement	5,365	839
Total (non-reference)	1,107,461	Total (non-reference)	1,107,461	301,257

Figure 32.13: Variant types sections without (left) and with (right) a filtered track.

Amino acid changes section This section summarizes the count of non-reference variants based on whether they are situated outside exons (so without any effect on amino acid change), and when exonic, whether the change is synonymous or not. To be present in the report, this section requires previous annotation of the variant track(s) with the Amino Acid Changes tool (see section 32.5.1). Figure 32.14 shows the content of this section when a filtered track is provided. The example to the right shows what happens when the filtered track is missing annotations (statistics are then reported as Not Available).

2 Amino acid changes			2 Amino acid changes		
	Before filtering	After filtering		Before filtering	After filtering
Non-exonic	956,961	283,268	Non-exonic	956,961	N/A
Synonymous	39,809	14,891	Synonymous	39,809	N/A
Non-synonymous	110,691	23,098	Non-synonymous	110,691	N/A
Total (non-reference)	1,107,461	301,257	Total (non-reference)	1,107,461	N/A

Figure 32.14: Amino acid changes sections with a filtered track that contains annotations (left), and with a filtered track missing relevant annotations (right).

Splice site effect section This section summarizes the effect on splice sites produced by variants: Possible splice site disruption, and No splice site disruption. It requires previous annotation of the variant track(s) with the Predict Splice Site Effect tool (see section 32.5.2). Figure 32.15 shows the content of this section when a filtered track was provided. The example to the right shows what happens when the input track misses the relevant annotations (statistics are then reported as Not Available). Note that the Predict Splice Site Effect tool only annotates variants that produce a possible splice site disruption. It is then possible that when no such variant is found, the annotated track is devoid of annotations, and the report section of the Create Variant Track Statistics Report tool will resemble the one obtained from a track that has not been annotated at all.

3 Splice site effect			3 Splice site effect		
	Before filtering	After filtering		Before filtering	After filtering
Possible splice site disruption	9,051	1,572	Possible splice site disruption	N/A	1,572
No splice site disruption	1,098,410	299,685	No splice site disruption	N/A	299,685
Total (non-reference)	1,107,461	301,257	Total (non-reference)	N/A	301,257

Figure 32.15: Splice site effect sections with a filter track that contains annotations (left), and with an input track missing relevant annotations (right).

32.5 Functional consequences

The tools for working with functional consequences all take a variant track as input and will predict or classify the functional impact of the variant.

32.5.1 Amino Acid Changes

This tool annotates variants with amino acid changes and creates a track for visual inspection of the amino acid changes. It takes a variant track as input and also requires a track with coding regions and a reference sequence.

To add information about amino acid changes to a variant track:

Tools | Resequencing Analysis (📁) | **Functional Consequences** (📁) | **Amino Acid Changes** (🔍)

If you are connected to a server, the first wizard step will ask you where you would like to run the analysis. Next, you must provide the variant track to be annotated with amino acid changes (see figure 32.16).

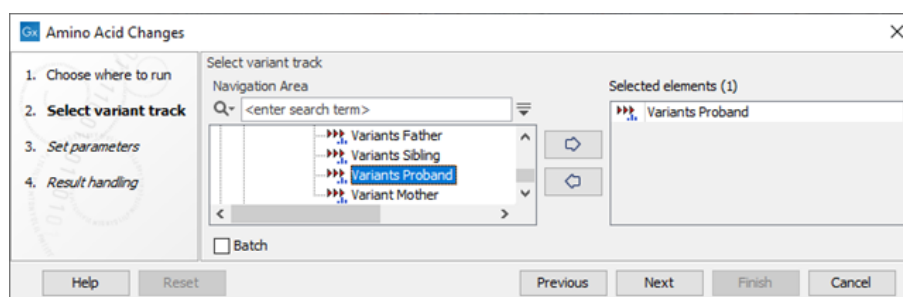


Figure 32.16: The Amino Acid Changes annotation tool takes variant tracks as input.

Click **Next** to go to the next wizard step (figure 32.17).

- **Select CDS track.** The CDS track is used to determine the reading frame and exon location to be used for translation. If you do not already have CDS, mRNA, and sequence tracks in the Workbench, you can download it with the Reference Data Manager found in the upper right corner of the Workbench.
- **Select mRNA track** (optional). The mRNA track is used to determine whether the variant is inside or outside the region covered by the transcript. Without an mRNA track, variants found outside the CDS region will not be annotated. When specifying an mRNA track, the tool will annotate variants that are located in the mRNA, but also outside the region covering the coding sequence in cases where such variants have been detected.
- **Use transcript priorities:** Check this option if you have provided an mRNA track that includes a "Priority" column, i.e. an integer value where "1" is higher priority than "2". When adding c. and p. annotations:
 1. Transcripts with changes in exons are preferred, then transcripts with changes in gene flanking regions, and finally transcripts with changes in introns. This means that, for example, a priority "2" transcript with exon changes is preferred over a priority "1" transcript with intron changes.

Amino Acid Changes

1. Choose where to run
2. Select variant track
3. **Set parameters**
4. Result handling

Set parameters

References

CDS track Homo_sapiens_ensembl_v87_hg19_CDS

mRNA track Homo_sapiens_ensembl_v87_hg19_mRNA

☐ Use transcript priorities

Sequence track Homo_sapiens_sequence_hg19

Variant location

☐ Move variants from VCF location to HGVS location

Flanking

Include upstream flanking positions

Include downstream flanking positions

Filtering and annotation

☐ Filter away synonymous variants

☒ Filter away CDS regions with no variants

☐ Use one letter codon code

Genetic code

Figure 32.17: Select CDS, mRNA, and sequence track and choose whether or not you would like to filter away synonymous variants.

2. If there are several transcripts with exon changes, for example, then only the annotation from the highest priority transcript intersecting with the variant will be added.
3. In cases where two or more genes overlap a variant, the highest priority transcript(s) will be reported from each gene.
4. Transcripts without any priority are ignored.

Note that a track with prioritized transcripts can be generated by modifying a gtf/gff file to add a "Priority" column.

- **Select sequence track.**

- **Variant location.** In VCF standard, variants with ambiguous positions are left-aligned, while HGVS standard places ambiguous variants most 3' relative to the transcript annotation. Checking the option **Move variants from VCF location to HGVS location** will output a track where ambiguous variants will be located following the HGVS standard, even when it moves the variant accross intron/exon boundaries and flanking regions. This option is recommended when comparing variants with databases following the HGVS standard.

This option does not affect the amino acid annotations added by the tool, as they always comply with the HGVS standard. Do, therefore, note that when "Move variants from VCF location to HGVS location" is unticked, variants with ambiguous positions will have the VCF standard position as the variant position, but the HGVS standard position in the annotation.

Also note that enabling this option may double some variants, for example in cases where a variant is overlapped by two genes - one on each strand - or overlapped by one gene and the flanking region of another on the other strand. Duplicating the variant ensures that the output contains a correctly positioned variant for each gene.

There are two columns in the variant track that can be used to identify variants that have been duplicated:

- The column **shift-original-id** is added when the option **Move variants from VCF location to HGVS location** is checked. The column contains an id for each original variant. If a variant has been duplicated, each of the resulting variants will have the same id.
- The column **Start by 3' rule** is always added to variant tracks. The column lists 3' variant positions for all variants. If two values are present, the variant will be duplicated if moved to HGVS location.

Note that when variants are duplicated, the variant track may no longer describe the sample genome accurately, and this option is therefore not recommended if the track will be exported to VCF or uploaded to QCI Interpret.

- When duplicated variants that do not overlap are exported to VCF, both will be included.
- When duplicated variants that overlap are exported to VCF:
 - * Both will be included when the option "Enforce ploidy" is not enabled.
 - * When "Enforce ploidy" is enabled, only the number of specified alleles will be included in the VCF. If the reference allele is more frequent than the variant, and maximum ploidy is two, only reference alleles will be included in the VCF.
- **Flanking.** It is possible to add c. annotations (HGVS DNA-level) to upstream and downstream flanking positions if they are within a certain distance from the transcript boundaries. The distance can be configured but the default distances are set to 5 kb upstream and 3 kb downstream.
- **Filtering and annotation.**
 - **Filter away synonymous variants** removes variants that do not cause any change to the encoded amino acids from the **variant track** output.
 - **Filter away CDS regions with no variants** removes CDS regions that have no variants from the **amino acid track** output.
 - **Use one letter codon code** gives one letter amino acid codes in the columns 'Amino acid change' and 'Amino acid change in the longest transcript' in the **variant track** output. When this option is not checked, three letter amino acid codes are used.
 - **Genetic code** is the code that is used for amino acid translation (see <https://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi>). The default option is "1 standard", the vertebrate standard code. If relevant, you can use the drop-down list to change to the genetic code that applies to your organism.

Click **Next**, choose whether you would like to **Open** or **Save** the results and click on the button labeled **Finish**.

Two types of outputs are generated:

1. **A variant track** that has been annotated with the amino acid changes. The added information can be accessed via the tooltips in the variant track or in the extra columns that have been added to the variant table. The extra columns provide information about

the amino acid changes (see <https://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi>). The variant track opens in track view and the table view can be accessed by clicking on the table icon found in the lower left corner of the **View Area**.

2. **An amino acid track** that displays a graphical presentation of the amino acid changes. The track is based on the CDS track and in addition to the amino acid sequence of the coding sequence, all amino acids that have been affected by variants are shown as individual amino acids below the amino acid track. Changes causing a frameshift are symbolized with two arrow heads, and variants causing premature stop are marked with an asterisk. An example is shown in figure 32.18. The information on the individual amino acids is displayed when the box is wide enough (three bases). This is typically the case – if not, the side panel settings can be adjusted accordingly (e.g by decreasing the "hide insertions below (%)" value). The information is always displayed in the tooltip on the box.

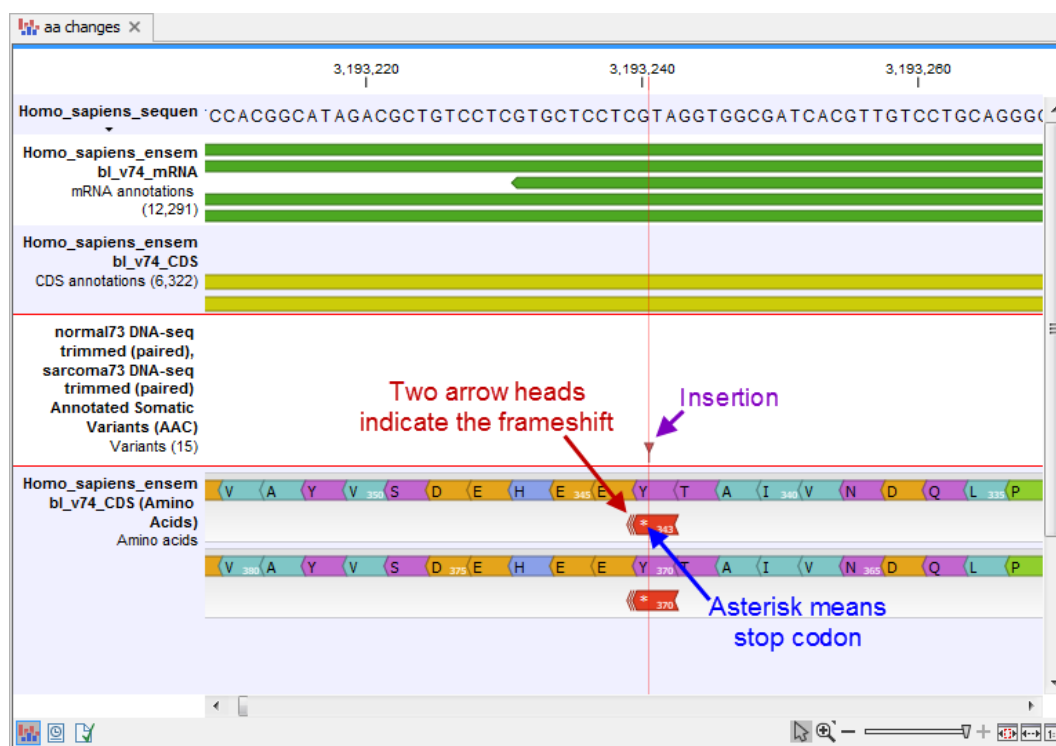


Figure 32.18: The variant track and the amino acid track is here presented together with the reference sequence and the mRNA and CDS tracks. An insertion (purple arrow) has caused a frameshift (red arrow) that has changed an alanine to a stop codon (blue arrow).

For each variant in the input track, the following information is added:

- **Coding region change.** This describes the relative position on the coding DNA level, using the nomenclature proposed at <https://hgvs-nomenclature.org/stable/>. Variants outside exons and in the untranslated regions of the transcript will also be annotated with the distance to the nearest exon. E.g. "c.-4A>C" describes a SNV four bases upstream of the start codon, while "c.*4A>C" describes a SNV four bases downstream of the stop codon.
- **Amino acid change.** This describes the change on the protein level. For example, single amino-acid changes caused by SNVs are listed as p.Gly261Cys, denoting that

in the protein sequence (hence the "p.") the Glycine at position 261 is changed into Cysteine. Frame-shifts caused by nucleotide insertions and deletions are listed with the extension *fs*, for example p.Pro244fs denoting a frameshift at position 244 coding for Proline. For further details about HGVS nomenclature as relates to proteins, please refer to <https://hgvs-nomenclature.org/stable/>.

- **Coding region change in longest transcript.** When there are many transcript variants for a gene, the coding region change for all transcripts are listed in the "Coding region change" column. For quick reference, the longest transcript is often used, and there is a special column only listing the coding region change for the longest transcript.
- **Amino acid change in longest transcript.** This is similar to the above, just on the protein level.
- **Other variants within codon.** If there are other variants within the same codon, this column will have a "Yes". In this case, it should be manually investigated whether the two variants are linked by reads.
- **Non-synonymous.** Will have a "Yes" if the variant is non-synonymous at the protein level for any transcript. If the filter "Filter synonymous" was applied, this column will only contain entries labeled "Yes". A hyphen, "-", indicates the variant was present outside of a coding region.

Note that variants located at the border of exons are considered intronic (i.e. located between the last intronic and first exonic base or between the last exonic and first intronic base). Amino acid changes will therefore not be determined for these variants.

An example of the output is given in figure 32.19.

The top track view displays a track list with the reference sequence, mRNA, CDS, variant, and amino acid tracks. The lower table view is the variant table that has been opened from the track list by double-clicking on the variant track name found in the left-hand side of the **View Area**. When opening the variant table in split view from the track list, the table and the variant track are linked.

An example illustrating a situation where different variants affect the same codon is shown in figure 32.20.

In this example three single nucleotide deletions are shown along with the resulting amino acid changes based on scenarios where only one deletion is present at the time. The first affected amino acid is shown for each of the three deletions. As the first deletion affect the encoded amino acid, this amino acid change is shown with a four nucleotide long arrow (that includes the deletion). The other two deletions do not affect the encoded amino acid as the frameshift was "synonymous" at the position encoded by the codon where the deletion was introduced. The effect is first seen at the next amino acid position (763 and 764, respectively), which does not contain a deletion, and therefore is illustrated with a three nucleotide long arrow.

The hash symbol (#) on the changed amino acids symbolize that more than one variant can be present in the region encoding this specific amino acid. The simultaneous presence of multiple variants within the same codon is not predicted by the amino acid changes tool. Manual inspection of the reads is required to be able to detect multiple variants within one codon.

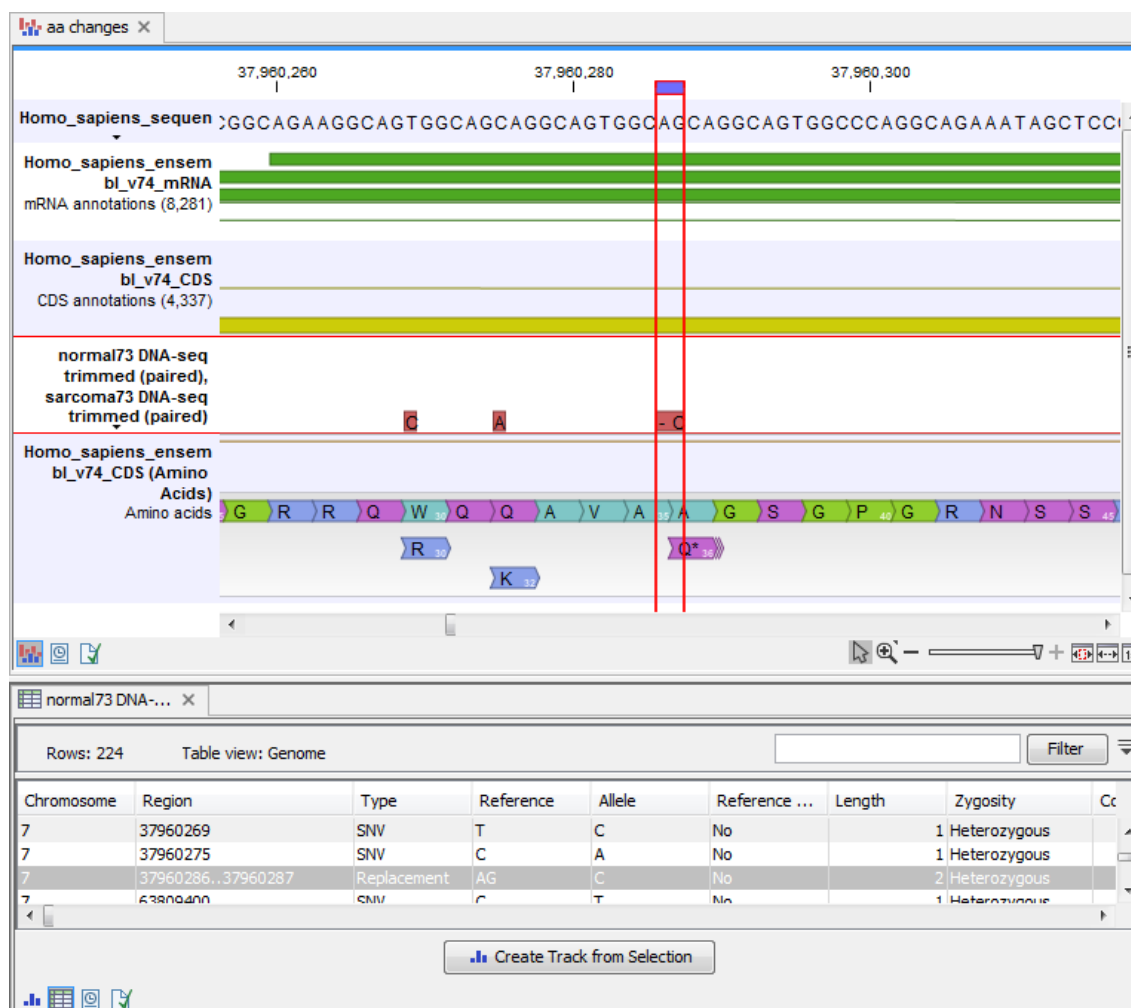


Figure 32.19: The resulting amino acid changes in track and table views. When the variant table has been opened by double-clicking on the text found in the left side of the View Area, the variant table and the variant track are linked. When clicking on an entry in the table, this position will be brought into focus in the variant track.

Known limitations

- When two genes overlap and an insertion in the form of a duplication occurs, this duplication will be labeled as an insertion.
- The Amino Acid Changes tool will not perform flanking checks for exons/CDS that wrap around the chromosome in a circular chromosome.
- For some transcripts, there is an error in the underlying reference sequence, leading to a difference between the reference sequence and the transcript sequence as provided in for example RefSeq. The Amino Acid Changes tool does not take reference errors into account when calculating amino acid changes. The results can therefore differ from software that uses the transcript sequence for calculating amino acid changes, such as QCI Interpret.

The amino acid track The colors of the amino acids in the amino acid track can be changed in the **Side Panel** under **Track layout** and "Amino acids track" (see figure 32.21).

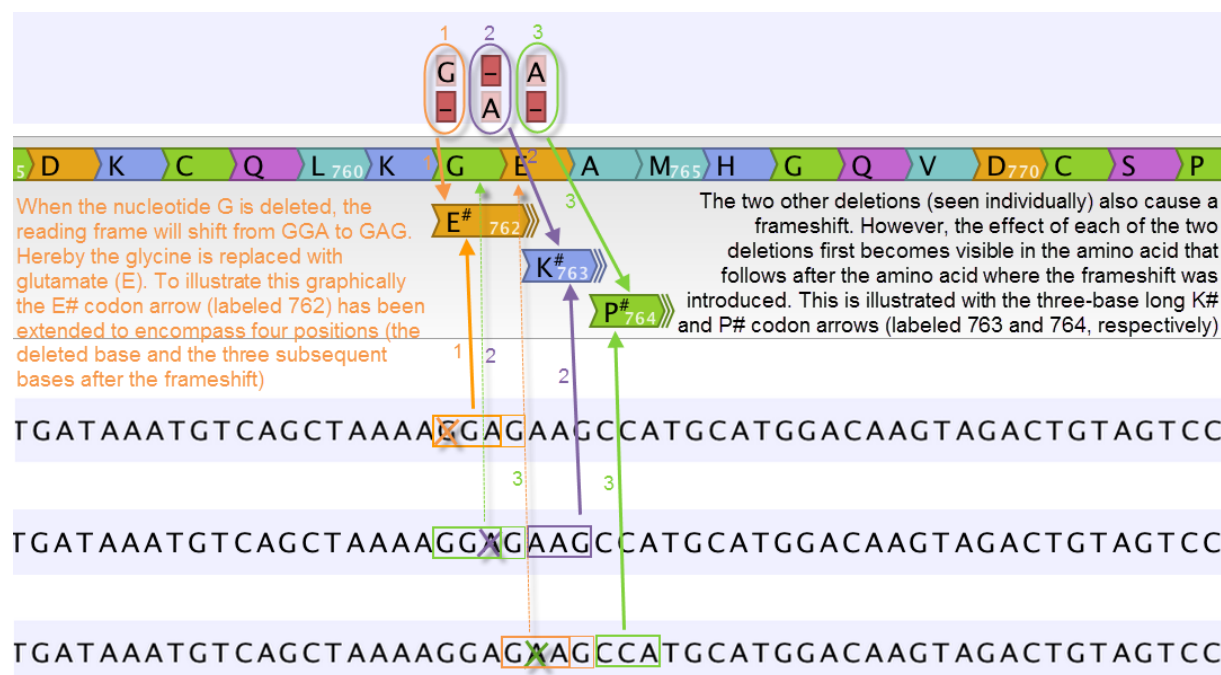


Figure 32.20: Amino acids encoded from codons that potentially could have been affected by more than one variant are marked with a hash symbol (#) as the graphically presented amino acid changes always only include a single variant (a SNV, MNV, insertion, or deletion). Shown here are three different variants, present only one at the time, and the consequences of the three individual deletions. In cases where the deletion is found in a codon that is affected with an amino acid change, the arrow also include the deletion (situation 1) in the two other scenarios, the codon containing the deletion is changed to a codon that encodes the same amino acid, and the effect is therefore not seen until in the subsequent amino acid.

Four different color schemes are available under "Amino acid colors":

- **Gray** All amino acids are shown in gray.
- **Group** Colors the amino acids in groups by the following properties:
 - **Purple** neutral, polar
 - **Turquoise** neutral, nonpolar
 - **Orange** acidic, polar
 - **Blue** basic ,polar
 - **Bright green** other (functional properties)
- **Polarity** Colors the amino acids according to the following categories:
 - **Green** neutral, polar
 - **Black** neutral, nonpolar
 - **Red** acidic, polar
 - **Blue** basic ,polar
- **Rasmol** Colors the amino acids according to the Rasmol color scheme (see <http://www.openrasmol.org/doc/rasmol.html>).

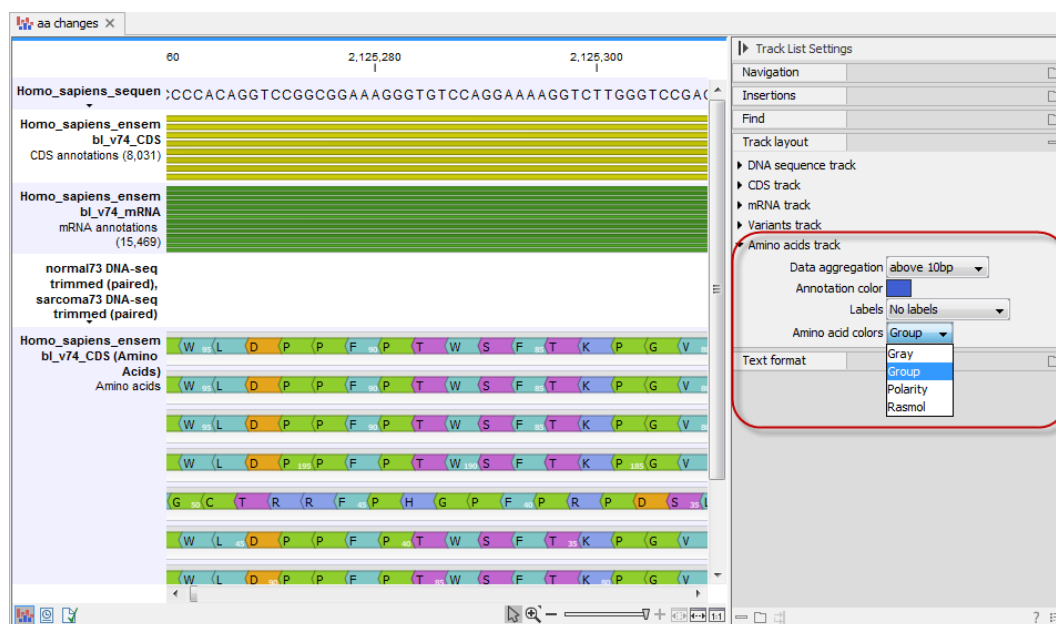


Figure 32.21: The colors of the amino acids can be changed in the Side Panel under "Amino acids track".

32.5.2 Predict Splice Site Effect

This tool will analyze a variant track to determine whether the variants fall within potential splice sites. First select your variant track (figure 32.22) followed by a transcript track (see figure 32.23). A variants that falls within a certain distance of an intron-exon boundary will be annotated as a possible splice site disruption (see figure 32.24). The distance is determined by the parameter **Splice site window size**. In the filter group you can choose to exclude all variants that do not fall within a splice site.

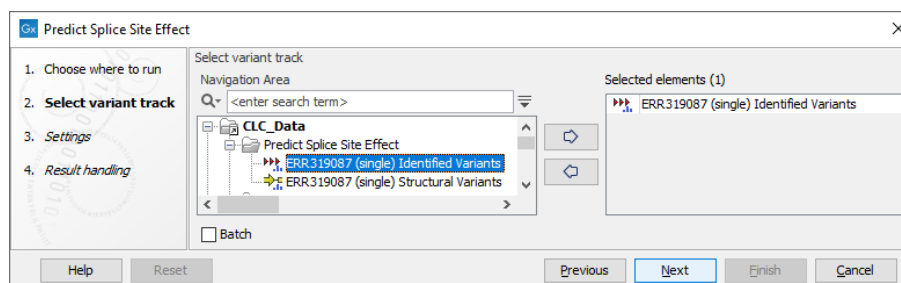


Figure 32.22: Variant track selection.

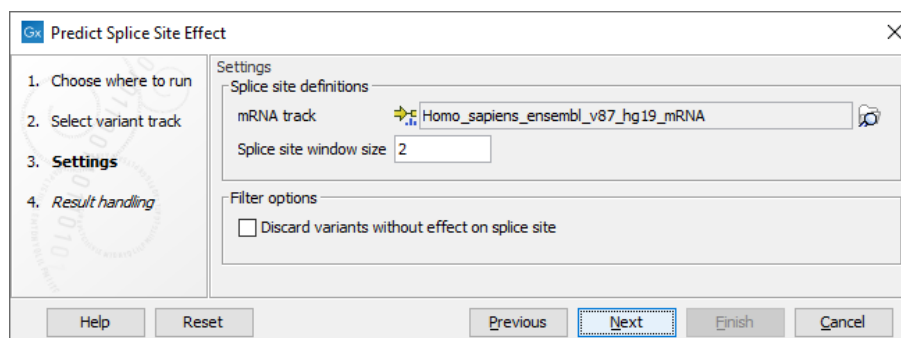


Figure 32.23: Transcript track selection.

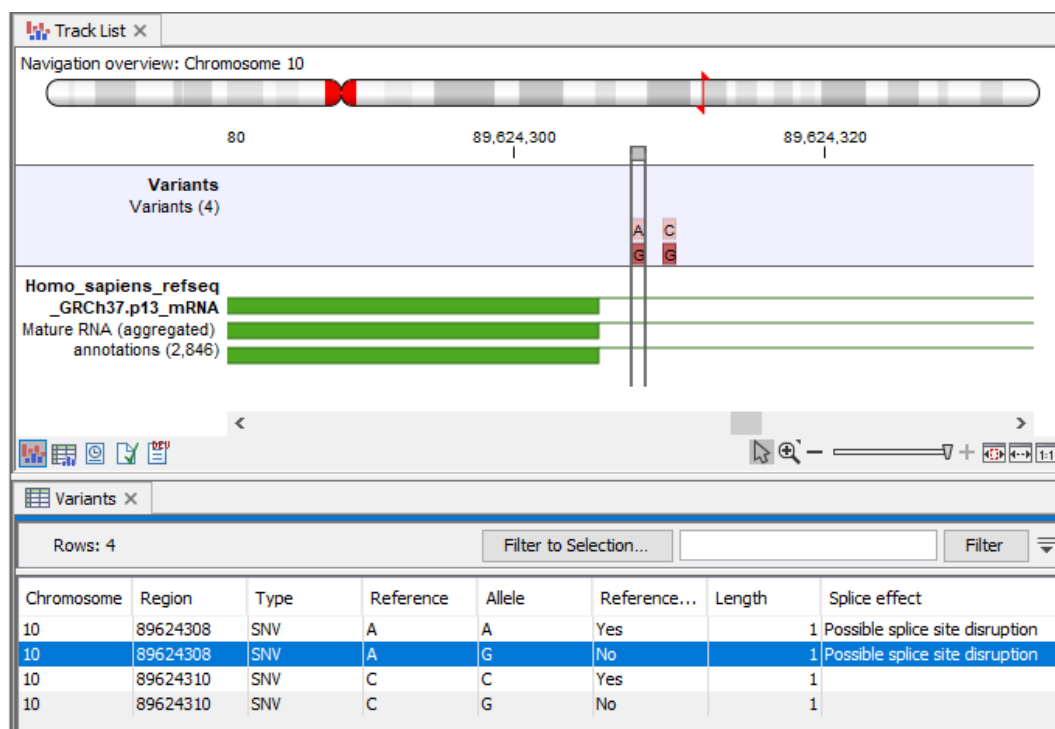


Figure 32.24: Results of the Predict Splice Site Effect tool.

32.5.3 GO Enrichment Analysis

This tool can be used to investigate candidate variants, or better their corresponding altered genes for a common functional role. For example if you would like to know what is interesting in the zebu cattle in comparison to bison and taurine cattle, you can use this tool. For that approach, first filter all found variants in zebu for zebu-specific variants and afterwards run the GO enrichment test for biological process to see that more variants than expected are in immune response genes. These can then be further investigated.

You will need a GO association file to run this tool. Such a file includes gene names and associated Gene Ontology terms and can be downloaded from the Gene Ontology web site for various species. Go to <http://www.geneontology.org/GO.downloads.annotations.shtml>, and find the relevant GAF download page from there. For example, if your species is available from the download page for commonly studied organisms, go to <https://geneontology.org/docs/download-go-annotations/> and download the GO Annotations of the relevant species by clicking on the *.gz link in the table (see figure 32.25). Import the downloaded annotations into the workbench using **Import | Standard Import**.

You will also need a Gene track for the relevant species (learn more about gene tracks in section 27.1).

To run the **GO Enrichment Analysis** tool, go to:

Tools | Resequencing Analysis | **Functional Consequences** | **GO Enrichment Analysis**

First, select the variant track containing the variants to analyse. You then have to specify both the annotation association file and the gene track. Finally, choose which ontology (cellular component, biological process or molecular function) you would like to test for (see figure 32.26).

Filtered Annotation File Downloads

Species/Database	Gene products annotated	Annotations	Submission date	README	File
Leishmania major Sanger GeneDB	2781	6271 (6271 non-IEA)	-	README	gene_association.GeneDB_Lmajor.gz (198 kb)
Plasmodium falciparum Sanger GeneDB	2373	6298 (6298 non-IEA)	-	README	gene_association.GeneDB_Pfalciparum.gz (182 kb)
Trypanosoma brucei Sanger GeneDB	6365	19050 (19050 non-IEA)	-	README	gene_association.GeneDB_Tbrucei.gz (512 kb)
Agrobacterium tumefaciens str. C58 PAMGO	83	250 (250 non-IEA)	-	README	gene_association.PAMGO_Atumefaciens.gz (3 kb)

Figure 32.25: Download the GO Annotations by clicking on the *.gz link in the table.

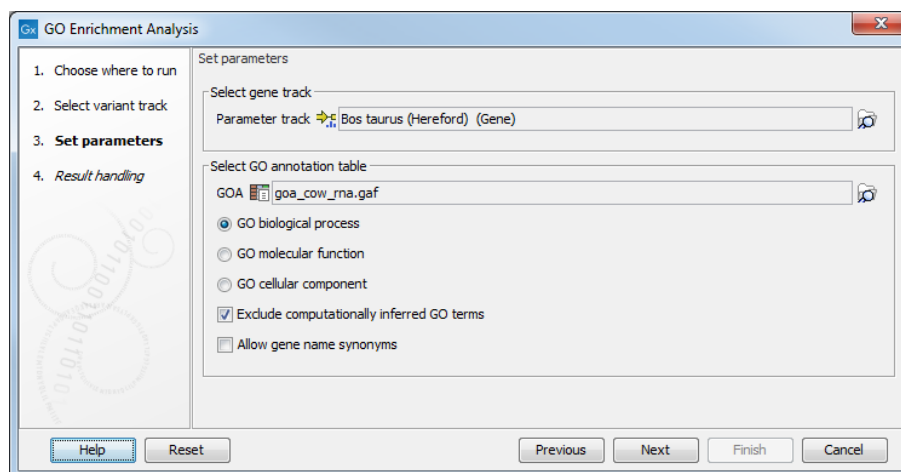


Figure 32.26: The GO enrichment settings.

The analysis starts by associating all of the variants from the input variant file with genes in the gene track, based on overlap with the gene annotations.

Next, the Workbench tries to match gene names from the gene (annotation) track with the gene names in the GO association file. Note that the same gene name definition should be used in both files.

Finally, an hypergeometric test is used to identify over-represented GO terms by testing whether some of the GO terms are over-represented in a given gene set, compared to a randomly selected set of genes.

The result is a table with GO terms and the calculated p-value for the candidate variants, as well as a new variant file with annotated GO terms and the corresponding p-value (see figure 32.27). The p-value is the probability of obtaining a test statistic at least as extreme as the one that was actually observed, or in other words how significant (trustworthy) a result is. In case of a small p-value the probability of achieving the same result by chance with the same test statistic is very small.

In addition to the p-values, the table lists for each GO term the number and names of the genes that matched, and the number and names of the matched genes that contains at least one variant.

GO term	Description	Matched Genes	Genes with Variations	P-values
0007605	sensory perception of sound	76	67	0.00
0030198	extracellular matrix organization	157	111	0.00
0001501	skeletal system development	101	76	1.15E-14
0045893	positive regulation of transcription, DNA-templated	512	275	3.01E-14
0007507	heart development	104	77	3.50E-14
0007601	visual perception	76	59	1.07E-12
0060048	cardiac muscle contraction	32	31	1.25E-12
0010628	positive regulation of gene expression	256	151	1.97E-12
0086091	regulation of heart rate by cardiac conduction	34	32	5.24E-12
0045494	photoreceptor cell maintenance	25	25	2.24E-11
0045944	positive regulation of transcription from RNA polymerase ...	840	405	5.49E-11
0060271	cilium assembly	155	97	1.97E-10
0001947	heart looping	36	32	2.24E-10
0006418	RNA aminoacylation for protein translation	33	29	2.99E-9
0035725	sodium ion transmembrane transport	33	29	2.99E-9
0001822	kidney development	46	37	3.18E-9
0042391	regulation of membrane potential	101	67	3.86E-9
0043066	negative regulation of apoptotic process	393	203	5.74E-9
0000122	negative regulation of transcription from RNA polymerase...	570	280	6.14E-9
0051289	protein homotetramerization	57	43	6.19E-9

Figure 32.27: The results of the analysis.

32.5.4 Download 3D Protein Structure Database

This tool downloads the 3D Protein Structure Database from a public accessible HTTP location hosted by QIAGEN Aarhus.

The database contains a curated set of sequences with known 3D structures, which are obtained from the Protein Data Bank (<https://www.rcsb.org/>) [Berman et al., 2003]. The information stored in the database (e.g. protein sequence, X-ray resolution) is used to identify suitable structural templates when using the **Link Variants to 3D Protein Structure** tool.

To download the database, go to:

Tools | Resequencing Analysis (📁) | **Functional Consequences** (📁) | **Download 3D Protein Structure Database** (📁)

If you are connected to a server, you will first be asked about whether you want to download the data locally or on a server. In the next wizard step you are asked to select the download location (see figure 32.28).

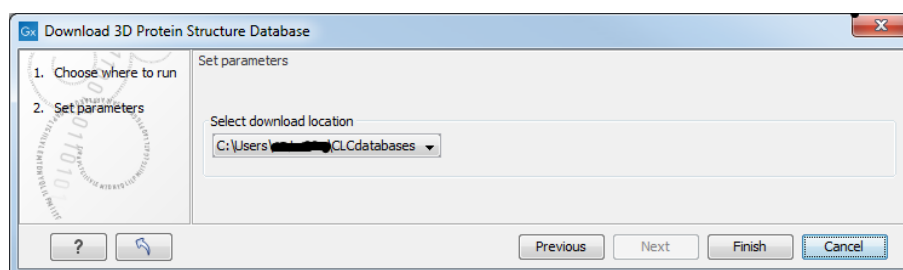


Figure 32.28: Select the download location.

The downloaded database will be installed in the same location as local BLAST databases (e.g. <username>/CLCdatabases) or at a server location if the tool was executed on a CLC Server. From the wizard it is possible to select alternative locations if more than one location is available.

When new databases are released, a new version of the database can be downloaded by invoking the tool again (the existing database will be replaced).

If needed, the **Manage BLAST Databases** tool can be used to inspect or delete the database (the database is listed with the name 'ProteinStructureSequences'). You can find the tool here:

BLAST (📁) | **Manage BLAST Databases** (📁)

32.5.5 Link Variants to 3D Protein Structure

This tool makes it possible to visualize variant consequences on 3D protein structures. It takes a variant track as input, and produces a new variant track as output, with two additional columns in the table view:

- **Link to 3D protein structure:** If a variant affects the amino acid composition of a protein, and a 3D structure of sufficient homology can be found in the Protein Data Bank (PDB), a link is provided in this column. Via the link, the structure can be downloaded and a 3D model and visualization of the variant consequences on the protein structure will be created.
- **Effect on drug binding site:** If any of the homologous structures found in PDB has a drug or ligand in contact with the amino acid variation, a link is provided in this column. Via the link, a list of drug hits can be inspected. The list has links for creating 3D models and visualizations of the variant-drug interaction.

In section 32.5.5 it is described how to interpret the output in the variant table and how the tool finds appropriate protein structures to use for the visualizations, and in section 32.5.5 and onwards it is described how the 3D models and visualizations are created.

Note: Before running the tool, a protein structure sequence database must be downloaded and installed using the **Download 3D Protein Structure Database** tool (see see section 32.5.4).

To run the tool, select:

Tools | Resequencing Analysis (📁) | **Functional Consequences** (🔍) | **Link Variants to 3D Protein Structure** (🔗)

If you are connected to a server, you will first be asked where you want to run the analysis. In the next wizard step you will be asked for an input file. The **Link Variants to 3D Protein Structure** accepts variant tracks as input (see figure 32.29).

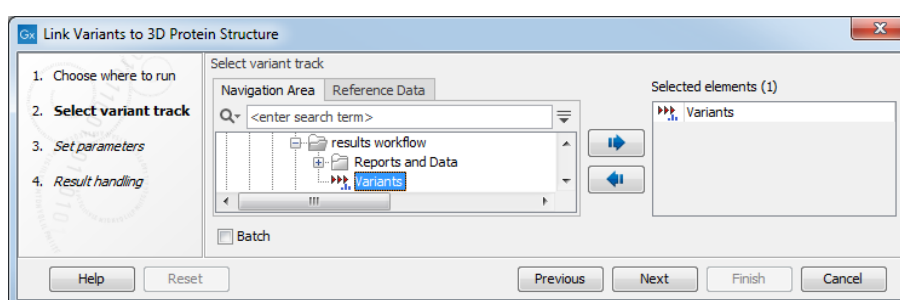


Figure 32.29: Select the variant track holding the variants that you would like to visualize on 3D protein structures.

Click **Next**. In the next wizard step, you must provide a CDS track and the reference sequence track (figure 32.30).

If you have not already downloaded a CDS and a reference sequence track, this can be done using the Reference Data Manager (see section 11.1).

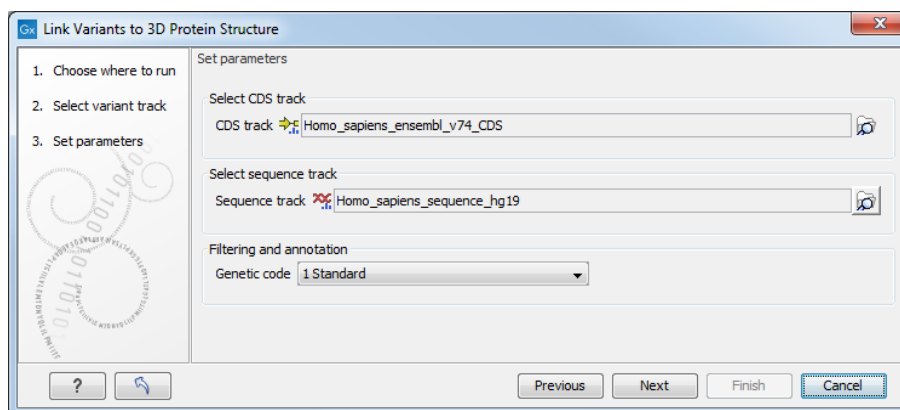


Figure 32.30: Select CDS and reference sequence.

Click **Next**, choose where you would like to save the data, and click on the button labeled **Finish**.

As output, the tool produces a new variant track, with two additional columns in the table view ('Link to 3D protein structure' and 'Effect on drug binding site' - figure 32.31). The default output view is the variant track. To shift to table view, click on the table icon found in the lower left corner of the View Area.

output_new-2				
Rows: 1,254		Table view: Genome		
				Filter
pt	Coding region change in longest transcript	Non-synonymous	Link to 3D protein structure	Effect on drug binding site
	NP_214517.1:c.167C>T	Yes	recF Ala56Val	No drug hits (recF Ala56Val)
	NP_214517.1:c.175C>T	Yes	recF Pro59Ser	No drug hits (recF Pro59Ser)
	NP_214517.1:c.734T>C	Yes	recF (no PDB hits)	recF (no PDB hits)
	NP_214520.1:c.61G>C	Yes	gyrA Glu21Gln	No drug hits (gyrA Glu21Gln)
	NP_214520.1:c.284G>C	Yes	gyrA Ser95Thr	1 drug hit (gyrA Ser95Thr)
	NP_214520.1:c.2003G>A	Yes	gyrA Gly668Asp	No drug hits (gyrA Gly668Asp)
	NP_214522.1:c.433T>C	Yes	Rv0008c (no PDB hits)	Rv0008c (no PDB hits)
	NP_214526.1:c.697T>C	Yes	Rv0012 (no PDB hits)	Rv0012 (no PDB hits)
	YP_177615.1:c.204C>G	Yes	trpG Ile68Met	No drug hits (trpG Ile68Met)
	NP_214528.1:c.1703G>A	Yes	pknB Gly568Asp	No drug hits (pknB Gly568Asp)
	NP_214532.1:c.1387C>T	Yes	pstP (no PDB hits)	pstP (no PDB hits)
	NP_214534.1:c.723_724delACinsGG	Yes	fhaA (no PDB hits)	fhaA (no PDB hits)
	NP_214534.1:c.720G>C	Yes	fhaA (no PDB hits)	fhaA (no PDB hits)
	NP_214535.1:c.134delG	Yes	Rv0021c (frame shift)	Rv0021c (frame shift)

Figure 32.31: The variant table output.

The variant table output

For each variant in the input, the Link Variants to 3D Protein Structure tool does the following, to prepare output for the "Link to 3D protein structure" and "Effect on drug binding site" columns in the output variant track:

1. Evaluate if the variant is found inside a CDS region. Otherwise the following is returned for the variant: **(outside CDS regions)**.
2. If the variant is in a CDS region, translate the reference sequence of the impacted gene into an amino acid sequence and evaluate if the variant can be expected to have an effect on protein structure that can be visualized. Overlapping genes (common in prokaryotic genomes) with different reading frames may cover a given variation, in which case multiple protein sequences will be considered.

For variants that cannot be visualized, the gene name and one of the reasons given below will be listed in the output table:

- **(nonsense)** - the variant would result in a stop codon being introduced in the protein sequence.
 - **(synonymous)** - the variant would not change the amino acid.
 - **(frame shift)** - the variant would introduce a frame shift.
3. BLAST the translated amino acid sequence (the query sequence) against the protein structure sequence database (see section 32.5.4) to identify structural candidates. Note that if multiple splicing variants exist, the protein structure search is based on the longest splicing variant. BLAST hits with E-value > 0.0001 are rejected and a maximum of 2500 BLAST hits are retrieved. If no hits are obtained, the gene name and the message **(no PDB hits)** are listed.
 4. For each BLAST hit, check if the variant is covered by the structure. For a variant resulting in one amino acid being replaced by another, the affected amino acid position should be present on the structure. For a variant resulting in amino acid insertions or deletions, the amino acids on both sides of the insertion/deletion must be present on the structure.
 5. For the BLAST hits covering the variant, rank the structures considering both structure quality and homology (see section 20.6.2).
 6. Add the gene name and the description of the amino acid change to the "Link variant to 3D protein structure" column in the output variant track. A link on the description gives access to a 3D view of the variant effect using the best ranked protein structure from point 5 (see section 32.5.5). Note that the amino acid numbering is based on the longest CDS annotation found.
 7. Extract all BLAST hits from point 5, where the affected amino acid(s) are in contact with a drug or ligand in the PDB file (heavy atoms within 5 Å). If no structures with variant-drug interaction are found, the following is returned to the "Effect on drug binding site" column: **No drug hits** together with the gene name and the description of the amino acid change. If structures with variant-drug interaction are found, the number of different drugs or ligands encountered are written to the "Effect on drug binding site" column as **X drug hits**. From a link on "X drug hits", a list describing the drug hits in more detail can be opened. The list also has a link for each drug, to create a 3D model and visualization of the variant-drug interaction, see section 32.5.5.

Create 3D visualization of variant

Clicking a link provided in the 'Link to 3D Protein Structure' column will show a menu with three options:

- 'Download and Show Structure' will open a 3D view visualizing the consequences of the variant on a protein structure (figure 32.31).
- 'Download and Show All Variants (x) on Structure' will open a 3D view visualizing the consequences of x variants on the same protein structure (figure 32.32).

Note 1: Only variants shown in the table will be included in the view (e.g. variants filtered out will be ignored).

Note 2: It is not always possible to visualize variants on the same gene together on the same structure, since many structures in the PDB only cover parts of the whole protein.

Note 3: Even though variants may be possible to visualize together, it does not necessarily mean they occur together on the same protein. For example, in diploid cells, heterozygous variants may not.

- "Help" gives access to this documentation.

If you have problems viewing 3D structures, please check your system matches the requirements for 3D viewers. See section 1.3.

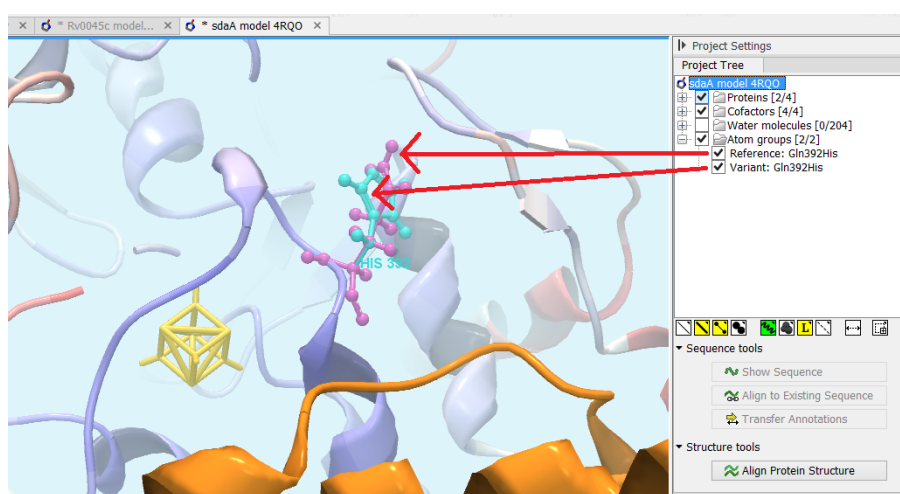


Figure 32.32: Generated 3D view of variant. The reference amino acid is seen in purple and the variant in cyan on top of each other. Only the backbone of protein structures are visualized by default. The modeled protein structure is colored to indicate local model uncertainty - red for flexible and uncertain parts of the structure model and blue for very well defined and accurate parts of the structure model. Other molecules from the PDB file are colored orange or yellow.

The "Download and Show...." options will do the following:

1. **Download and import** the PDB file containing the protein structure for the variant (found by the 'Link Variants to 3D Protein Structure' tool, see section 32.5.5).
2. **Generate biomolecule** involving the modeled chain, if the information is available in the PDB file (see Infobox below).
3. **Create an alignment** between the reference protein sequence for the gene impacted by the variant (the query sequence) and the sequence from the protein structure (the template structure).
4. **Create a model structure for the reference** by mapping it onto the template structure based on the sequence alignment (see section 20.6.2).

5. **Create a model structure with variant(s)** by mapping the protein sequence with the variant consequences onto the reference structure model (see section 20.6.2).
6. **Open a 3D view** (a Molecule Project) with the variant structure model shown in backbone representation. The model is colored by temperature (see figure 32.32), to indicate local model uncertainty (see section 20.6.2). The consequence(s) of the variant(s) are highlighted by showing involved amino acids in ball n' sticks representation with the reference colored purple and the variant cyan. Other molecules from the PDB file are shown in orange or yellow coloring (figure 32.32).

From the Project Tree in the Side Panel of the Molecule Project, the category 'Atom groups' contains two entries for each variant shown on the structure - one entry for the reference and one for the variant (figure 32.32). The atom groups contain the visualization of the variant consequence on structure. For variants resulting in amino acid replacements, the affected amino acid is visualized. For variants resulting in amino acid insertions or deletions, the amino acids on each side of the deletion/insertion are visualized.

The template structure is also available from the Proteins category in the Project Tree, but hidden in the initial view. The initial view settings are saved on the Molecule Project as "Initial visualization", and can always be reapplied from the View Settings menu (⚙) found in the bottom right corner of the Molecule Project (see section 4.6).

Tip: Double-click an entry in the Project Tree to zoom the 3D view to the atoms.

You can save the 3D view (Molecule Project) in the Navigation Area for later inspection and analysis. Read more about how to customize visualization of molecules in section 17.3.

Protein structures imported from a PDB file show the tertiary structure of proteins, but not necessarily the biologically relevant form (the quaternary structure). Oftentimes, several copies of a protein chain need to arrange in a multi-subunit complex to form a functioning biomolecule. In some PDB files several copies of a biomolecule are present and in others only one chain from a multi-subunit complex is present. In many cases, PDB files have information about how the molecule structures in the file can form biomolecules. In *CLC Genomics Workbench* variants are therefore shown in a protein structure context representing a functioning biomolecule, if this information is available in the selected template PDB file.

Visualize drug interaction

Clicking a link provided in the 'Drug interaction in protein 3D structure' column will open a list with information about the drug hits (figure 32.33).

The information available in the drug hit list is the following:

- **PDB hit.** Clicking a link provided in the PDB hit column will show a menu with two options: "Download and Show Structure" and "Help". "Help" gives access to this documentation. The "Download and Show Structure" option does exactly as described in section 32.5.5, except that the final 3D visualization is centered on the drug, and the drug is shown in ball n' sticks representation with atoms colored according to their atom types (figure 32.34).
- **PDB drug name.** (Hidden by default) The identifier used by PDB for the ligand or drug.

Rows: 7 Summary of hits found				
PDB hit	Drug name	DrugBank info	Organism	Description
2Y3P	SIMOCYCLINONE D8		ESCHERICHIA COLI	CRYSTAL STRUCTURE OF N-TERMINAL DOMAIN OF GYRA WITH THE ANTIBIOTIC SIMOCYCLINO
4KCF			STREPTOCOCCUS PNEUMONIAE	NOVEL FLUOROQUINOLONES IN COMPLEX WITH TOPOISOMERASE IV FROM S. PNEUMONIAE AN
3E0E	MOXIFLOXACIN (AVELOX)	DB00218	STREPTOCOCCUS PNEUMONIAE	STRUCTURAL INSIGHT INTO THE QUINOLONE-DNA CLEAVAGE COMPLEX OF TYPE IIA TOPOISON
3E0E	CLINAFLOXACIN		STREPTOCOCCUS PNEUMONIAE	STRUCTURAL INSIGHT INTO THE QUINOLONE-DNA CLEAVAGE COMPLEX OF TYPE IIA TOPOISON
3QX3	ETOPOSIDE (EPO SIN)	DB00773	HOMO SAPIENS	HUMAN TOPOISOMERASE IIBETA IN COMPLEX WITH DNA AND ETOPOSIDE
4G0V	MITOXANTHONE (MITOX)	DB01204	HOMO SAPIENS	HUMAN TOPOISOMERASE IIBETA IN COMPLEX WITH DNA AND MITOXANTHONE
4G0V	AMETANTRONE		HOMO SAPIENS	HUMAN TOPOISOMERASE IIBETA IN COMPLEX WITH DNA AND AMETANTRONE

Figure 32.33: An example of a drug hit table with information about the drug and with links to 3D visualizations of variant-drug interaction.

- **Drug name.** When possible, a common name for the potential drug is listed here. The name is taken from the corresponding DrugBank entry (if available) or from the PDB header information for the PDB hit.
- **DrugBank info.** If information about the drug or ligand is available in DrugBank (www.drugbank.ca [Law et al., 2014, Wishart et al., 2006]), a weblink to the appropriate site is listed here.
- **E-value.** (Hidden by default) The E-value is a measure of the quality of the match returned from the BLAST search. The closer to zero, the more homologous is the template structure to the query sequence.
- **Organism.** (Hidden by default) The organism for which the PDB structure has been obtained.
- **Description.** The description of the PDB file content, as given in the header of the PDB hit.

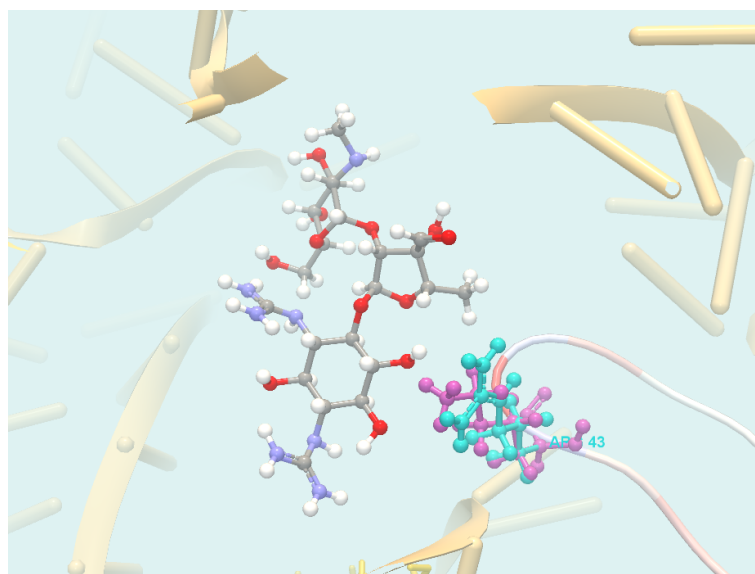







Figure 32.34: 3D visualization of variant-drug interaction. The drug (streptomycin) is in center of the view, and colored according to its atom types. In this picture, the foreground has been cut away using the clipping plane functionality found in the Visualization tab in the Side Panel, to better see the variant-drug interaction.

To learn more about model structure, see section [20.6.2](#).

32.6 Create Consensus Sequences from Variants

Using the Create Consensus Sequences from Variants tool, consensus sequences can be created from a Variant track () and the matching reference genome (.

It is possible to mask out low coverage regions () with N's when a coverage track is provided. To create a low coverage region track use the **Create Mapping Graph** () tool, see section 27.9.2, to identify coverage in your sample followed by filtering on specified frequency using the **Identify Graph Threshold Areas** () tool, see section 27.9.3.

A number of variant filtering options defining criteria for inclusion in the consensus sequence are available:

- **Minimum frequency for inclusion** Include variants above a selected frequency in the consensus.
- **Ambiguity threshold** Mask out positions where the most commonly observed nucleotide is seen in fewer than the specified fraction of reads.
- **Ignore frameshift variants** Filter out Indels of size 1,2,4,5 etc. A minimum frequency can be specified for when to include a variant.

For variants remaining after filtering, the following rules apply for sites where variants overlap:

- If multiple SNVs overlap, the relevant IUPAC ambiguity code is inserted in the consensus sequence. For example, if a site has the variants G and A, the ambiguity code R is inserted.
- If a deletion overlaps an insertion or an SNV, only the variant with the highest frequency is included in the consensus sequence. In the case where the frequency is the same, only the deletion is included.

Running the Create Consensus Sequences from Variants tool

To run the Create Consensus Sequences from Variants tool, go to:

Tools | Resequencing Analysis () | **Create Consensus Sequences from Variants** (

In the first step, select the variant track that contains the variants from which the consensus should be created.

The next step provides options for how to construct the consensus sequences (see figure 32.35).

In the Tracks parameters, specify the reference genome and optionally provide an annotation track containing low coverage regions.

Variant handling parameters include:

- **Minimum frequency for inclusion** ranging from 0.0 to 1.0 (Default 0.8).
- **Ambiguity threshold** for masking variants with N's range from 0.0 to minimum frequency for inclusion (Default is 0.5). When generating the consensus, positions with variants that have a frequency between the ambiguity threshold and the minimum frequency for

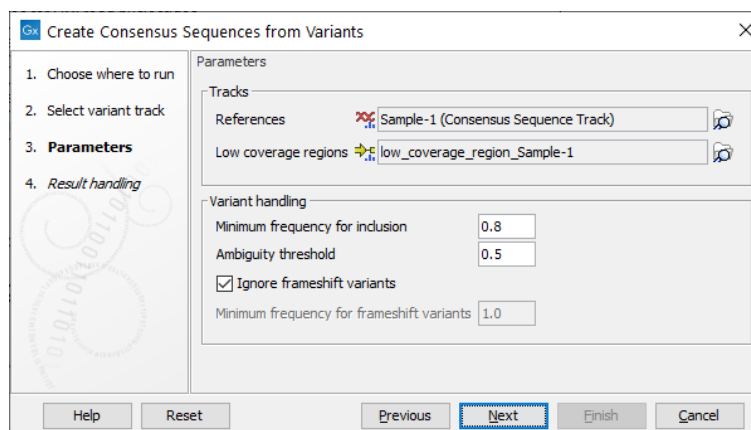


Figure 32.35: Options available to configure when running the *Create Consensus Sequences from Variants* tool.

inclusion are replaced with N's. Positions with variants below the ambiguity threshold use the nucleotide from the reference. Note: only none-Indel variants can be masked using this threshold.

- **Ignore frameshift variants.** Enabled when ticked, otherwise a frequency can be specified in order to include frameshift at a certain frequency (default is 1.0). Note, frameshift variants are here defined to cover Indels not fitting with a 3 codon base structure in its simplest form. This feature is especially relevant for Virus consensus creation where frameshift variants are unlikely.

In the final step, specify the output type and a save data location.

The tool offers two types of consensus sequence format:

- **Sequence Track** A Consensus Genome Track.
- **Sequence List** A Consensus Sequence list.

Chapter 33

RNA-Seq and Small RNA analysis

Contents

33.1 RNA-Seq normalization	949
33.2 Create Expression Browser	951
33.2.1 The expression browser	953
33.2.2 Expression browser plot	955
33.3 miRNA analysis	960
33.3.1 Quantify miRNA	960
33.3.2 Annotate with RNAcentral Accession Numbers	967
33.3.3 Create Combined miRNA Report	968
33.3.4 Extract IsomiR Counts	968
33.3.5 Explore Novel miRNAs	970
33.4 RNA-Seq Tools	971
33.4.1 RNA-Seq Analysis	971
33.4.2 RNA-Seq Analysis for Long Reads	995
33.4.3 Detect and Refine Fusion Genes	1000
33.4.4 Import Expression Data	1015
33.5 Expression Plots	1018
33.5.1 PCA for RNA-Seq	1018
33.5.2 Create Sample Level Heat Map for RNA-Seq	1022
33.5.3 Create Feature Level Heat Map for RNA-Seq	1025
33.5.4 Create K-medoids Clustering for RNA-Seq	1030
33.6 Differential Expression	1036
33.6.1 Pre-filtering data for Differential Expression	1037
33.6.2 The GLM model	1038
33.6.3 Differential Expression in Two Groups	1042
33.6.4 Differential Expression for RNA-Seq	1045
33.6.5 Output of the Differential Expression tools	1050
33.6.6 Create Venn Diagram for RNA-Seq	1055
33.6.7 Gene Set Test	1057

Based on an annotated reference genome, *CLC Genomics Workbench* supports **RNA-Seq Analysis** by mapping next-generation sequencing reads and distributing and counting the reads across genes and transcripts. Subsequently, the results can be used for expression analysis. The tools from the RNA-Seq and Small RNA Analysis folder automatically account for differences due to sequencing depth, removing the need to normalize input data.

RNA-Seq analysis, expression analysis, and other tools can be included in workflows. Designing a workflow that includes an RNA-Seq Analysis step, which is typically run once per sample, and an expression analysis step, typically run once to analyze all the samples, is described in section 14.3.3.

Metadata and RNA-Seq

The statistical analysis and visualization tools of the RNA-Seq folder make extensive use of the metadata system. For example, metadata are required when defining the experimental design in the Differential Expression for RNA-Seq tool, and can be used to add extra layers of insight in Create Sample Level Heat Map, Create Feature Level Heat Map for RNA-Seq, and PCA for RNA-Seq.

To get the most out of these tools we recommend that all input expression tracks are associated with metadata. Importing metadata and associated data elements with it is described in section 13.

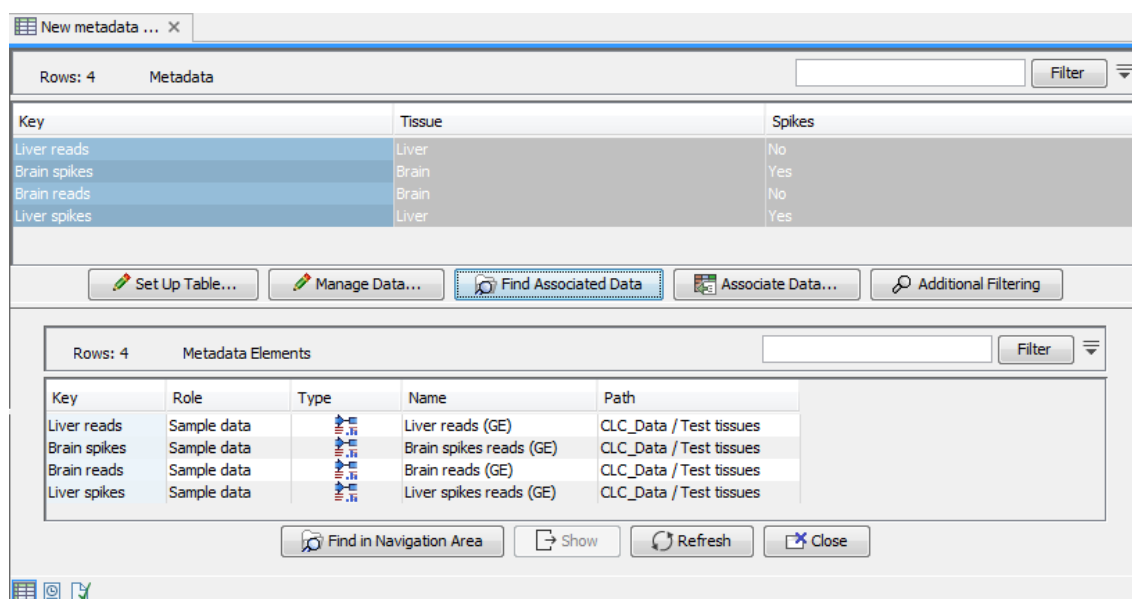


Figure 33.1: A metadata table with expression samples associated with it.

33.1 RNA-Seq normalization

Many RNA-Seq tools compare samples based on their read counts. This section provides a brief overview of the normalization used by these tools so as to make read counts from different samples more comparable to each other.

Since the sequencing depth might differ between samples, a per-sample library size normalization must be performed before samples can be compared. Two such normalizations are supported: TMM normalization, and Housekeeping gene normalization.

For all relevant tools included in the RNA-Seq folder, either the TMM normalization is automatically applied, or an option is provided to choose between TMM normalization and Housekeeping gene normalization.

Per-sample library size normalization produces a single number for each sample that can be used to weight the counts data from that sample. The tools Differential Expression for RNA-Seq and Differential Expression in Two Groups use this number in their statistical model: for sample i , the library size normalization factor is the constant_i described in section 33.6.2.

Other tools do not use a statistical model and instead generate normalized counts.

PCA for RNA-Seq, Create Sample Level Heat Map for RNA-Seq and Create Feature Level Heat Map for RNA-Seq automatically perform TMM normalization (Robinson and Oshlack, 2010), followed by logCPM and Z-Score normalization.

Create Expression Browser automatically perform TMM normalization, followed by logCPM.

Differential Expression for RNA-Seq and Differential Expression in Two Groups can perform TMM or housekeeping gene normalization.

TMM Normalization For TMM normalization, a TMM factor is computed by comparing the samples against a reference sample. The reference is the sample that has the count-per-million upper quartile closest to the mean upper quartile.

TMM normalization adjusts library sizes based on the assumption that most genes are not differentially expressed. Therefore, it is important not to make subsets of the count data before doing statistical analysis or visualization, as this can lead to differences being normalized away.

Housekeeping gene normalization Housekeeping genes can either be specified directly, or the most suitable subset of a short list of genes can be selected using the GeNorm algorithm of Vandesompele et al., 2002 (see <https://genomebiology.biomedcentral.com/articles/10.1186/gb-2002-3-7-research0034>).

Once a set of housekeeping genes has been chosen, the normalization factor for a sample is the natural logarithm of the geometric mean of the expressions of the genes for that sample.

We recommend the use of housekeeping genes rather than TMM when working with Targeted RNA Panels, or in situations where the TMM assumption that most genes are not differentially expressed does not hold.

It is not possible to view housekeeping gene normalized expressions within the Workbench. However, these values are relatively easy to calculate. To do so:

- Export the raw expression values. It is recommended to use the Create Expression Browser tool to create a single table of all the samples, then to export this to a table format, such as .xlsx or .csv choosing to **Export table as currently shown**. For more details on export options see section 8.1.6.
- For each sample, find the geometric mean of the house-keeping gene expressions.

- The housekeeping gene normalized expression of a gene is $1000 * \text{raw expression} / \text{geometric mean}$. This factor of 1000 is arbitrary - any number can be used such that the normalized expressions have a size that is easy to work with.

As an example of this procedure, consider the following expressions, where HKG1 and HKG2 are housekeeping genes:

Gene	Sample1	Sample2	Sample3
HKG1	1	3	2
HKG2	3	3	2
Gene1	8	6	2
Gene2	4	4	4
Gene3	1	0	1

The geometric mean of the housekeeping gene expressions is:

Sample1	Sample2	Sample3
$\sqrt[3]{3}$	3	2

The normalized expressions are then:

Gene	Sample1	Sample2	Sample3
Gene1	4619	2000	1000
Gene2	2309	1333	2000
Gene3	577	0	500

logCPM TMM-adjusted logCPM counts (similar to the EdgeR approach ([Robinson et al., 2010])) are calculated as follows:

1. We add a prior to the raw counts. This prior is 1.0 per default, but is scaled based on the library size as `scaled_prior = prior*library_size/average_library_size`.
2. The library sizes are also adjusted by adding a factor of 2.0 times the prior to them (for explanation, see <https://support.bioconductor.org/p/76300/>).
3. The logCPM is now calculated as `log2(adjusted_count * 1E6 / adjusted_library_size)`.

Z-Score normalization Z-Score normalization performs a final cross-sample normalization. For each row (gene/transcript), a Gaussian normalization (Z-Score normalization) is applied: data is shifted and scaled so that the mean is zero, and the standard deviation one.

33.2 Create Expression Browser

The **Create Expression Browser** tool makes it possible to inspect gene, transcript or miRNA expression level counts, annotations and statistics for many samples at the same time. Further, the plotting functionality helps with visualizing the data in an easy to interpret way.

To run the tool, go to:

Tools | RNA-Seq and Small RNA Analysis (🇺🇸) | Create Expression Browser (🇬🇧)

Select some expression tracks (🇺🇸🇬🇧). These can be either Gene level (GE) or Transcript level Expression (TE) tracks, but not a combination of both (see figure 33.2).

Click on the **Next** button.

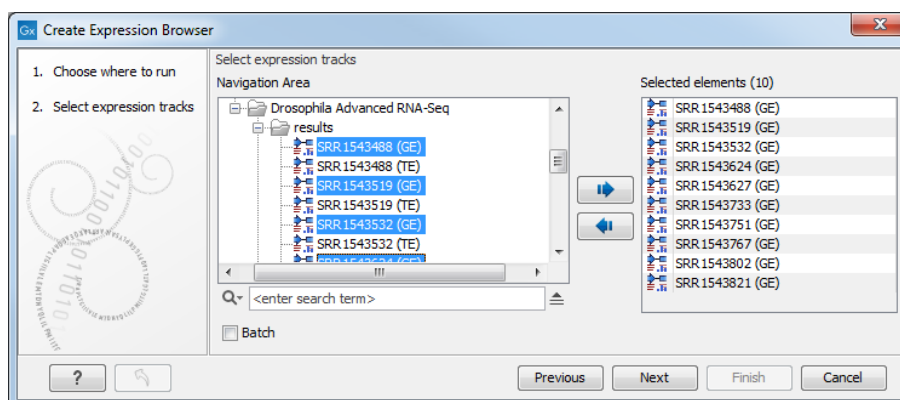


Figure 33.2: Select expression tracks, either GE or TE.

In the second wizard dialog, statistical comparisons and an annotation resource can be selected (see figure 33.3). Information from the selected elements is included in the expression browser.

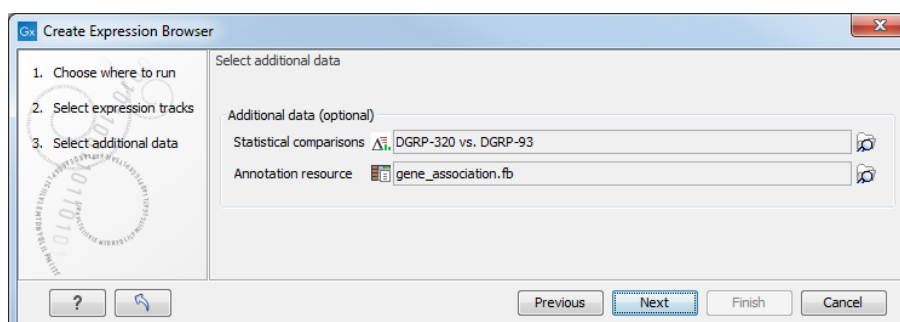


Figure 33.3: Information from statistical comparisons and an annotation resource can optionally be included in the expression browser being created.

Statistical comparisons are generated by differential expression tools, described in section 33.6. The selected statistical comparisons must have been created using the same kind of expression tracks as those selected in the first wizard step. For example, when creating an expression browser using GE expression tracks, statistical comparisons must have been generated using GE tracks.

Annotation resources can come from various sources:

- Annotations from the Gene Ontology consortium at <https://current.geneontology.org/products/pages/downloads.html>. The files downloaded from there can be imported using the Standard Import tool.
- Annotations contained in tables. To use such tables as an annotation resource, at least one column must have a header from the list in table 33.4, spelled exactly as shown,

Accession	GeneName	RefSeq transcript ID
Ensembl	GeneSymbol	RefSeq_ID
EnsemblID	GI	RefSeqAccession
Entrez gene	Gid	RGD name
Entrez_Gene_ID	Identifier	SGD accession number
EntrezGeneID	MGI name	Symbol
Feature ID	Name	Transcript
FlyBase	Primary accession	Transcript ID
GenBank accession	PrimaryAccession	Transcript ID(Array Design)
Genbank identifier	Probe Set ID	transcript_cluster_id
Gene ID	Probe_Id	WormBase
Gene name	ProbeID	
Gene symbol	RefSeq accession	

Table 33.4: Column headers for columns containing identifiers.

and contain identifiers that can be matched with identifier information in the selected expression tracks. Where more than one column has a header from that list, the column with the most matching identifiers is used to determine the row each annotation should be added to.

Excel and CSV format files can be imported as tables using the Standard Import tool. For CSV format files, force the import type to "Table in CSV format (.csv)".

- Annotations generated by the "Blast2GO PRO" plugin (see <https://digitalinsights.qiagen.com/plugins/blast2go-pro/>).
- Annotations for human, mouse and rat bundled with Reference Data Sets.

33.2.1 The expression browser

An Expression Browser is shown in figure 33.4.

The screenshot shows the 'Expression Browser' window with 17,490 rows. The table displays data for genes/transcripts, including their names, identifiers, and expression values across various samples. The table is organized into groups like DGRP-93 and DGRP-370. The right panel shows settings for column width, feature information, and expression values.

Figure 33.4: Expression browser table when no statistical comparison or annotations resources were provided.

Each row represents a gene or a transcript, defined by its name, the chromosome and the region where it is located, as well as an identifier linking to the relevant online database.

The expression values for each sample - or aggregation of samples - can be given by total counts, RPKM, TPM or CPM (TMM-adjusted) (Trimmed Mean of M values adjusted Counts Per Million). These measurements differ from each other in three key ways:

1. RPKM and TPM measure the number of *transcripts* whereas total counts and CPM measure the number of *reads*. The distinction is important because in an RNA-Seq experiment, more reads are typically sequenced from longer transcripts than from shorter ones.
2. RPKM, TPM and CPM are normalized for sequencing-depth so their values are *comparable* between samples. Total counts are not normalized, so values are *not comparable* between samples.
3. CPM (TMM-adjusted) is obtained by applying TMM Normalization (section 33.1) to the CPM values. These values depend on which other samples are included in the Expression browser view. Note also that when comparing multiple samples the sum of CPM (TMM-adjusted) values is no longer one million. In contrast, RPKM and TPM values are not TMM-adjusted, and thus not affected by the presence of other samples in the expression browser (and the sum of TPM values for a given sample is one million).

How do I get the normalized counts used to calculate fold changes? The CPM expression values are most comparable to the results of the Differential Expression for RNA-Seq tool. However, normalized counts are not used to calculate fold changes; instead the Differential Expression for RNA-Seq tool works by fitting a statistical model (which accounts for differences in sequencing-depth) to raw counts. It is therefore not possible to derive these fold changes from the CPM values by simple algebraic calculations.

It is possible to display the values for individual samples, or for groups of samples as defined by the metadata. Using the drop down menus in the "Grouping" section of the right-hand side setting panel, you can choose to group samples according to up to three metadata layers as shown in figure 33.4.

When individual samples are aggregated, an additional "summary statistic" column can be displayed to give either the mean, the minimum, or the maximum expression value for each group of samples. The table in figure 33.4 shows the mean of the expression values for the first group layer that was selected.

If one or more statistical comparisons are provided, extra columns can be displayed in the table using the "Statistical comparison" section of the Settings panel (figure 33.5). The columns correspond to the different statistical values generated by the Differential Expression for RNA-Seq tool as detailed in section 33.6.5.

Expression Browser										
DGRP-320 vs. DGRP-93										
Name	Identifier	Max group m...	Fold change	Log fold cha...	P-value	FDR p-value	Bonferroni	Database	Database identifier	Database symt
Tsp42Ec	FBgn0033124	178.41	-6.44	-2.69	1.35E-3	1.00	1.00	FB	FBgn0033124	Tsp42Ec
betaTry	FBgn0010357	3,076.23	-5.35	-2.42	1.38E-3	1.00	1.00	FB	FBgn0010357	betaTry
GstE5	FBgn0063495	79.64	13.50	3.75	1.68E-3	1.00	1.00	FB	FBgn0063495	GstE5
IM4	FBgn0040653	1,967.23	-4.14	-2.05	1.75E-3	1.00	1.00	FB	FBgn0040653	IM4
CG11400	FBgn0034198	83.33	-180.43	-7.50	1.78E-3	1.00	1.00			

Figure 33.5: Expression browser table when a statistical comparison is present.

If an annotation database is provided, extra columns can be displayed in the table using the "Annotation" section of the Settings panel (figure 33.6). Which columns are available depends on the annotation file used. When using a GO annotation file, the GO biological process column will list for each gene or transcript one or several biological processes. Click on the process

name to open the corresponding page on the Consortium for Gene Ontology webpage. It is also possible to access additional online information by clicking on the PMID, RefSeq, HGNC or UniProt accession number when available.

The screenshot shows the 'Expression Browser' interface. The main table displays gene data with columns: Name, Identifier, Database, Database identifier, Database symbol, and a detailed GO annotation. The table lists genes like CR44667, CG30025, CR43794, PGRP-5C1b, CG43393, CG7744, dap, Ir60d, Jabba, and dpa. The right-hand side panel, titled 'Expression Browser Table Settings', allows users to filter and display various features like Database, Database identifier, Database symbol, Qualifier, GO biological process, GO cellular component, GO molecular function, Database object name, Synonym, Database object type, Taxon ID, Date, Assigned by, Annotation extension, and Gene product form ID. Buttons for 'Select All' and 'Deselect All' are at the bottom of the settings panel.

Figure 33.6: Expression browser table when a GO annotation file is present.

Select the genes of interest and use the button present at the bottom of the table to highlight the genes in other views (volcano plot for instance) or to copy the genes of interest to a clipboard.

33.2.2 Expression browser plot

A plot with bar charts for one or more selected genes/transcripts can be displayed as shown in figure 33.7.

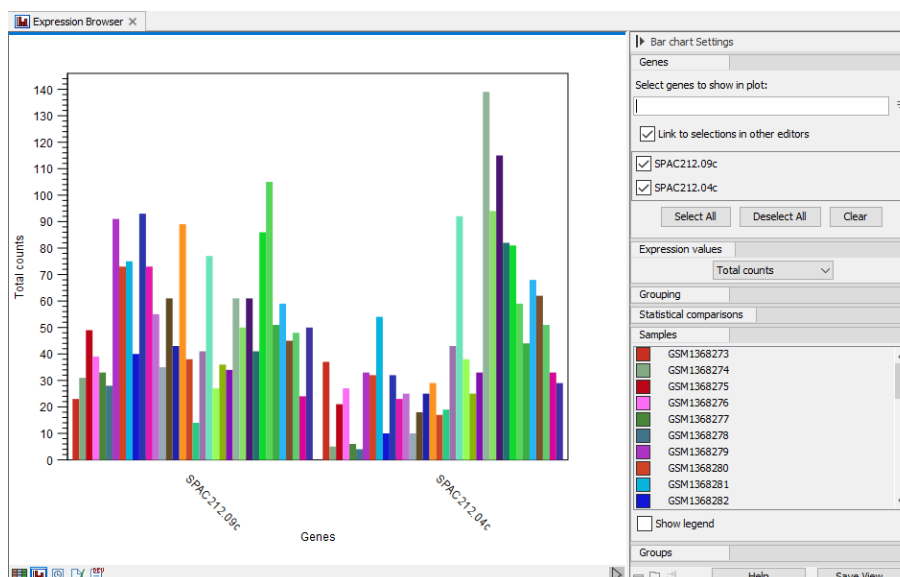


Figure 33.7: Expression browser plot with total counts per sample for two genes

The genes/transcripts can be selected from the search box in the top section of the side panel, titled "Genes" in figure 33.7. Or they can be selected in another editor. A useful setup is a split view of the expression browser with the table in one and the plot in another editor as shown in

figure 33.8. The genes selected in the table will be shown in the plot. The automatic selection can be disabled by unchecking "Link to selections in other editors".

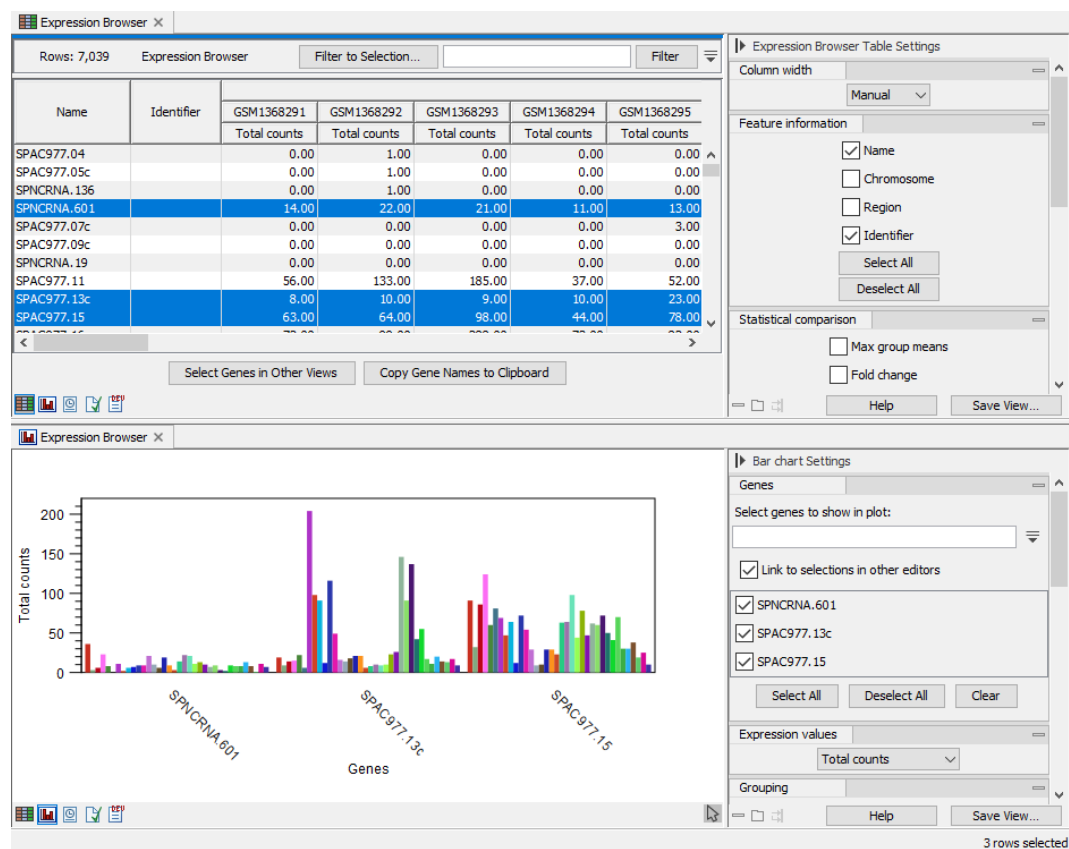


Figure 33.8: Split view of expression browser in table and plot with selections in the table reflected in the plot

The "Expression values" section in the side panel determines what the bars represent: total counts, RPKM, TPM or CPM (TMM-adjusted). See section 33.2.1 for more details on these measurements.

If the samples are associated with metadata, there will be a "Grouping" section in the side panel where it can be selected how to group the samples as shown in figure 33.9. The metadata must be associated prior to the creation of the expression browser.

If multiple groups are selected, they will be nested with all possible combinations starting from the top as shown in figure 33.10.

The bars can also be grouped by metadata whilst being colored by sample as shown in figure 33.11. Or the samples can be collapsed altogether so there is only one bar per group, representing the minimum and maximum values along with an indicator for the average (figure 33.12).

If the expression browser was created with statistical comparisons, there will be a "Statistical comparison" section in the side panel. The statistical comparisons are used to indicate pairs of groups where a gene is differentially expressed subject to a certain threshold. See figure 33.13.

When the first metadata column selected under the "Grouping" section of the side panel matches

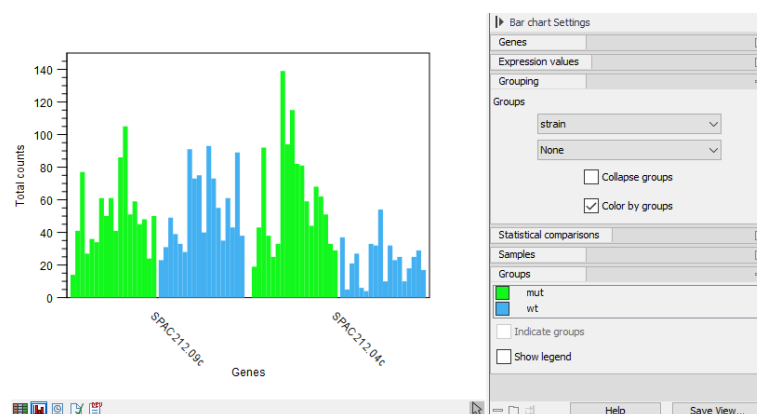


Figure 33.9: Expression browser plot where samples are grouped and colored as defined by metadata

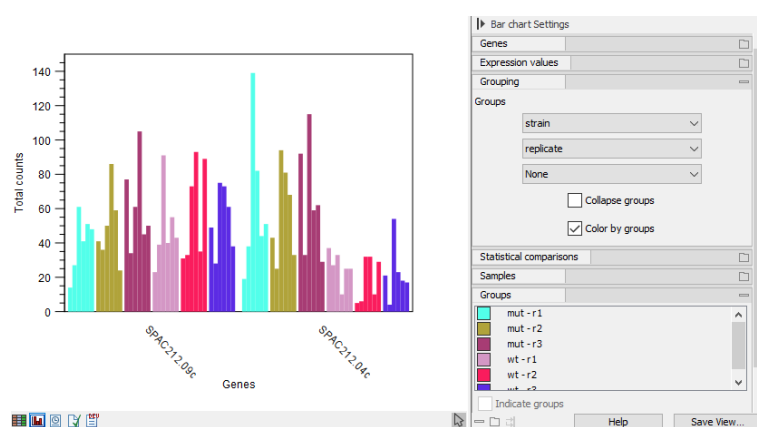


Figure 33.10: Expression browser plot with nested grouping

the factor used when creating one or more statistical comparisons, then these statistical comparisons are applicable and will be listed under the "Statistical comparison" section.

The thresholds can be defined by plain, FDR or Bonferroni corrected p-value. Multiple thresholds can be input. Pairs of groups are indicated if they satisfy one or more thresholds. An example with multiple thresholds is shown in figure 33.14.

If multiple threshold are defined for the same metric - e.g., FDR p-value below 0.05 and FDR p-value below 0.01 - then a pair of groups has an indication for just the smallest (most specific) threshold.

See section 33.6.4 for more details on statistical comparisons.

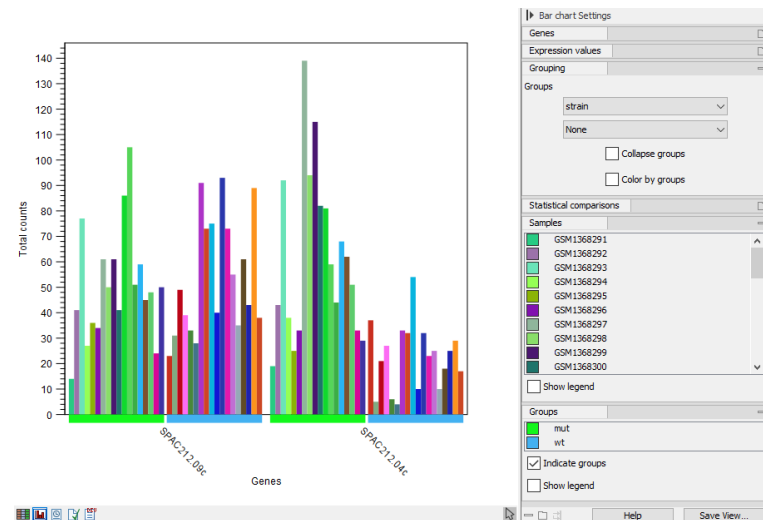


Figure 33.11: Expression browser plot where samples are grouped and colored by metadata

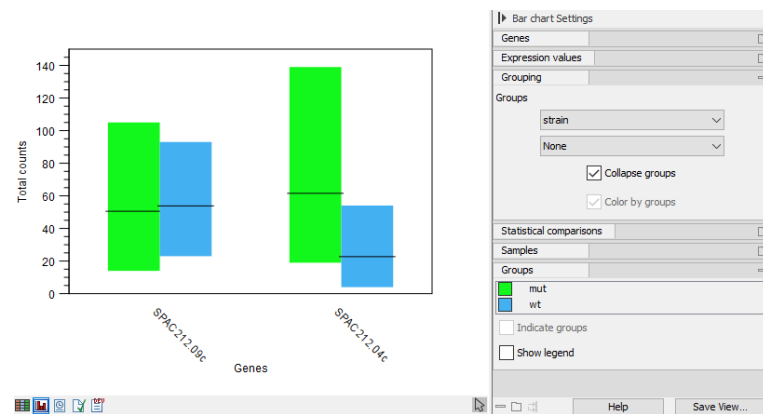


Figure 33.12: Expression browser plot where samples are collapsed to one bar per metadata group

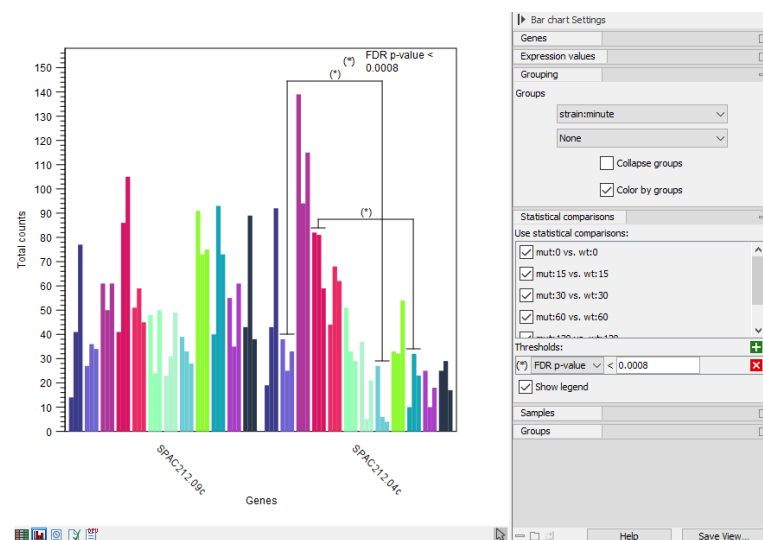


Figure 33.13: Expression browser plot with indication of differentially expressed groups

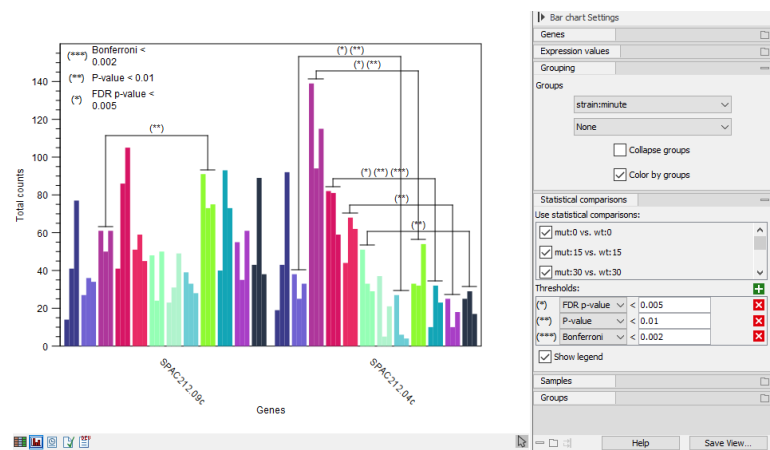


Figure 33.14: Expression browser plot with multiple statistical comparison thresholds

33.3 miRNA analysis

33.3.1 Quantify miRNA

The Quantify miRNA tool counts and annotates miRNAs using miRBase. It maps the reads against the full precursor miRNA and further processes the mapping to produce expression values for:

- Mature miRNAs. Note that the same mature miRNA may be produced from different precursor miRNAs.
- Seeds. Note that the same seed sequence may be found in different mature miRNAs.

The tool performs the mapping using unique search sequences, i.e. collapsed identical reads. This significantly reduces the computational time. The report contains values for both reads and unique search sequences.

The tool will take:

- Trimmed reads (UMI or normal). Reads shorter than 15 bp are ignored.
- A miRBase database. Custom databases that include isoforms of small RNA, such as isopiRNA databases, are not supported.
- Spike-ins (optional): A list of sequences that have been spiked-in. Mapping against this set of sequences will be performed before mapping of the reads against miRBase and other databases. The spike-ins are counted as exact matches and stored in the report for further analysis by the Combined miRNA Report tool.

To run the tool, go to:

Tools | RNA-Seq and Small RNA Analysis | **miRNA Analysis** | **Quantify miRNA**

First select the trimmed reads as in figure 33.15.

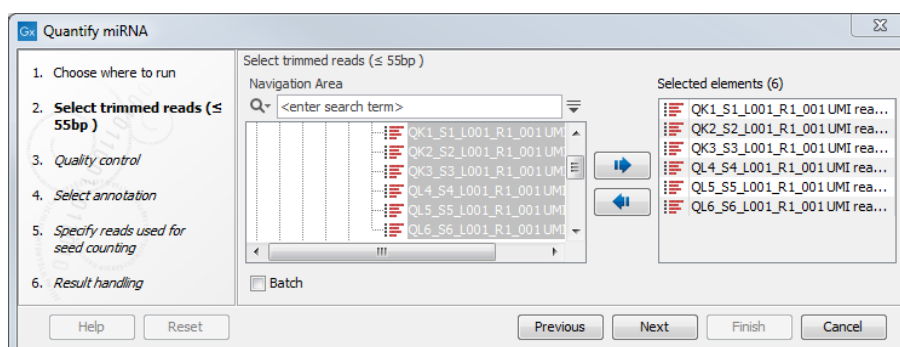


Figure 33.15: Select the reads.

If the sequencing was performed using **Spike-ins** controls, the option "Enable spike-ins" can be enabled in the Quality control dialog (figure 33.16), and a spike-ins file can be specified. You can also change the **Low expression** "Minimum supporting count", i.e., the minimum number of supporting reads for a small RNA to be considered expressed.

In the annotation dialog, several configurations are available.

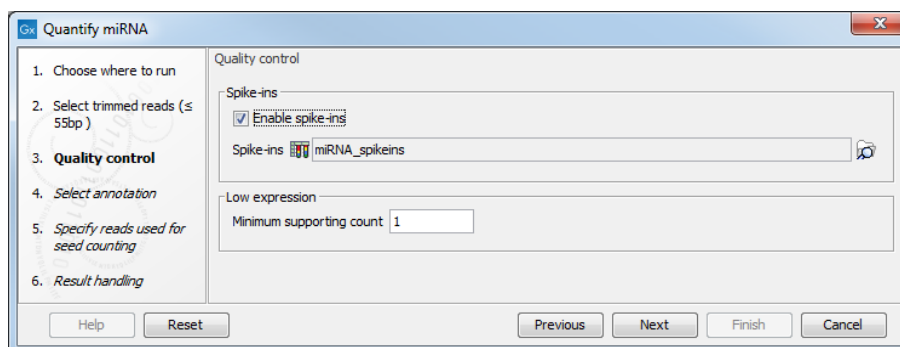


Figure 33.16: Specifying spike-ins is optional, and you can change the threshold under which a small RNA will be considered expressed.

In the **miRBase annotations** section, specify a single reference - miRBase in most cases.

miRBase can be downloaded using the Reference Data Manager under QIAGEN Sets | Reference Data Elements | mirBase (figure 33.17).

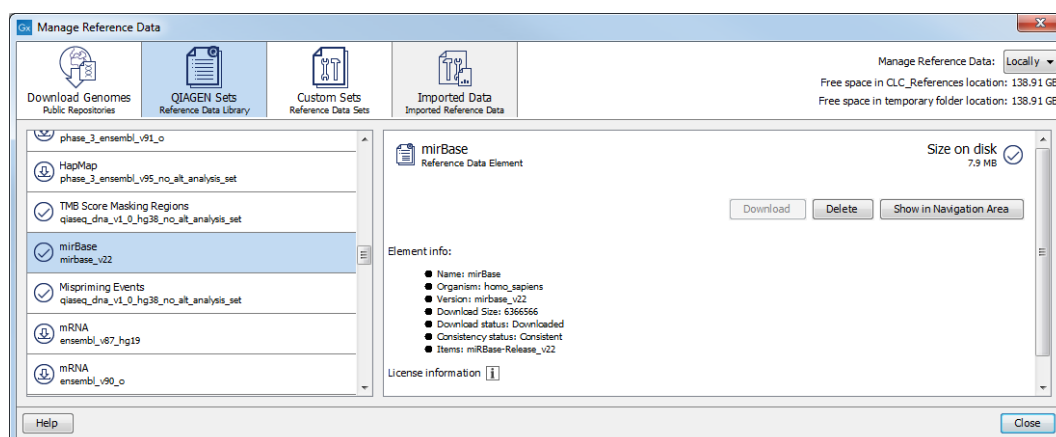


Figure 33.17: Download the latest miRBase database in the Workbench.

You can also import miRBase into the *CLC Genomics Workbench* using Standard Import (📁). The miRBase data file can be downloaded from <ftp://mirbase.org/pub/mirbase/CURRENT/miRNA.dat.gz>. Select **MiRBase (.dat)** in the **Force import as type** menu of the Standard Import dialog. Information about the miRBase dat format is provided in section 1.1.7.

Once miRBase has been selected, click the green plus sign to see the list of species available. It can take a while for all species to load. Species to be used for annotation should be selected using the left and right arrows, and prioritized using the up and down arrows, with the species sequenced always selected as top priority in the list (figure 33.18). The naming of the miRNA will depend on this prioritization.

In addition, it is possible to configure how specific the association between the isomiRs and the reads has to be by allowing mismatches and additional or missing bases upstream and downstream of the isomiR.

In the **Custom databases**, you can optionally add sequence lists with additional smallRNA reference databases, e.g. piRNAs, tRNAs, rRNAs, mRNAs, lncRNAs. An output with quantification against the custom databases can be generated, which can be used for subsequent expression

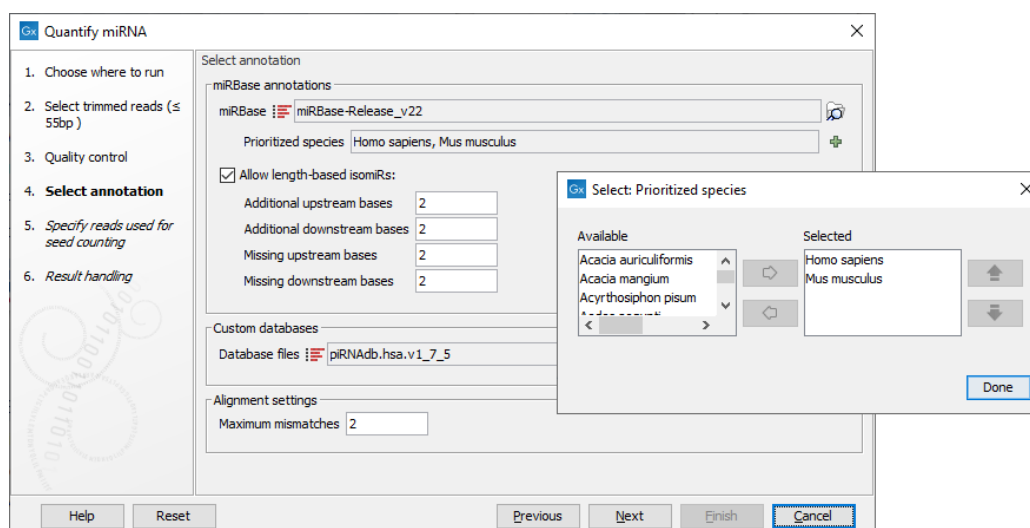


Figure 33.18: Specify and prioritize species to use for annotation, and how stringent the annotation should be.

analyses. Reads count towards the reference to which they map best, regardless of which database (miRBase or custom) the reference is from.

Finally, configure the **Alignment settings** by defining how many "Maximum mismatches" are allowed between the reads and the references, i.e. miRBase and custom databases. Reads matching more than 100 references are ignored.

In the next dialog (figure 33.19), specify the length of the reads used for seed counting. Reads of the specified length, corresponding to the length of mature miRNA (18-25 bp by default, but this parameter can be configured) are used for seed counting. The seed is a 7 nucleotide sequence from positions 2-8 on the mature miRNA. The "Grouped on seed" output table includes a row for every seed that is observed in miRBase together with the expression of the same seed in the sample. In addition, the 20 most highly expressed novel seeds are output in the report.

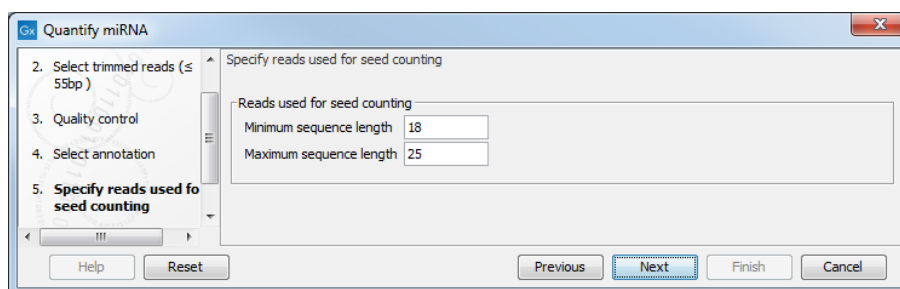


Figure 33.19: This dialog defines the length of the reads that will be merged according to their seed.

The tool will output the following expression tables:

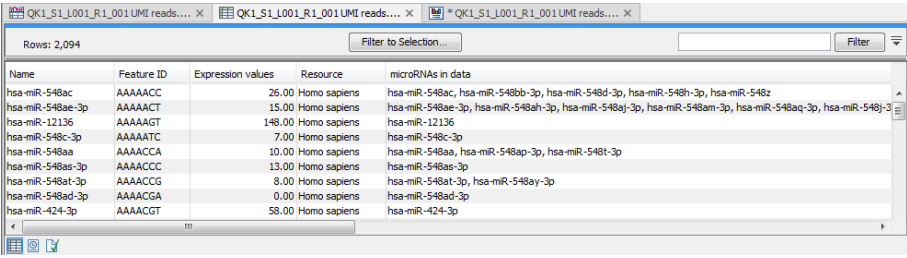
- Grouped on mature, with a row for each mature miRNA in the database
- Grouped on seed, with a row for each seed sequence
- Grouped on custom databases, if one or more custom databases is provided.

The expression tables can be used for subsequent expression analysis tools such as Differential Expression (section 33.6), PCA for RNA-Seq (section 33.5.1) and Create Heat-Map for RNA-Seq (section 33.5.3). In addition, and depending on the options selected in the last dialog, the tool can output a report and a sequence list of reads that could not be mapped to any references. For a detailed description of the outputs from Quantify miRNA see section 33.3.1

Quantify miRNA outputs

Grouped on seed

In this expression table, there is a row for each seed sequence (figure 33.20).



Name	Feature ID	Expression values	Resource	microRNAs in data
hsa-miR-548ac	AAAAACC	26.00	Homo sapiens	hsa-miR-548ac, hsa-miR-548bb-3p, hsa-miR-548d-3p, hsa-miR-548h-3p, hsa-miR-548z
hsa-miR-548ae-3p	AAAAACT	15.00	Homo sapiens	hsa-miR-548ae-3p, hsa-miR-548ah-3p, hsa-miR-548aj-3p, hsa-miR-548am-3p, hsa-miR-548aq-3p, hsa-miR-548b-3p
hsa-miR-12136	AAAAAGT	148.00	Homo sapiens	hsa-miR-12136
hsa-miR-548c-3p	AAAAATC	7.00	Homo sapiens	hsa-miR-548c-3p
hsa-miR-548aa	AAAACCA	10.00	Homo sapiens	hsa-miR-548aa, hsa-miR-548ap-3p, hsa-miR-548t-3p
hsa-miR-548as-3p	AAAACCC	13.00	Homo sapiens	hsa-miR-548as-3p
hsa-miR-548at-3p	AAAACCG	8.00	Homo sapiens	hsa-miR-548at-3p, hsa-miR-548ay-3p
hsa-miR-548ad-3p	AAAACGA	0.00	Homo sapiens	hsa-miR-548ad-3p
hsa-miR-424-3p	AAAACGT	58.00	Homo sapiens	hsa-miR-424-3p

Figure 33.20: Expression table grouped on seed.

This table contains the following information:

- **Name** An example of an expressed mature miRNA that has this seed sequence.
- **Feature ID** The sequence of the miRNA seed
- **Expression value** Counts
- **Resource** The database used for identifying miRNAs. For miRBase the species name will be shown.
- **microRNAs in data** A complete list of expressed mature miRNAs with this seed sequence

Grouped on mature

In this table, there is a row for each mature miRNA in the database, including those for which the expression is zero (figure 33.21). Double click on a row to open a unique reads alignment (seen at the bottom of figure 33.21). Unique reads result from collapsing identical reads into one. The number of reads that are collapsed into a unique read is indicated in parentheses to the right of the miR name of the unique mature read. The alignment shows all possible unique reads that have aligned to a specific miRNA from the database. Mismatches to the mature reference are highlighted in the alignment and recapitulated in their name as explained in section 33.3.1.

This table contains the following information:

- **Feature ID** Sequence of the mature miRNA
- **Identifier** The RNAcentral Accession of the mature miRNA

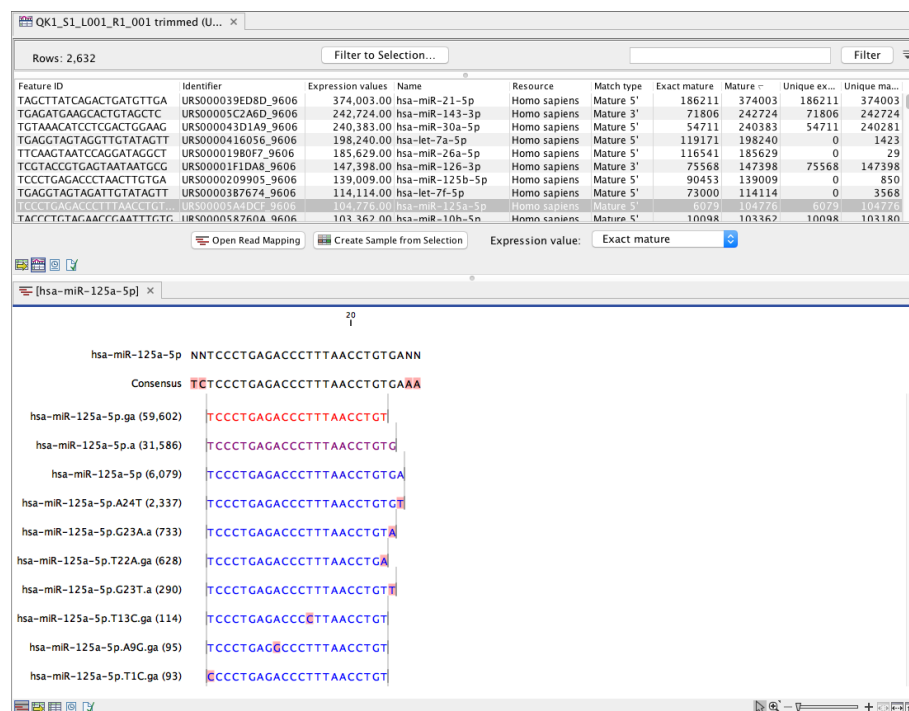


Figure 33.21: Expression table grouped on mature, with a view of a unique reads alignment.

- **Expression value** Counts in the Mature column
- **Name** Name of the annotation sequence
- **Resource** This is the source of the annotation. For miRBase the species name will be shown.
- **Match type** Mature 5' or Mature 3'
- **Exact mature** Number of mature reads that exactly match the miRBase sequence.
- **Mature** Number of all mature reads, i.e., exact and variants
- **Unique exact mature** In cases where one read has several hits, the counts are distributed evenly across the references. The difference between Exact mature and Unique exact mature is that the latter only includes reads that are unique to this reference.
- **Unique mature** Same as above but for all mature, including variants

Grouped on custom database

In this table, there is a row for each mature smallRNA in the database, including those for which the expression is zero (figure 33.22). Double click on a row to open a unique reads alignment (seen at the bottom of figure 33.22). Unique reads result from collapsing identical reads into one. The number of reads that are collapsed into a unique read is indicated in parentheses to the right of the miR name of the unique mature read. The alignment shows all possible unique reads that have aligned to a specific miRNA from the database. As with the table *Grouped on mature*, mismatches to the reference are highlighted in the alignment and recapitulated in their name as explained in section 33.3.1.

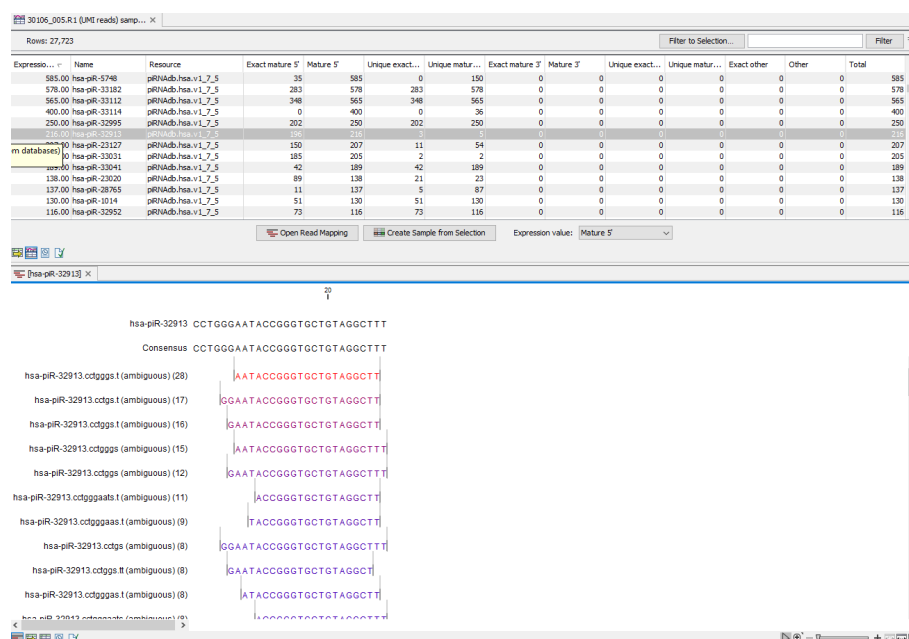


Figure 33.22: Expression table grouped on custom database, with a view of a unique reads alignment.

This table contains the following information:

- **Feature ID** Sequence of the mature miRNA
- **Expression values** Counts in the Mature column
- **Name** Name of the annotation sequence
- **Resource** This is the source of the annotation, usually the name of the custom database input.
- **Exact mature** Number of reads matching to a subsequence of the custom database reference.
- **Mature** Number of reads matching to a subsequence of the custom database reference where mismatches are allowed (within the limit of what was specified in the wizard during configuration)
- **Unique exact mature** In cases where one read has several hits, the counts are distributed evenly across the references. The difference between Exact mature and Unique exact mature is that the latter only includes reads that are unique to this reference.
- **Unique mature** Same as above but for all mature, including variants
- **Exact other** Always 0
- **Other** Always 0
- **Total**

Report

The quantification report contains the following main sections:

- Quantification summary, with information of the number of features that were annotated in the sample.
- Spike-ins, a statistical summary of the reads mapping to the spike-ins (only when spike-ins were enabled).
- Unique search sequences counts, a small RNA reads count distribution.
- Map and Annotate, with Summary, Resources, Unique search sequences, Reads, Read count proportions and Annotations (miRBase).
- Reference sequences, a table with the Top 20 mature sequences, and a table with the Top custom databases sequences when one was provided.
- Seeds report, with tables listing the Top 20 seeds (reference) and Top 20 novel seeds.

It is later possible to combine all miRNA related reports issued for one sample using the Create Combined miRNA Report tool, see section [33.3.3](#).

Naming isomiRs

The names of aligned sequences in mature groups adhere to a naming convention that generates unique names for all isomiRs. This convention is inspired by the discussion available here: https://github.com/miRTop/incubator/blob/master/isomirs/isomir_naming.md

Deletions are in lowercase and there is a suffix s for 5' deletions (figure [33.23](#)):

```
ACGT (ref)
CGT .as
ACG .t
GT .acs
```

Figure 33.23: *Naming of deletions.*

Insertions are in uppercase and there is a suffix s for 5' insertions (figure [33.24](#)):

```
ACGT (ref)
ACGTT .T
ACGTG .G
AACGT .As
ACGTTT .TT
ACGTGT .GT
```

Figure 33.24: *Naming of insertions.*

Note that indels within miRNAs are not supported.

Mutations (SNVs) are indicated with reference symbol, position and new symbol. Consecutive mutations will not be merged into MNVs. The position is relative to the reference, so preceding (5') indels will not offset it (figure 33.25):

```
ACGT (ref)
ATGT .C2T
ATAT .C2T.G3A
AATGT .As.C2T (and not C3T)
TGT .as.C2T (and not C1T)
```

Figure 33.25: *Naming of mutations.*

Deletions followed by insertions will be annotated as shown in figure 33.26:

```
ACGT (ref)
TCGT .A1T (and not .as.Tes)
ACGA .T4A (and not .t.Ae)
```

Figure 33.26: *Naming of deletions followed by insertions.*

If a sequence maps to multiple miRBase entries or to multiple entries in a custom database, we will add the suffix 'ambiguous' to its name. This can happen when multiple species are selected, as they will often share the same miRBase (or other reference) sequence, or when a read does not map perfectly to any miRBase entry, but is close to two or more entries, distinguished by just one SNV, for example.

33.3.2 Annotate with RNACentral Accession Numbers

To be able to link entries in the Grouped by Mature expression table to entries in the Gene Ontology database, both must have identical names or identifiers. The human GO annotations distributed in the QIAseq Small RNA Reference Data Set identify miRNAs by RNACentral Accession numbers. The Annotate with RNACentral Accession Numbers adds these identifiers to the Grouped by Mature expression table.

To run the tool, go to:

Tools | RNA-Seq and Small RNA Analysis  **miRNA Analysis**  **Annotate with RNACentral Accession Numbers** 

First select one small RNA sample - or several samples when using the batch option. In the next dialog (figure 33.27), specify the miRBase and the RNACentral table linking mappings from miRBase to RNACentral identifiers. This table is available from:

ftp://ftp.ebi.ac.uk/pub/databases/RNACentral/releases/10.0/id_mapping/database_mappings/mirbase.tsv

The tool will add identifiers to mature miRNAs that can be used to match Gene Ontology identifiers. These will be passed on to the comparison table when doing differential expressions, so that they in turn can be passed on to Gene Set Test (section 33.6.7) against gene ontology.

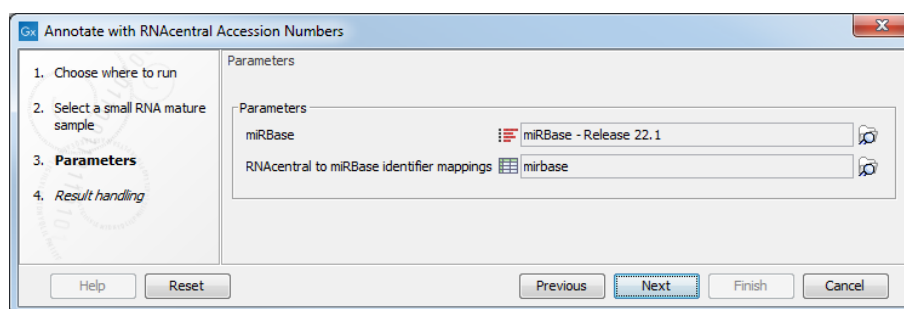


Figure 33.27: Specify the miRBase database and RNACentral table.

33.3.3 Create Combined miRNA Report

Create Combined miRNA Report combines all reports from miRNA related tools (including the Biomedical Genomics Analysis tool Create UMI Reads for miRNA https://resources.qiagenbioinformatics.com/manuals/biomedicalgenomicsanalysis/current/index.php?manual=Create_UMI_Reads_miRNA.html) across multiple samples into a single report. This is to provide a better overview of samples in the experiment and to more easily perform quality control. The tool should be run on samples that are created using the same parameters. Samples with the same name will be merged.

To start the tool, go to:

Tools | RNA-Seq and Small RNA Analysis (📁) | **miRNA Analysis** (📁) | **Create Combined miRNA Report** (🔗)

In the first dialog, select the reports to be merged (at least 2). Click Next to choose whether you wish to use short aliases (*S1, S2, etc...) or the full sample names in the headers of the tables. This is merely a question of readability of the resulting report.

From the **UMI reads report**, the UMI statistics section is extracted. From the **miRNA Quantification Report** spike-ins are consolidated in a single correlation table (figure 33.28). The color in the table represents the strength of the correlation. Below 0.95, the table entry is colored red; Between 0.95 and 0.99, the table entry will be green. A correlation higher than 0.99 will lead to a gray entry, as this result can be considered suspiciously high.

When combining different samples report, a table will compile the top 20 mature sequences (figure 33.29), and another table for the Top non-miRNA sequences when one was provided.

In the Seeds report section, tables recapitulate the top 20 seeds (when any) and top 20 novel seeds with counts for the different samples (as seen in figure 33.30).

33.3.4 Extract IsomiR Counts

The **Extract IsomiR Counts** takes as input one or more "grouped on mature" expression tables, output by the **Quantify miRNA** tool. It extracts IsomiR composition and count information from each underlying miRNA alignment and produces tables containing this information.

If only a subset of miRNAs are of interest, we recommend creating a sample containing only the those prior to running this tool. To do this, open the expression table, select the miRNAs of interest, and click on the **Create Sample from Selection** (📁) button.

4 Spike-in

4.1 Spike-in QC

Sample	% spike-in
S1	70.41
S2	65.357
S3	74.422

~1-3% is the expected range when spike-ins are used.

4.2 Spike-in correlation (R²)

Sample	S1	S2	S3
S1		0.98	0.97
S2	0.98		0.98
S3	0.97	0.98	

Color legend:
 Gray: >= 0.99
 Green: >= 0.96
 Yellow: >= 0.9
 Red: < 0.9

4.3 Excluded Spike-ins

Sample	Excluded Spike-ins
S1	
S2	
S3	

The table lists, for each sample, spike-ins which were not expressed sufficiently to be included when calculating correlations for the sample. Other spike-ins might have been excluded in some correlation calculations if they were not expressed sufficiently in any of the two samples being correlated.

Figure 33.28: The combined report includes a table of the Top 20 mature sequences for all samples. A question mark ? indicates when a feature is not among the top 20 mature sequences from a particular sample.

5.1 Top 20 mature sequences

Mature	Species	Average %	S1	S2	S3	S4	S5	S6
hsa-miR-486-5p	Homo sapiens/Mus musculus	2.63%	40,735	50,289	20,338	2,940 ?		?
hsa-miR-16-5p	Homo sapiens/Mus musculus	2.39%	32,699	66,983	12,100	1,294 ?		?
hsa-let-7b-5p	Homo sapiens/Mus musculus	2.10%	30,468	46,988	14,240	2,502 ?		?
hsa-miR-103a-3p	Homo sapiens/Mus musculus	0.37%	5,271	20,243	?	?	?	?
hsa-miR-196b-5p	Homo sapiens/Mus musculus	0.32%	?	?	?	?	28,050	19,127
hsa-miR-93-5p	Homo sapiens/Mus musculus	0.30%	3,784	13,249	1,634	?	?	?
hsa-miR-26a-5p	Homo sapiens/Mus musculus	0.30%	4,027	16,420	?	?	?	?

Figure 33.29: The combined report includes a table of the Top 20 mature sequences for all samples. A question mark ? indicates when a feature is not among the top 20 mature sequences from a particular sample.

If multiple expression tables are provided as input, each of the resulting isomiR counts tables will list all isomiRs that are present in any of the input samples. These result tables can be used as input to **Differential Expression for RNA-Seq** or **Differential Expression in Two Groups** to identify differentially expressed isomiRs between samples - see section 33.6. Note that isomiR counts tables generated from different executions of **Extract IsomiR Counts** cannot be used together in a differential expression analysis.

Names are not always unique for isomiR sequences. If the same sequence have multiple different names, **Extract IsomiR Counts** write the name as a comma separated list of the different names used in the input.

To run **Extract IsomiR Counts**, go to:

Tools | RNA-Seq and Small RNA Analysis  **miRNA Analysis**  **Extract IsomiR Counts** 

6.2 Top 20 novel seeds

Seeds	Average %	S1	S2	S3	S4	S5	S6
GAGGTAA	0.02%	138	237	101	52	98	138
GGTAATA	0.01%	55	?	57	52	?	?
TTGGAAT	0.01%	86	137	79	23	?	?
TTGGAAT	0.01%	76	110	71	25	?	?
TCGAATC	0.01%	76	106	84	20	?	?
TCGAATC	0.01%	61	62	55	21	?	?
CCACAGG	0.01%	37	64	39	19	?	?
GGGATGT	0.00%	?	74	34	14	?	?
CGATTCC	0.00%	56	62	49	?	?	?
TGGATTT	0.00%	?	?	46	14	?	?
AGGATTG	0.00%	?	?	33	16	?	?
TGGCAAT	0.00%	?	?	?	?	203	283
TTGGAAT	0.00%	45	72	36	?	?	?
GGTGTAG	0.00%	?	?	38	13	?	?
AAAGCTA	0.00%	?	?	?	17	?	?
TAATAGG	0.00%	?	?	?	16	?	?
GCTAAMC	0.00%	?	?	?	?	173	166
TTGACCT	0.00%	?	?	?	15	?	?
GTTAAGT	0.00%	?	?	?	15	?	?
GTTTCAT	0.00%	38	?	41	?	?	?

Figure 33.30: The combined report includes a table of the Top 20 novel seeds for all samples. A question mark ? indicates when a feature is not among the top 20 novel seeds from a particular sample.

Select one or more "grouped on mature" expression tables as input. Information from all miRNAs in this table will be extracted.

Extract IsomiR Counts output

Extract IsomiR Counts outputs a table with four columns:

- **Name** The name of the IsomiR
- **Sequence** The sequence of the IsomiR
- **Count** The number of times the sequence was found in the sample. If UMI technology was used, this refers to the number of UMI reads the sequence was found in.
- **Ambiguous** A check in this column indicates that the sequence mapped to multiple entries in miRBase or multiple entries in a custom database. See section 33.3.1 for further details.

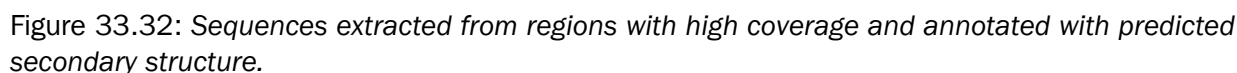
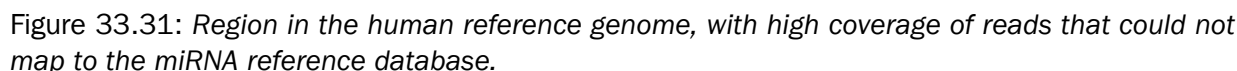
33.3.5 Explore Novel miRNAs

The Quantify miRNA tool only finds and quantifies the expression of known miRNAs. Here we will describe an approach to exploring novel miRNAs in the data.

First, take the "Unmapped reads" output from **Quantify miRNA** and use these reads as input to **Map Reads to Reference** (📄), see section 30.1. To home in on sequences with a decent amount of expression and avoid sequences that map as a result of sequencing errors, we can use the tools **Create Mapping Graph** (📄) (see section 27.9.2) and **Identify Graph Threshold Areas** (📈) (see section 27.9.3) to only consider sequences expressed above a user defined level.

Next, to extract the sequence in the regions of high coverage use the tool **Extract Annotations** (📄) (see section 37.1).

Finally, the tool **Predict Secondary Structure** (🌀) (see section 26.1) can be used to check if the structure matches the expected hairpin structure of miRNA precursors.



33.4.1 RNA-Seq Analysis

The RNA-Seq analysis is done in several steps: First, all annotated transcripts are extracted (using an *mRNA* track). If there are several annotated splice variants, they are all extracted.

This is a simple gene with three exons and two splice variants. The transcripts are extracted as

shown in figure 33.34.

```

Splice variant 1
GGACAGTGTCTGGAGATCCGCTCGCGCGCGGAAGTACTGCAAAATACAACGTGATCACATTCTTCCGAG
Splice variant 2
GGACAGTGTCTGGAGATCCGCTCGCGCGCGGAAGGTTATGAGAAGACAGATGATGTTTCAGAGAAGACCT

```

Figure 33.34: All the exon-exon junctions are joined in the extracted transcript.

Next, the reads are mapped against all the transcripts, and to the whole genome. For more information about the read mapper, see section 30.1.

From this mapping, the reads are categorized and assigned to the transcripts using the EM estimation algorithm, and expression values for each gene are obtained by summing the transcript counts belonging to the gene.

Reads and reference settings

To start the RNA-Seq analysis, go to:

Tools | RNA-Seq and Small RNA Analysis (📁) | **RNA-Seq Tools** (📁) | **RNA-Seq Analysis** (🇺🇸)

This opens a dialog where you select the **sequencing reads**. Note that you need to import the sequencing data into the Workbench before it can be used for analysis. Importing read data is described in section 7.3.

If you have several samples that you wish to analyze independently and compare afterwards, you can run the analysis in batch mode, see section 12.3.

Click **Next** when the sequencing data are listed in the right-hand side of the dialog.

You are now presented with the dialog shown in figure 33.35.

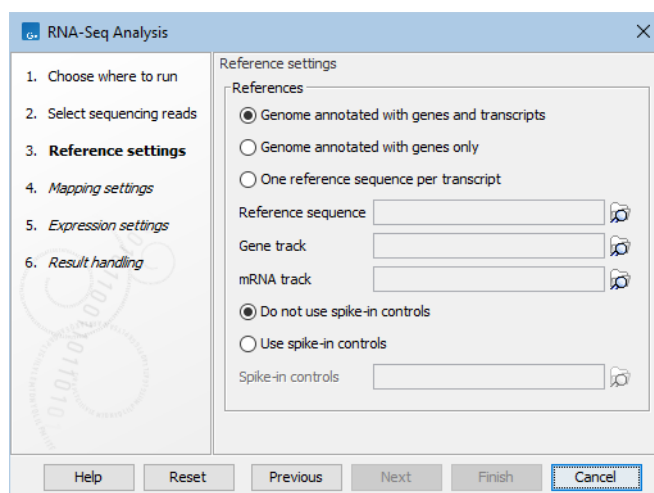


Figure 33.35: Defining a reference genome for RNA-Seq.

At the top, there are three options concerning how the reference sequences are annotated.

- **Genome annotated with genes and transcripts.** This option should be used when both

gene and mRNA annotations are available. When this option is enabled, the EM will distribute the reads over the transcripts. Gene counts are then obtained by summing over the (EM-distributed) transcript counts. The mRNA annotations are used to define how the transcripts are spliced (as shown in figure 33.33). This option should be used for **Eukaryotes** since it is the only option where splicing is taken into account. Note that genes and transcripts are linked by name only (not by position, ID etc).

When this option is selected, both a *Gene* and an *mRNA* track should be provided in the boxes below. Annotated reference genomes can be obtained in various ways:

- Directly downloaded as tracks using the Reference Data Manager (see section 11.1).
- Imported as tracks from fasta and gff/gtf files (see section 7.2).
- Imported from Genbank or EMBL files and converted to tracks (see section 27.8).
- Downloaded from Genbank (see section 10.1) and converted to tracks (see section 27.8).

When using this option, Expression values, RPKM and TPM are calculated based on the lengths of the transcripts provided by the mRNA track. If a gene's transcript annotation is absent from the mRNA track, all values will be set to 0 unless the option "Calculate expression for genes without transcript" is checked in a later dialog.

- **Genome annotated with genes only.** This option should be used for **Prokaryotes** where transcripts are not spliced. When this option is selected, a *Gene* track should be provided in the box below. The data can be obtained in the same ways as described above.

When using this option, Expression values, RPKM and TPM are calculated based on the lengths of the genes provided by the *Genes* track.

- **One reference sequence per transcript.** This option is suitable for situations where the reference is a list of sequences. Each sequence in the list will be treated as a "transcript" and expression values are calculated for each sequence. This option is most often used if the reference is a product of a *de novo* assembly of RNA-Seq data. It is also a suitable option for references where genes are particularly close to each other or clustered in operon structures (see section 33.4.1). When this option is selected, only the reference sequence should be provided, either as a sequence track or a sequence list. Expression values, RPKM and TPM are calculated based on the lengths of sequences from the sequence track or sequence list.

Tightly packed genes and genes in operons

For annotated references containing genes located very close to each other (including operon structures) only reads mapping completely within a gene's boundaries will be counted towards the expression value for that gene. If any part of a read maps outside a given gene's boundaries, then it will be considered intergenic and will not be counted towards the expression value. For tightly packed genes, especially in cases where non-coding 5' regions are not included in the gene annotation, this can be too conservative: if there are short genes, where the read length exceeds the gene length in some cases, then some granularity may be lost. That is, reads mapping to short genes might not be counted at all.

If this situation arises in your data, you can do the following:

- Use the option "One reference per transcript" in the "Select reference" wizard, and input a list of transcript sequences instead of a track. A list of sequences can be generated from a mRNA track (or a gene track for bacteria if no mRNA track is available) using the Extract Annotations tool (see section 37.1).
- In cases where the input reads are paired-end, choose the option "Count paired reads as two" in the Expression settings dialog. This will ensure that each read of the pair is counted towards the expression of the gene with which it overlaps, (by default, paired reads that map to different genes are not counted).

This strategy is equivalent to the option "Map to gene regions only (fast)" option that was available in the workbench released before February 2017.

At the bottom of the dialog you can choose between these two options:

- **Do not use spike-in controls.**
- **Use spike-in controls.** In this case, you can provide a spike-in control file in the field situated at the bottom of the dialog window. Make sure you remember to check the option to output a report in the last wizard step, as the report is the only place where the spike-in controls results will be available. During analysis, the spike-in data is added to the references. However, all traces of having used spike-ins are removed from the output tracks.

If spike-ins have been used, the quality control results are shown in the output report. So when using spike-in, make sure that the option to output a report is checked.

To learn how to import spike-in control files, see section 7.5.

Mapping settings

When the reference has been defined, click **Next** and you are presented with the dialog shown in figure 33.36.

The mapping parameters are identical to those applying to **Map Reads to Reference**, as the underlying mapping is performed in the same way. For a description of the parameters, please see section 30.1.3.

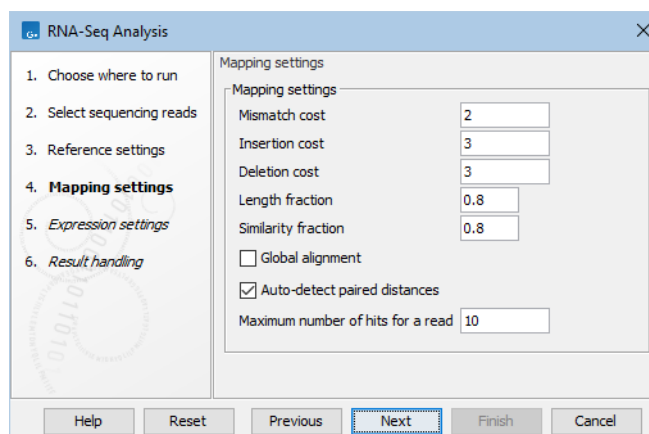


Figure 33.36: Defining mapping parameters for RNA-Seq.

For the estimation of paired reads distances, RNA-Seq uses the transcript level reference sequence information. This means that introns are not included in the distance measurement. The paired distance measurement will only include transcript sequence, reflecting the true nature of the sequence on which the paired reads were produced.

In addition to the generic mapping parameters, two RNA-Seq specific parameters can be set:

- **Maximum number of hits for a read.** A read that matches equally well to more *distinct* places in the reference sequence than the 'Maximum number of hits for a read' specified will not be mapped. If a read matches to multiple distinct places, but less than or equal to the specified maximum number, it will be assigned to one of these places by the EM algorithm (see section 33.4.1). Note that to favor accurate expression analysis, it is recommended to have this value set to 10 or more.

Concept of hits and distinct places in the reference

The definition of a *distinct* place in the reference sequence is complicated. We are describing here example cases where the option "Genome annotated with genes and transcripts" is selected in the previous "Reference settings" step, meaning that reads are aligned to genes and transcripts.

- In an example case where 2 genes are overlapping, a read will count as one hit because it corresponds to the same reference sequence location. This read will be assigned to one of the genes by the EM algorithm.
- In an example case where a gene has 10 transcripts and 11 exons, and all transcripts have exon 1 plus one of the exons 2 to 11. Exon 1 is thus represented 11 times in the references (once for the gene region and once for each of the 10 transcripts). Reads that match to exon 1 will thus match to 11 of the extracted references. However, when the mappings are considered in the coordinates of the main reference genome, it becomes evident that the 11 match places are not distinct but in fact identical. In this case this will just count as one hit.
- In a more complicated example, a gene has different splicing, for example transcripts with longer versions of an exon than the others. In this case you may have reads that may either

be mapped entirely within the long version of the exon, or across the exon-exon boundary of one of the transcripts with the short version of the exon. These reads are ambiguously mapped (they appear in yellow in a track view), and count as as many hits as different ways they map to the reference. Setting the 'Maximum number of hits for a read' parameter too low could leave these reads unmapped, eliminating the evidence for the expression of the gene to which they mapped.

The EM estimation algorithm

The EM estimation algorithm is inspired by the RSEM and eXpress methods. It iteratively estimates the abundance of transcripts/genes, and assigns reads to transcripts/genes according to these abundances.

To illustrate the interpretation of 'abundance', consider the following two examples:

In the first example, we have a gene with two transcripts, where one transcript is twice as long as the other (figure 33.37). This longer transcript is also twice as abundant, meaning that in the exon common to both transcripts two of the three reads come from the longer transcript, and the final read comes from the shorter transcript. The longer transcript has a second exon which also generates two reads. We see then that the longer transcript is twice as abundant, but because it is twice as long it generates four times as many reads.

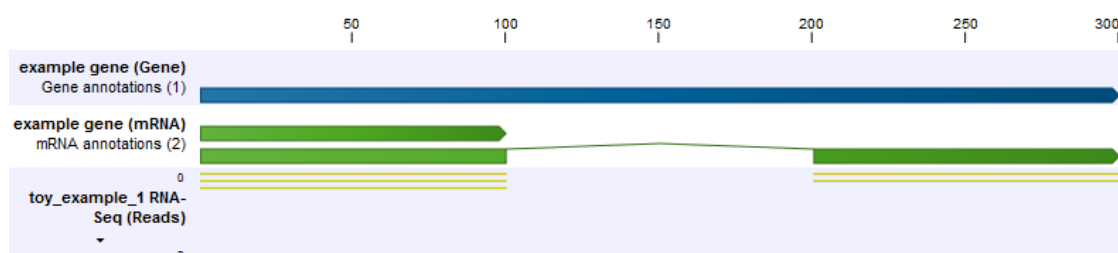


Figure 33.37: *The longer transcript has twice the abundance, but four times the number of reads as the shorter transcript.*

In the second example, the set up is the same, but now the shorter transcript is twice as abundant as the longer transcript (figure 33.38). Because the longer transcript is twice as long, there are equal numbers of reads from each transcript.

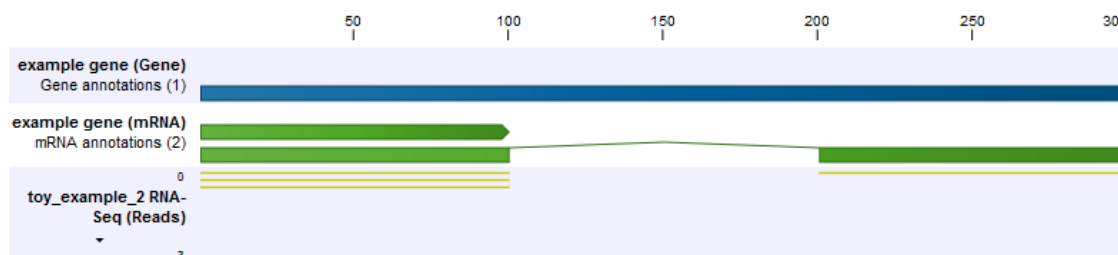


Figure 33.38: *The longer transcript has half the abundance, but the same number of reads as the shorter transcript.*

To estimate the transcript abundances, we carry out an expectation-maximization procedure. Before explaining the procedure, first we define the concept of a mapping. A mapping is a set of transcripts, to which a read may map. In the above examples, some reads have the mapping $a_1 = \{t_1, t_2\}$ (these are non-uniquely mapping reads), and some reads have the mapping

$a_2 = \{t_2\}$ (these are 'uniquely' mapping reads). In both examples, the count of mapping a_1 is 3, because there are 3 shared reads between the transcripts. The count of mapping a_2 is 2 in the first example, and 1 in the second example.

The expectation-maximization algorithm proceeds as follows:

1. The transcript abundances are initialized to the uniform distribution, i.e. at the start all transcripts are assumed to be equally expressed.
2. Expectation step: the current (assumed) transcript abundances are used to calculate the expected count of each transcript, i.e. the number of reads we expect should be assigned to the given transcript. This is done by looping over all mappings that include the given transcript, and assigning a proportion of the total count of that mapping to the transcript. The proportion corresponds to the proportion of the total transcript abundance in the mapping that is due to the target.
3. Maximization step: the currently assigned counts of each transcript are used to re-compute the transcript abundances. This is done by looping over all targets, and for each target, dividing the proportion of currently assigned counts for the transcript (=total counts for transcript/total number of reads) by the target length. This is necessary because longer transcripts are expected to generate proportionally more reads.
4. Repeat from step 2 until convergence.

Below, we illustrate how the expectation-maximization algorithm converges to the expected abundances for the above two examples.

Example 1:

Initially: transcript 2 abundance = 0.50, count: 0.00, transcript 1 abundance = 0.50, count: 0.00
 After 1 round: transcript 2 abundance = 0.54, count: 3.50, transcript 1 abundance = 0.46, count: 1.50
 After 2 rounds: transcript 2 abundance = 0.57, count: 3.62, transcript 1 abundance = 0.43, count: 1.38
 After 3 rounds: transcript 2 abundance = 0.59, count: 3.70, transcript 1 abundance = 0.41, count: 1.30
 After 4 rounds: transcript 2 abundance = 0.60, count: 3.76, transcript 1 abundance = 0.40, count: 1.24
 After 5 rounds: transcript 2 abundance = 0.62, count: 3.81, transcript 1 abundance = 0.38, count: 1.19
 After 6 rounds: transcript 2 abundance = 0.62, count: 3.85, transcript 1 abundance = 0.38, count: 1.15
 After 7 rounds: transcript 2 abundance = 0.63, count: 3.87, transcript 1 abundance = 0.37, count: 1.13
 After 8 rounds: transcript 2 abundance = 0.64, count: 3.90, transcript 1 abundance = 0.36, count: 1.10
 After 9 rounds: transcript 2 abundance = 0.64, count: 3.92, transcript 1 abundance = 0.36, count: 1.08
 After 10 rounds: transcript 2 abundance = 0.65, count: 3.93, transcript 1 abundance = 0.35, count: 1.07
 After 11 rounds: transcript 2 abundance = 0.65, count: 3.94, transcript 1 abundance = 0.35, count: 1.06
 After 12 rounds: transcript 2 abundance = 0.65, count: 3.95, transcript 1 abundance = 0.35, count: 1.05
 After 13 rounds: transcript 2 abundance = 0.66, count: 3.96, transcript 1 abundance = 0.34, count: 1.04
 After 14 rounds: transcript 2 abundance = 0.66, count: 3.97, transcript 1 abundance = 0.34, count: 1.03
 After 15 rounds: transcript 2 abundance = 0.66, count: 3.97, transcript 1 abundance = 0.34, count: 1.03

Example 2:

Initially: transcript 2 abundance = 0.50, count: 0.00, transcript 1 abundance = 0.50, count: 0.00
 After 1 round: transcript 2 abundance = 0.45, count: 2.50, transcript 1 abundance = 0.55, count: 1.50
 After 2 rounds: transcript 2 abundance = 0.42, count: 2.36, transcript 1 abundance = 0.58, count: 1.64
 After 3 rounds: transcript 2 abundance = 0.39, count: 2.26, transcript 1 abundance = 0.61, count: 1.74
 After 4 rounds: transcript 2 abundance = 0.37, count: 2.18, transcript 1 abundance = 0.63, count: 1.82
 After 5 rounds: transcript 2 abundance = 0.36, count: 2.12, transcript 1 abundance = 0.64, count: 1.88
 After 6 rounds: transcript 2 abundance = 0.35, count: 2.08, transcript 1 abundance = 0.65, count: 1.92
 After 7 rounds: transcript 2 abundance = 0.35, count: 2.06, transcript 1 abundance = 0.65, count: 1.94
 After 8 rounds: transcript 2 abundance = 0.34, count: 2.04, transcript 1 abundance = 0.66, count: 1.96
 After 9 rounds: transcript 2 abundance = 0.34, count: 2.03, transcript 1 abundance = 0.66, count: 1.97
 After 10 rounds: transcript 2 abundance = 0.34, count: 2.02, transcript 1 abundance = 0.66, count: 1.98
 After 11 rounds: transcript 2 abundance = 0.34, count: 2.01, transcript 1 abundance = 0.66, count: 1.99

After 12 rounds: transcript 2 abundance = 0.34, count: 2.01, transcript 1 abundance = 0.66, count: 1.99
 After 13 rounds: transcript 2 abundance = 0.33, count: 2.01, transcript 1 abundance = 0.67, count: 1.99
 After 14 rounds: transcript 2 abundance = 0.33, count: 2.00, transcript 1 abundance = 0.67, count: 2.00
 After 15 rounds: transcript 2 abundance = 0.33, count: 2.00, transcript 1 abundance = 0.67, count: 2.00

Once the algorithm has converged, every non-uniquely mapping read is assigned randomly to a particular transcript according to the abundances of transcripts within the same mapping. The total transcript reads column reflects these assignments. The RPKM and TPM values are then computed from the counts assigned to each transcript.

Expression settings

When the reference has been defined, click **Next** and you are presented with the dialog shown in figure 33.39.

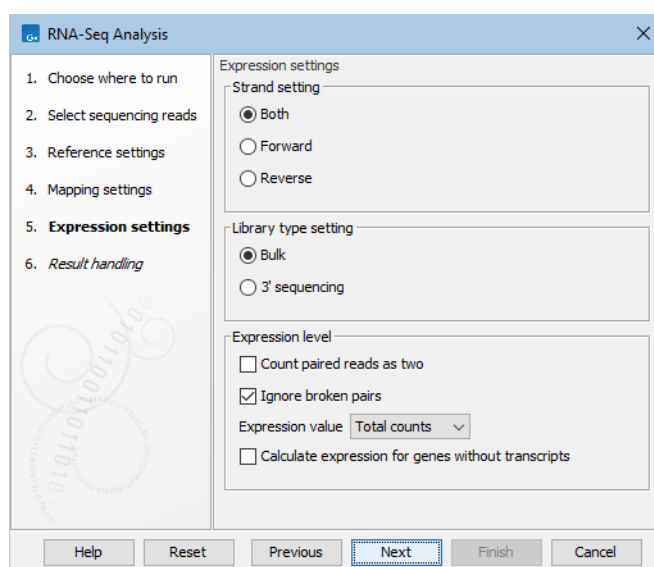


Figure 33.39: Defining how expression values should be calculated.

These parameters determine the way expression values are counted.

Strand setting

- **Both.** Reads are mapped both in the same and reversed orientation as the transcript from which they originate. This is the default.
- **Forward.** Reads are mapped in the same orientation as the transcript from which they originate.
- **Reversed.** Reads are mapped in the reverse orientation as the transcript from which they originate.

If a strand specific protocol for read generation has been used, the user should choose the corresponding setting. This allows assignment of the reads to the right gene in cases where

overlapping genes are located on different strands. Without the strand-specific protocol, this would not be possible (see [Parkhomchuk et al., 2009]). Note that when not running RNA-Seq with 'Both', only pairs in forward-reverse orientation are used, meaning that mate pairs are not supported.

Library type setting

- **Bulk.** Reads are expected to be uniformly distributed across the full length of transcript. This is the default.
- **3' sequencing.** Reads are expected to be biased towards the 3' end of transcripts. Then:
 - Report quality control is tailored for low input 3' sequencing applications.
 - No TE tracks are produced because the EM algorithm requirement for uniform coverage along transcript bodies is not fulfilled.
 - TPM (Transcripts per million) is calculated as (exon reads in gene) / (total exon reads) x 1 million. This is because, in the absence of fragmentation, each read corresponds to a sequenced transcript.
 - RPKM is set equal to TPM, which preserves the expected property that RPKM is proportional to TPM. This is because the standard definition of RPKM normalizes by the length of the transcript that generates each read, and it is often not possible to uniquely identify a transcript based on the 3' end.
 - When analyzing reads that have been annotated with and/or grouped by UMIs by tools of the Biomedical Genomics Analysis plugin:
 - * Single end reads are grouped to UMIs if they map to the same gene and have the same UMI sequence. This is done even if the reads have previously been grouped, e.g. by **Create UMI Reads from Reads**. Thus, if UMI reads are given as input, they might be additionally grouped to fewer but larger UMI reads.
 - * Expression values in the GE track are based on the number of distinct UMIs for each gene, rather than the number of reads.
 - * The "Fragment statistics" section of the RNA-Seq report includes both the number of distinct UMI fragment counts as well as raw read fragment counts. The "Distribution of biotypes" section of RNA-Seq report is based on the number of distinct UMIs for each gene. Other values in the report are as described in section 33.4.1.

Paired reads counting scheme

The *CLC Genomics Workbench* supports the direct use of paired data for RNA-Seq. A combination of single reads and paired reads can also be used. There are two major advantages of using paired data:

- Since the mapped reads span a larger portion of the reference, there will be fewer non-specifically mapped reads. This means that generally there is a greater accuracy in the expression values.

- This in turn means that there is a greater chance of accurately measuring the expression of transcript splice variants. As single reads (especially from the short reads platforms) typically only span one or two exons, many cases will occur where expression of splice variants sharing the same exons cannot be determined accurately. With paired reads, more combinations of exons will be identified as being unique for a particular splice variant.¹

You can read more about how paired data are imported and handled in section [7.3.10](#).

When counting the mapped reads to generate expression values, the *CLC Genomics Workbench* needs to be told how to handle the counting of paired reads that map as

- an intact pair;
- a broken pair, when the reads map outside the estimated pair distance, map in the wrong orientation, or only one of the reads of the pair maps.

Single mapped reads are always given a count of one.

The default behavior of the *CLC Genomics Workbench* is to count fragments (FPKM) rather than individual reads for intact pairs. That is, an intact pair is given a count of one ('Count paired reads as two' is not checked). Neither member of a broken pair is counted ('Ignore broken pairs' is checked). The reasoning is that when reads map as a broken pair, it is an indication that something is not right. For example, perhaps the transcripts are not represented correctly on the reference or there are errors in the data. In general, more confidence can be placed on an intact pair representing transcription within the sample. If a combination of paired and single reads are input into the analysis, then single reads that map are given a count of one. This is different from reads input into the analysis as part of a pair, but where their partner did not map.

In some situations it may be too strict to disregard broken pairs. This could be the case where there is a high degree of variation in the sample compared to the reference or where the reference lacks comprehensive transcript annotations. By checking 'Count paired reads as two', mapped 'reads' (RPKM) rather than mapped 'fragments' (FPKM) are counted. The two reads in an intact pair are each counted as one mapped read (so an intact pair contributes with a total count of two), and mapped members of broken pairs will each get a count of one ('Ignore broken pairs' is disabled).

For specific protocols generating chimeric reads, it is desirable to count intact pairs as one ('Count paired reads as two' is not checked) and to also count the mapped members of broken pairs ('Ignore broken pairs' is not checked).

Note that if not using the default behavior, the value does not correctly represent the abundance of fragments being sequenced, since the two reads of a pair derive from the same fragment and can increase the value by two, whereas a fragment sequenced with single reads only give rise to one read and increases the value by one.

Regardless of the chosen option, the value will be given in a column called "RPKM" for all subsequent analysis.

Expression value

¹Note that the *CLC Genomics Workbench* only calculates the expression of the transcripts already annotated on the reference.

Please note that reads that map outside genes are counted as intergenic hits only and thus do not contribute to the expression values. If a read maps equally well to a gene and to an inter-genic region, the read will be placed in the gene.

The expression values are created on two levels as two separate result files: one for genes and one for transcripts (if the "Genome annotated with genes and transcripts" is selected in figure 33.35). The content of the result files is described in section 33.4.1.

The **Expression value** parameter describes how expression per gene or transcript can be defined in different ways on both levels:

- **Total counts.** When the reference is annotated with genes only, this value is the total number of reads mapped to the gene. For un-annotated references, this value is the total number of reads mapped to the reference sequence. For references annotated with transcripts and genes, the value reported for each gene is the number of reads that map to the exons of that gene. The value reported per transcript is the total number of reads mapped to the transcript.
- **Unique counts.** This is similar to the above, except only reads that are uniquely mapped are counted (read more about the distribution of non-specific matches in section 33.4.1).
- **TPM.** (Transcripts per million). This is computed as $\frac{RPKM \cdot 10^6}{\sum RPKM}$, where the sum is over the RPKM values of all genes/transcripts.
- **RPKM.** This is a normalized form of the "Total counts" option (see more in section 33.4.1).

Please note that all values are present in the output. The choice of expression value only affects how Expression Tracks are visualized in the track view but the results will not be affected by this choice as the most appropriate expression value is automatically selected for the analysis being performed: for detection of differential expression this is the "Total counts" value, and for the other tools this is a normalized and transformed version of the "Total counts" as described below.

Calculate expression for genes without transcripts

For genes without annotated transcripts, the RPKM cannot be calculated since the total length of all exons is needed. By checking the **Calculate expression for genes without transcripts**, the length of the gene will be used in place of an "exon length". If the option is not checked, there will be no RPKM value reported for those genes.

Definition of RPKM RPKM, Reads Per Kilobase of exon model per Million mapped reads, is defined in this way [Mortazavi et al., 2008]:

$$RPKM = \frac{\text{total exon reads}}{\text{mapped reads(millions)} \times \text{exon length (KB)}}$$

For prokaryotic genes and other non-exon based regions, the calculation is performed in this way:

$$RPKM = \frac{\text{total gene reads}}{\text{mapped reads(millions)} \times \text{gene length (KB)}}$$

Total exon reads This value can be found in the column with header **Total exon reads** in the expression track. This is the number of reads that have been mapped to exons (either within an exon or at the exon junction). When the reference genome is annotated with gene and transcript annotations, the mRNA track defines the exons, and the total exon reads are the reads mapped to all transcripts for that gene. When only genes are used, each gene in the gene track is considered an exon. When an un-annotated sequence list is used, each sequence is considered an exon.

Exon length This is the number in the column with the header **Exon length** in the expression track, divided by 1000. This is calculated as the sum of the lengths of all exons (see definition of exon above). Each exon is included only once in this sum, even if it is present in more annotated transcripts for the gene. Partly overlapping exons will count with their full length, even though they share the same region.

Mapped reads The sum of all mapped reads as listed in the RNA-Seq analysis report. If paired reads were used in the mapping, mapped fragments are counted here instead of reads, unless the **Count paired reads as two** option was selected. For more information on how expression is calculated in this case, see section 33.4.1.

RNA-Seq result handling

Clicking **Next** will allow you to specify the output options as shown in figure 33.40.

The main results of the RNA-Seq analysis are Expression Tracks. A track with a name ending with **(GE)** summarizes expression at the gene level. If the "Genome annotated with genes and transcripts" option was selected, a second track, with a name ending in **(TE)**, is produced, which summarizes expression at the transcript level.

In addition, other results can be generated using the following options:

- **Create reads track.** This track contains the mapping of the reads to the references. This track has a name ending with (Reads).
- **Create report.** See section 33.4.1 for a description of the information contained in the report. This report is the only place results of the spike-in controls will be available.
- **Create list of unmapped reads.** This list is made of reads that did not map to the reference at all, or that were non-specific matches with more placements than specified (see section 33.4.1). If you started with paired reads, then more than one list of unmapped

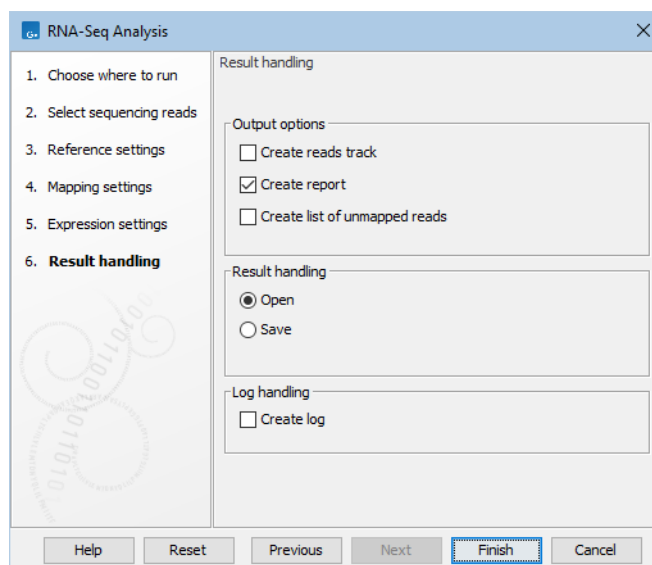


Figure 33.40: Selecting the output of the RNA-Seq analysis.

reads may be produced: paired reads are put in a list with a name that ends in (paired) while single reads, including members of broken pairs, are put in a read list with a name than ends in (single).

Expression tracks

Both tracks can be shown in a **Table**  and a **Graphical**  view.

The expression track table view has the following options (figure 33.41).

- The "Filter to selection" only displays pre-selected rows in the table.
- The "Create track from Selection" will create a Track using selected rows.
- The "Select Genes in Other Views" button finds and selects the currently selected genes and transcripts in all other open expression track table views.
- The "Copy Gene Names to Clipboard" button copies the currently selected gene names to the clipboard.

An example of a track list containing expression results is shown in figure 33.42. In that figure, one of the tracks referred to in the track list has been opened in a linked view. Clicking on a row in that table moves the focus in the track list to the location referred to in that row. General information about linked views is available in section 2.1. Information on creating and working with track lists is provided in section 27.2.

Reads spanning two exons are shown with a dashed line between each end (figure 33.42). The thin solid line represents the connection between two reads in a pair.

Table views of some track types offer buttons at the bottom related to linked viewing. For example, Expression or Statistical Comparison tracks have buttons for putting the focus on selected genes or transcripts in other views.

Rows: 173,446 Table view: Homo sapiens Filter to selection... Filter

Name	Gene name	Transcript...	Exons	ENSEMBL
DDX11L1_1	DDX11L1	1657	3	ENST00000456328 , ENSE00002234944 , ENSG00000223972 , EN
DDX11L1_4	DDX11L1	1653	3	ENSE00002234632 , ENSG00000223972 , ENST00000515242 , EN
DDX11L1_2	DDX11L1	1483	4	ENSG00000223972 , ENST00000518655 , ENSE00002269724 , EN
DDX11L1_3	DDX11L1	632	6	ENSE00001758273 , ENST00000450305 , ENSE00001863096 , EN
WASH7P_4	WASH7P	1416	9	ENSE00003497546 , ENSG00000227232 , ENSE00003638984 , EN
WASH7P_5	WASH7P	1669	10	ENST00000423562 , ENSE00003565315 , ENSG00000227232 , EN
WASH7P_3	WASH7P	1783	12	ENSE00003497546 , ENSE00003638984 , ENSE00001642865 , EN
WASH7P_2	WASH7P	1351	11	ENSE00001890219 , ENSE00003475637 , ENSE00003502542 , EN
WASH7P_1	WASH7P	1583	13	ENSE00002317443 , ENSE00003632482 , ENSE00003638984 , EN
MIR1302-10_2	MIR1302-10	717	3	ENSE00001827679 , ENSE00001947070 , ENSG00000243485 , EN
MIR1302-10_1	MIR1302-10	1583	2	ENSG00000243485 , ENSE00001890064 , ENSE00001841699 , EN
FAM138A_1	FAM138A	1187	3	ENST00000417324 , ENSE00001669267 , ENSE00001656588 , EN
FAM138A_2	FAM138A	590	2	ENSE00001618781 , ENSE00001874421 , ENSG00000237613 , EN
OR4G4P_1	OR4G4P	126	2	ENSE00003074125 , ENST00000594647 , ENSE00003076518 , EN
OR4F5_1	OR4F5	918	1	ENSG00000186092 , ENSE00002319515 , ENST00000335137 , EN
RP11-34P13.7_3	RP11-34P13.7	2748	4	ENSE00001846804 , ENST00000466430 , ENSG00000238009 , EN
RP11-34P13.7_1	RP11-34P13.7	1210	2	ENSE00001847705 , ENSE00001877735 , ENSE00000238009 , EN

Create Track from Selection

Select Transcripts in Other Views

Copy Transcript Names to Clipboard

Figure 33.41: RNA-Seq results shown in a table view.

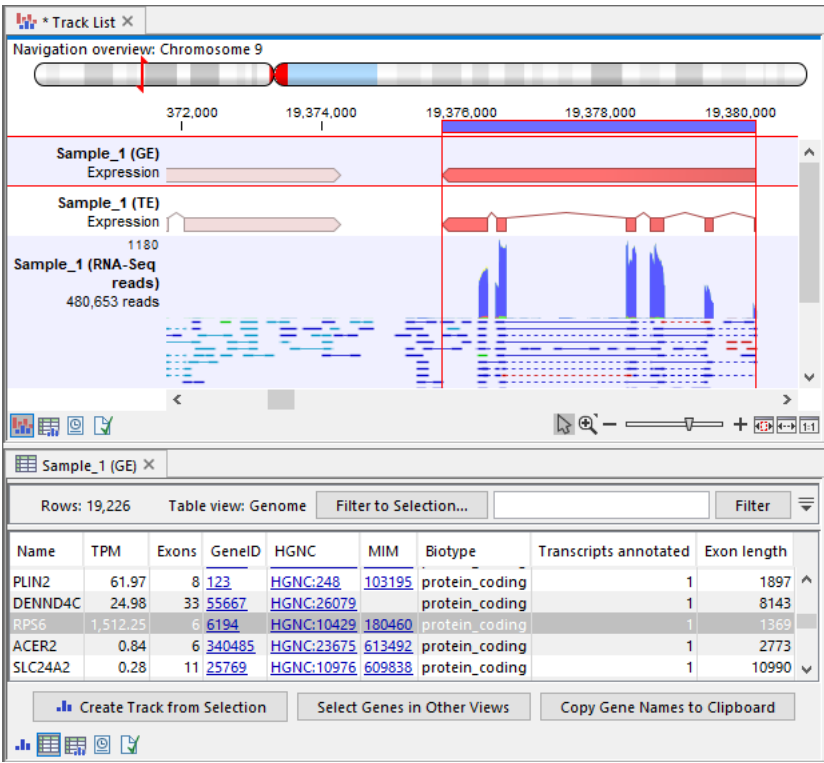


Figure 33.42: A track list containing RNA-Seq results is open in the top of the viewing area, with an expression track open in a linked view beneath.

Expression tracks can also be used to annotate variants using the **Annotate with Overlap Information** tool. Select the variant track as input and annotate with the expression track. For variants inside genes or transcripts, information will be added about expression (counts, expression value etc) from the gene or transcript in the expression track. Read more about the annotation tool in section 27.5.3.

Gene-level expression

The gene-level expression track (GE) holds information about counts and expression values for each gene. It can be opened in a **Table view** (📊) allowing sorting and filtering on all the information in the track (see figure 33.43 for an example subset of an expression track).

Name	Expression value	RPKM	Unique gene reads	Total gene reads
CD44	5,099.00	85.31	6529	6628
MAN2C1	42.00	0.73	48	48
ST3GAL3	3.00	7.08	229	242
MUTYH	4.00	0.63	19	19
SYNE1	58.00	0.71	246	250
VEZT	355.00	7.96	535	538
MOK	12.00	0.72	61	61
C17orf62	16.00	0.30	17	17
AKT2	32.00	0.54	55	55
GUK1	86.00	2.49	91	91
MYB	30.00	1.68	71	71
DMTF1	217.00	4.08	273	274
FGFR1	27.00	0.61	57	60
CTNND1	2,651.00	63.05	3290	4301
PKD1	9.00	0.20	26	28
TEX41	3.00	1.44	114	118
EIF4G1	15.00	0.28	15	18
SYBU	27.00	1.71	89	93

Figure 33.43: A subset of a result of an RNA-Seq analysis on the gene level. Not all columns are shown in this figure


Each row in the table corresponds to a gene (or reference sequence, if the **One reference sequence per transcript** option was used). The corresponding counts and other information is shown for each gene:

- **Name.** The name of the gene, or the name of the reference sequence if "one reference sequence per transcript" is used.
- **Chromosome and region.** The position of the gene on the genome.
- **Expression value.** This is based on the expression measure chosen as described in section 33.4.1.
- **Gene length** The length of the gene as annotated.
- **TPM (Transcripts per million).** This is computed as $\frac{RPKM \cdot 10^6}{\sum RPKM}$, where the sum is over the RPKM values of all genes/transcripts (see <https://bioinformatics.oxfordjournals.org/content/26/4/493.long>).
- **RPKM.** This is the expression value measured in RPKM [Mortazavi et al., 2008]: $RPKM = \frac{\text{total exon reads}}{\text{mapped reads (millions)} \times \text{exon length (KB)}}$. See section 33.4.1 for a detailed definition.
- **Unique gene reads.** This is the number of reads that match uniquely to the gene or its transcripts.

- **Total gene reads.** This is all the reads that are mapped to this gene - both reads that map uniquely to the gene or its transcripts and reads that matched to more positions in the reference (but fewer than the 'Maximum number of hits for a read' parameter) which were assigned to this gene.
- **Transcripts annotated.** The number of transcripts annotated for the gene. Note that this is not based on the sequencing data - only on the annotations already on the reference sequence(s).
- **Uniquely identified transcripts.** The number of transcripts with at least one mapped read that matches only that transcript and no others. Note that if a gene has 4 detected transcripts, and 8 undetected transcripts, all $4+8=12$ transcripts will have the value "Uniquely identified transcripts = 4".
- **Exon length.** The total length of all exons (not all transcripts).
- **Exons.** The total number of exons across all transcripts.
- **Unique exon reads.** The number of reads that match uniquely to the exons (including across exon-exon junctions).
- **Total exon reads.** The total number of reads assigned to an exon or an exon-exon junction of this gene. As for the 'Total gene reads' this includes both uniquely mapped reads and reads with multiple matches that were assigned to an exon of this gene.
- **Ratio of unique to total (exon reads).** The ratio of the unique reads to the total number of reads in the exons. This can be convenient for filtering the results to exclude the ones where you have low confidence because of a relatively high number of non-unique exon reads.
- **Unique exon-exon reads.** Reads that uniquely match across an exon-exon junction of the gene (as specified in figure 33.42). The read is only counted once even though it covers several exons.
- **Total exon-exon reads.** Reads that match across an exon-exon junction of the gene (as specified in figure 33.42). As for the 'Total gene reads' this includes both uniquely mapped reads and reads with multiple matches that were assigned to an exon-exon junction of this gene.
- **Total intron reads.** The total number of reads that map to an intron of the gene.
- **Ratio of intron to total gene reads.** This can be convenient to identify genes with poor or lacking transcript annotations. If one or more exons are missing from the annotations, there will be a relatively high number of reads mapping in the intron.

Transcript-level expression

If the "Genome annotated with genes and transcripts" option is selected in figure 33.35, a transcript-level expression track (TE) is also generated.

The track can be opened in a **Table view**  allowing sorting and filtering on all the information in the track. Each row in the table corresponds to an mRNA annotation in the mRNA track used as reference.

- **Name.** The name of the transcript, or the name of the reference sequence if "one reference sequence per transcript" is used.
- **Chromosome and region.** The position of the gene on the genome.
- **Expression value.** This is based on the expression measure chosen as described in section 33.4.1.
- **TPM (Transcripts per million).** This is computed as $\frac{RPKM \cdot 10^6}{\sum RPKM}$, where the sum is over the RPKM values of all genes/transcripts (see <https://bioinformatics.oxfordjournals.org/content/26/4/493.long>).
- **RPKM.** This is the expression value measured in RPKM [Mortazavi et al., 2008]: $RPKM = \frac{\text{total exon reads}}{\text{mapped reads (millions)} \times \text{exon length (KB)}}$. See section 33.4.1 for a detailed definition.
- **Relative RPKM.** The RPKM for the transcript divided by the maximum of the RPKM values among all transcripts of the same gene. This value describes the relative expression of alternative transcripts for the gene.
- **Gene name.** The name of the corresponding gene.
- **Transcript length.** This is the length of the transcript.
- **Exons.** The total number of exons in the transcript.
- **Transcript ID.** The transcript ID is taken from the transcript_id note in the mRNA track annotations and can be used to differentiate between different transcripts of the same gene.
- **Transcripts annotated.** The number of transcripts based on the mRNA annotations on the reference. Note that this is not based on the sequencing data - only on the annotations already on the reference sequence(s).
- **Uniquely identified transcripts.** The number of transcripts with at least one mapped read that matches only that transcript and no others. Note that if a gene has 4 detected transcripts, and 8 undetected transcripts, all 4+8=12 transcripts will have the value "Uniquely identified transcripts = 4".
- **Unique transcript reads.** This is the number of reads in the mapping for the gene that are uniquely assignable to the transcript.
- **Total transcript reads.** Once the 'Unique transcript read's have been identified and their counts calculated for each transcript, the remaining (non-unique) transcript reads are assigned to one of the transcripts to which they match. The 'Total transcript reads' counts are the total number of reads that are assigned to the transcript once this assignment has been done. As for the assignment of reads among genes, the assignment of reads within a gene but among transcripts, is done by the EM estimation algorithm (section 33.4.1).
- **Ratio of unique to total (transcript reads).** The ratio of the unique reads to the total number of reads in the transcripts. This can be convenient for filtering the results to exclude the ones where you have low confidence because of a relatively high number of non-unique transcript reads.

Additional information to GE or TE tracks

Both GE and TE tables can offer additional information such as hyperlinks to various databases (e.g., ENSEMBL, HGNC (HUGO Gene Nomenclature Committee), RefSeq, GeneID, etc.) In cases where the mRNA track or the gene track provided have biotype information, a biotype column will be added to the table.

RNA-Seq reads track

A track containing the mapped reads can be generated by the tool if the option to do so is enabled. Details about viewing and editing of reads tracks are described in section 27 and section 30.3.2.

If you have chosen the strand specific option when setting up your analysis, it may be helpful to note that the colors of mapped single reads represent the orientation of the read relative to the reference provided. When a gene track is provided along with the reference genome, the reads will be mapped using the strand you specified, but the coloring of the read will be relative to the reference gene. If the reads matches the orientation of the gene it is colored green, and if it is opposite to the orientation of the gene it is colored red. A summary list of the colors to expect with different combinations of gene orientation and strand specific mapping options is:

- Strand specific, forward orientation chosen + gene on plus strand of reference = single reads colored green.
- Strand specific, forward orientation chosen + gene on minus strand of reference = single reads colored red.
- Strand specific, reverse orientation chosen + gene on plus strand of reference = single reads colored red.
- Strand specific, reverse orientation chosen + gene on minus strand of reference = single reads colored green.

See figure 33.44 for an example of forward and reverse reads mapped to a gene on the plus strand.

Note: Reads mapping to intergenic regions will not be mapped in a strand specific way.

Although paired reads are colored blue, they can be viewed as red and green 'single' reads by selecting the **Show strands of paired reads** box, within the Read Mapping Settings bar on the right-hand side of the track.

RNA-Seq report

An example of an RNA-Seq report generated if you choose the **Create report** option is shown in figure 33.45.

The report is a collection of the sections described below, some sections included only based on the input provided when starting the tool. If a section is flagged with a pink highlight, it means that something has almost certainly gone wrong in the sample preparation or analysis. A warning message tailored to the highlighted section is added to the report to help troubleshoot the issue. The report can be exported in PDF or Excel format.

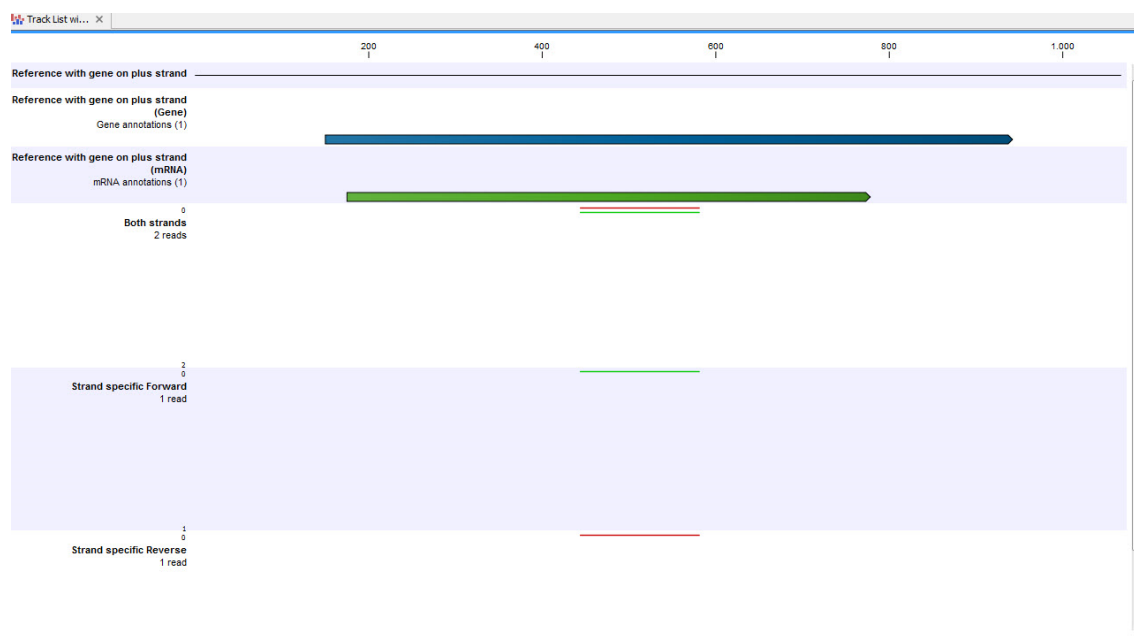


Figure 33.44: A track list showing a gene and transcript on the plus strand, and various mapping results. The first reads track shows a mapping of two reads (one 'forward' and one 'reverse') using strand specific 'both' option. Both reads map successfully; the forward read colored green (because it matches the direction of the gene), and the reverse read colored red. The second reads track shows a mapping of the same reads using strand specific 'forward' option. The reverse read does not map because it is not in the correct direction, therefore only the green forward read is shown. The final reads track shows a mapping of the same reads again but using strand specific 'reverse' option. This time, the green forward read does not map because it is in the wrong direction, and only the red reverse read is shown.

Selected input sequences

Information about the sequence reads provided as input, including the number of reads in each sample, as well as information about the reference sequences used and their lengths.

References

Information about the total number of genes and transcripts found in the reference:

- **Transcripts per gene.** A graph showing the number of transcripts per gene.
- **Exons per transcript.** A graph showing the number of exons per transcript.
- **Length of transcripts.** A graph showing the distribution of transcript lengths.

Spike-in quality control

- **Spike-in plot.** A plot shows the expression of each spike-in as a function of the known concentration of that spike-in (see figure 33.46 to see an optimal spike-in plot).

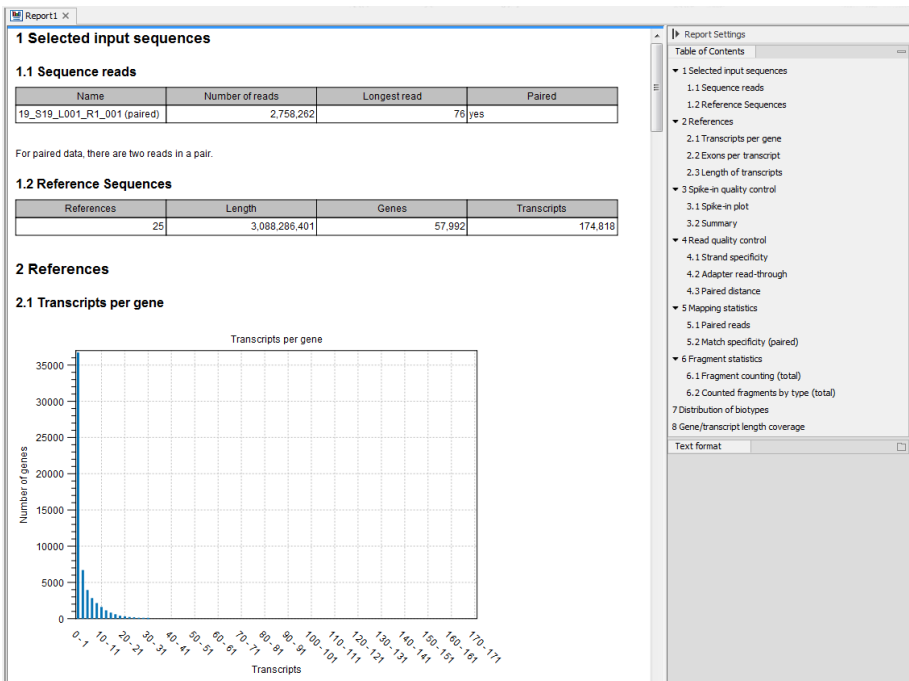


Figure 33.45: Report of an RNA-Seq run.

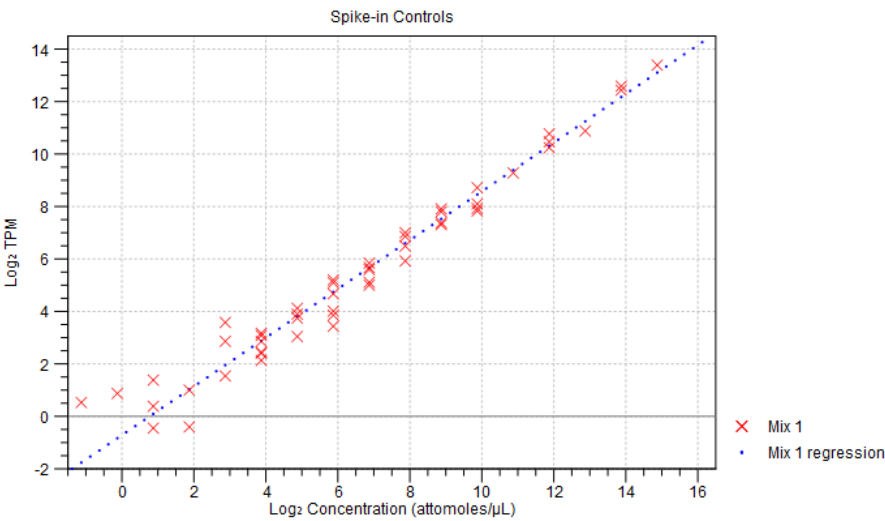
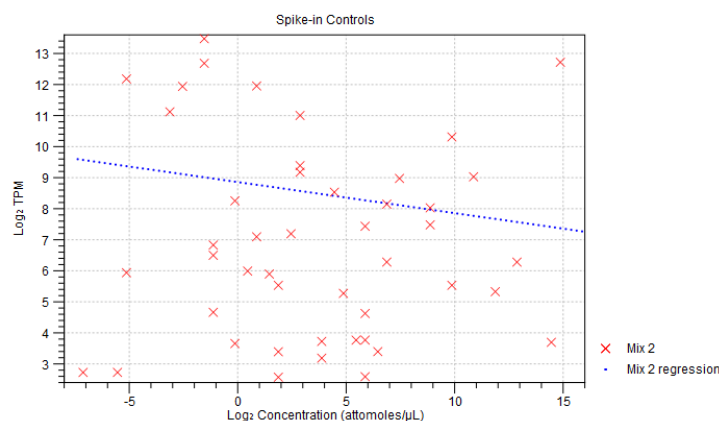


Figure 33.46: Spike-in plot showing how the points fall close to the regression line at high concentration.

- **Summary table.** A table provides more details on the spike-in detection. Figure 33.47 shows a failed spike-in control, with a table where results that require attention are highlighted in pink.
Under the table, a **warning message** explains what the optimal value was, and offers some troubleshooting measures: When samples have poor correlation ($R^2 < 0.8$) between known and measured spike-in concentrations, it indicates problems with the spike-in protocol, or a more serious problem with the sample. To troubleshoot, check that the correct spike-in file has been selected, and control the integrity of the sample RNA. Also, if fewer than 10000



3.2 Summary

Number of spike-ins detected	46/92
R ²	0.05
Reads mapped to spike-ins	7.336
% of reads mapped to spike-ins	3.10
Lower limit of detection (attomoles/μL)	0.03

Figure 33.47: Summary table where less than optimal results are highlighted.

reads mapped to spike-ins, check that the correct spike-in sequences are specified, and consider using more spike-in mix in future experiments.

Read quality control

This section includes:

- A **strand specificity table** that indicates the direction of the RNA fragment that generated the read. Strandedness can only be defined for reads that map to a gene or transcript. Of these reads, the number of "Reads with known strand" is used in determining the percentage of reads ignored due to being on the wrong strand, and the subsequent percentage of reads with the wrong strand. In a strand-specific protocol, almost all reads are generated from a specific orientation, but otherwise a mix of both orientations is expected.
 - A warning message will appear if over 90% of reads were mapped in the same orientation but the tool was run without using a strand specific setting ("Forward"/"Reverse").
 - If over 25% of the reads were filtered away due to the strand specific setting, try to re-run the tool with strand specific setting "Both". However, if a strand-specific protocol was used, library preparation may have failed.
- A **percentage of mapped paired-end reads containing read-through adapters**. If present in above 10% of the reads, adapters may lead to false positive variant calls or incorrect transcript quantification (because reads must align within transcript annotations to be counted towards expression). Read-through adapters can be removed using the Trim Reads tool. In future experiments, consider selecting fragments that are longer than the read size. Note that single base extensions such as TA overhangs will also be classed as read-through adapters, and in these cases the additional base should be trimmed. Note also that trimming of read starts (5' trim) can lead to spurious detection of read-through

adapters, because the trimming increases the number of read pairs where the end of one read aligns over the (trimmed) start of the other.

- A **paired distance graph** (only included if paired reads are used) shows the distribution of paired-end distances, which is equivalent to the distribution of sequenced RNA fragment sizes. There should be a single broad peak at the target fragment size. An asymmetric peak may indicate problems in size selection.

Mapping statistics

Shows statistics on:

- **Paired reads** or **Single reads**. The table included depends on the reads used. The table shows the number of reads mapped or unmapped, and in the case of paired reads, how many reads mapped in pairs and in broken pairs.

If over 50% of the reads did not map, and the correct reference genome was selected, this indicates a serious problem with the sample. To troubleshoot, the report offers the following options:

- Check that the correct reference genome and any relevant gene/mRNA tracks have been provided.
- The mapping parameters may be too strict. Try resetting them to the default values.
- Try mapping the unmapped reads against possible contaminants. If the sample is contaminated, enrich for the target species before library preparation in future experiments.
- Library preparation may have failed. Check the quality of the sample RNA.

In case paired reads are used and over 40% of them mapped as broken pairs, if the counting scheme is not set to "Count paired reads as one and broken pairs as two" in the Expression settings dialog, the report hints that there could be problems with the tool settings, a low quality reference sequence, or incomplete gene/mRNA annotations. It could also indicate a more serious problem with the sample. To troubleshoot, it is suggested to:

- Check that the correct reference genome and any relevant gene/mRNA tracks have been provided.
- Try re-running the tool with the "Auto-detect paired distances" option selected.
- Check that the paired-end distances on the reads are set correctly. These are shown in the "Element Information" view on the reads. If these are correct, try re-running the tool without the "Auto-detect paired distances" option.
- Try mapping the reads against possible contaminants. If the sample is contaminated, enrich for the target species before library preparation in future experiments.

- **Match specificity**. Shows a graph of the number of match positions for the reads. Most reads will be mapped 0 or 1 time, but there will also be reads matching more than once in the reference. The maximum number of match positions is limited in the **Maximum number of hits for a read** setting in figure 33.36. Note that the number of reads that are mapped 0 times includes both the number of reads that cannot be mapped at all and the number of reads that matches to more than the **Maximum number of hits for a read** parameter.

Fragment statistics

- **Fragment counting.** Lists the total number of fragments used for calculating expression, divided into uniquely and non-specifically mapped reads, as well as uncounted fragments (see the point below on match specificity for details).
- **UMI fragment counting.** Lists the total number of distinct UMI fragments used for calculating expression. This table is only included if the Library type setting is 3' sequencing and if the input reads are single end reads annotated with UMIs by tools of the Biomedical Genomics Analysis plugin.
- **Counted fragments by type.** Divides the fragments that are counted into different types, e.g., uniquely mapped, non-specifically mapped, mapped. A last column gives the percentage of fragments mapped for a particular type.
 - **Total gene reads.** All reads that map to the gene.
 - **- Intron.** From the total gene reads, reads that fall partly or entirely within an intron.
 - **- Exon.** From the total gene reads, reads that fall entirely within an exon or in an exon-exon junction.
 - **-- Exon.** From the total gene - exon reads, reads that map completely within an exon
 - **- - Exon-exon.** From the total gene - exon reads, reads that map across an exon junction as specified in figure 33.42 .
 - **Intergenic.** All reads that map partly or entirely between genes.
 - **Total.** Total amount of reads for a particular type.
- **Counted UMI fragments by type.** Divides the distinct UMI fragments that are counted into different types, e.g., uniquely mapped, non-specifically mapped, mapped. The table contains the same rows as the 'Counted fragments by type' table (see above). It is only included in the report if the Library type setting is 3' sequencing and if the input reads are single end reads annotated with UMIs by tools of the Biomedical Genomics Analysis plugin.

Distribution of biotypes

Table generated from biotype annotations present on the input gene or mRNA tracks. If using both gene and mRNA tracks, the biotypes in the report are taken from the mRNA track.

- For genes, biotypes can be any of the following columns: "gene_biotype", "biotype", "gbkey", "type". The first one in this list is chosen.
- For transcripts, biotypes can be any of the following columns: "transcript_biotype", "biotype", "gbkey", "type". The first one in this list is chosen.

The biotypes are "as a percentage of all transcripts" or "as a percentage of all genes". For a poly-A enrichment experiment, it is expected that the majority of reads correspond to protein-coding regions. For an rRNA depletion protocol, a variety of non-coding RNA regions may also be observed. The percentage of reads mapping to rRNA should usually be <15%.

If over 15% of the reads mapped to rRNA, it could be that the poly-A enrichment/rRNA depletion protocol failed. The sample can still be used for differential expression and variant calling, but

expression values such as TPM and RPKM may not be comparable to those of other samples. To troubleshoot the issues in future experiments, check for rRNA depletion prior to library preparation. Also, if an rRNA depletion kit was used, check that the kit matches the species being studied.

Gene/transcript length coverage

Plot showing the normalized coverage across a gene/transcript body for four different groupings of gene/transcript length (figure 33.48).

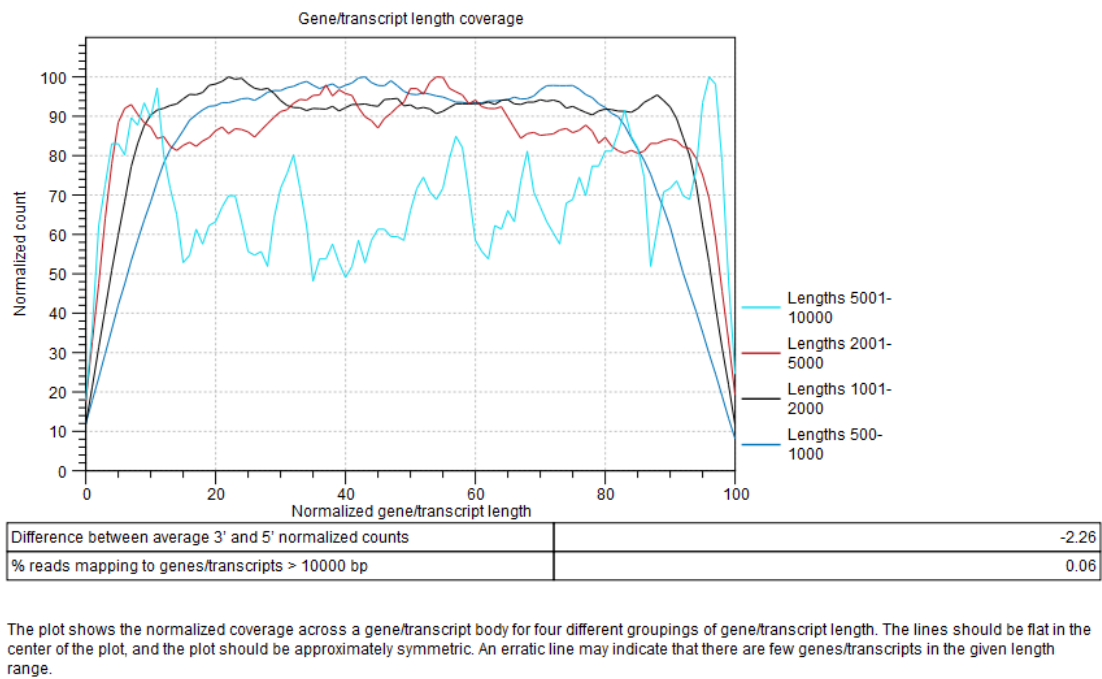


Figure 33.48: Gene/transcript length coverage plot.

To generate this plot, every transcript is rescaled to have a length of 100. For every read that is assigned to a transcript, we get its start and end coordinates in this "transcript-length-normalized" coordinate system [0,100]. We then increment counters from the read start position to the read end position. After all the reads have been counted, the average 5' count is the average value of the counters at position 0,1,2...49. The average 3' count is the value at positions 51,52,53...100. The difference between average 3' and 5' normalized counts is the difference between these values as a percentage of the maximum number of counts seen at any position.

The lines should be flat in the center of the plot, and the plot should be approximately symmetric. An erratic line may indicate that there are few genes/transcripts in the given length range. Lines showing normalized count higher on the 3'end indicates the presence of polyA tails in the reads, consequence of degraded RNAs. Future experiments may benefit from using an rRNA depletion protocol.

In the table below the plot, a difference between average 3' and 5' normalized counts higher than 25 warns that variants may not be called in low coverage regions, and that TPM or RPKM values may be unreliable. Most transcripts are <10000 bp long, so a warning is raised if many reads map to features longer than this. One possible cause is that no mRNA track has been

provided for an organism with extensive splicing.

33.4.2 RNA-Seq Analysis for Long Reads

The RNA-Seq Analysis for Long Reads tool supports analysis of RNA-Seq data by mapping sequencing reads to an annotated reference genome with minimap2 [Li, 2018] and distributing and counting the reads across genes and transcripts. Subsequently, the results can be used for expression analysis.

RNA-Seq analysis with long reads is done in several steps: First, all annotated transcripts or genes are extracted. If there are several annotated splice variants, they are all extracted. Next, the reads are mapped against all the transcripts, and to the whole genome using minimap2. For more information about the read mapper, see section 30.2.

From this mapping, the reads are categorized and assigned to the transcripts using the EM estimation algorithm, and expression values for each gene are obtained by summing the transcript counts belonging to the gene.

For detailed information on RNA-Seq analysis including the EM algorithm, see section 33.4.1.

Reads and reference settings

To run the RNA-Seq Analysis for Long Reads tool, go to:

Tools | RNA-Seq and Small RNA Analysis (🇺🇸) | **RNA-Seq Tools** (🇬🇧) | **RNA-Seq Analysis for Long Reads** (🇯🇵)

Select one or more sequence lists containing long reads.

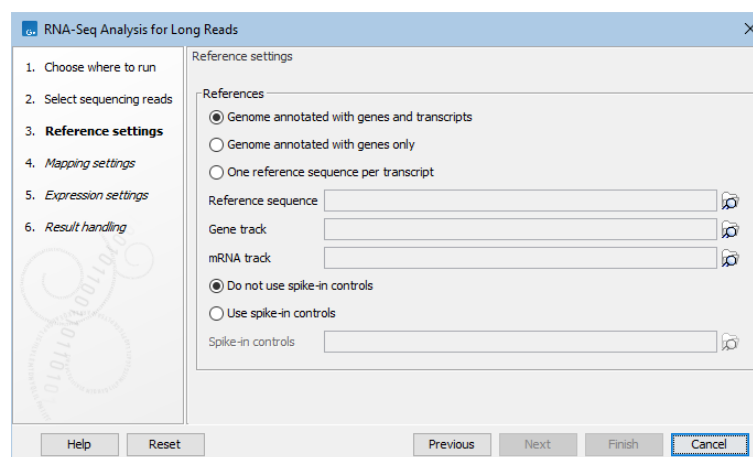


Figure 33.49: RNA-Seq Analysis for Long Reads reference settings.

In the *Reference settings* dialog (figure 33.49), at the top there are three options concerning how the reference sequences are annotated.

- **Genome annotated with genes and transcripts.** This option should be used when both gene and mRNA annotations are available. When this option is enabled, the EM algorithm will distribute the reads over the transcripts. Gene counts are then obtained by summing over the (EM-distributed) transcript counts. The mRNA annotations are used to define how

the transcripts are spliced. This option should be used for Eukaryotes since it is the only option where splicing is taken into account. Note that genes and transcripts are linked by name only (not by position, ID etc).

When this option is selected, both a *Gene* and an *mRNA* track should be provided in the boxes below. Annotated reference genomes can be obtained in various ways:

- Directly downloaded as tracks using the Reference Data Manager.
- Imported as tracks from fasta and gff/gtf files.
- Imported from Genbank or EMBL files and converted to tracks.
- Downloaded from Genbank.

When using this option, Expression values, RPKM and TPM are calculated based on the lengths of the transcripts provided by the mRNA track. If a gene's transcript annotation is absent from the mRNA track, all values will be set to zero unless the option "Calculate expression for genes without transcript" is checked in a later dialog.

- **Genome annotated with genes only.** This option should be used for Prokaryotes where transcripts are not spliced. When this option is selected, a *Gene* track should be provided in the box below. The data can be obtained in the same ways as described above.

When using this option, Expression values, RPKM and TPM are calculated based on the lengths of the genes provided by the Genes track.

- **One reference sequence per transcript.** This option is suitable for situations where the reference is a list of sequences. Each sequence in the list will be treated as a "transcript" and expression values are calculated for each sequence. This option is most often used if the reference is a product of a *de novo* assembly of RNA-Seq data. It is also a suitable option for references where genes are particularly close to each other or clustered in operon structures. When this option is selected, only the reference sequence should be provided, either as a sequence track or a sequence list. Expression values, RPKM and TPM are calculated based on the lengths of sequences from the sequence track or sequence list.

At the bottom of the dialog you can choose between these two options:

- **Do not use spike-in controls.**
- **Use spike-in controls.** When selected, you provide a spike-in control file in the field below.

To learn how to import spike-in control files, see section 7.5.

Mapping settings

In the next dialog, set the mapping parameters (figure 33.50).

The generic mapping parameters are identical to those applied for **Map Long Reads to Reference**, see section 30.2.

In addition, the following RNA-Seq specific setting is available:

- **Maximum number of hits for a read.** Reads mapping equally well to more places than this number will be ignored. If a read matches to multiple distinct places, but less than or equal to the specified maximum number, it will be assigned to one of these places by the EM algorithm.

Figure 33.50: Define mapping settings for RNA-Seq Analysis for Long Reads.

Expression settings

Expression settings are defined in the dialog shown in figure 33.51.

Figure 33.51: Set strand setting and define how expression values should be calculated.

These parameters determine the way expression values are counted.

- **Strand setting**

- **Both.** Reads are mapped both in the same and reversed orientation as the transcript from which they originate. This is the default.
- **Forward.** Reads are mapped in the same orientation as the transcript from which they originate.
- **Reverse.** Reads are mapped in the reverse orientation as the transcript from which they originate.

If a strand specific protocol for read generation was used, choose the corresponding setting. This allows assignment of the reads to the right gene in cases where overlapping genes are located on different strands.

- **Expression value.** This parameter describes how expression per gene or transcript can be defined in different ways on both levels:

- **Total counts.** When the reference is annotated with genes only, this value is the total number of reads mapped to the gene. For un-annotated references, this value is the total number of reads mapped to the reference sequence. For references annotated with transcripts and genes, the value reported for each gene is the number of reads that map to the exons of that gene. The value reported per transcript is the total number of reads mapped to the transcript.
- **Unique counts.** This is similar to the above, except only reads that are uniquely mapped are counted.
- **TPM.** (Transcripts per million). This is computed as $\frac{RPKM \cdot 10^6}{\sum RPKM}$, where the sum is over the RPKM values of all genes/transcripts.
- **RPKM.** This is a normalized form of the "Total counts" option (see more under *Definition of RPKM* below).

All values are present in the output. The choice of expression value only affects how Expression Tracks are visualized in the track view but the results will not be affected by this choice as the most appropriate expression value is automatically selected for the analysis being performed. For detection of differential expression this is the "Total counts" value, and for the other tools this is a normalized and transformed version of the "Total counts" as described below.

- **Calculate expression for genes without transcripts.** For genes without annotated transcripts, the RPKM cannot be calculated since the total length of all exons is needed. By selecting this option, the length of the gene will be used in place of an "exon length". If the option is not checked, there will be no RPKM value reported for those genes.

Definition of RPKM

RPKM, Reads Per Kilobase of exon model per Million mapped reads, is defined in this way [Mortazavi et al., 2008]:

$$RPKM = \frac{\text{total exon reads}}{\text{mapped reads(millions)} \times \text{exon length (KB)}}.$$

For prokaryotic genes and other non-exon based regions, the calculation is performed in this way:

$$RPKM = \frac{\text{total gene reads}}{\text{mapped reads(millions)} \times \text{gene length (KB)}}.$$

Total exon reads. This value can be found in the column with header **Total exon reads** in the expression track. This is the number of reads that have been mapped to exons (either within an exon or at the exon junction). When the reference genome is annotated with gene and transcript annotations, the mRNA track defines the exons, and the total exon reads are the reads mapped to all transcripts for that gene. When only genes are used, each gene in the gene track is considered an exon. When an un-annotated sequence list is used, each sequence is considered an exon.

Exon length. This is the number in the column with the header **Exon length** in the expression track, divided by 1000. This is calculated as the sum of the lengths of all exons (see definition of exon above). Each exon is included only once in this sum, even if it is present in more annotated transcripts for the gene. Partly overlapping exons will count with their full length, even though they share the same region.

Mapped reads. The sum of all mapped reads as listed in the RNA-Seq analysis report. For more information on how expression is calculated in this case, see above.

RNA-Seq Analysis for Long Reads output

Output options are specified in the dialog shown in figure 33.52.

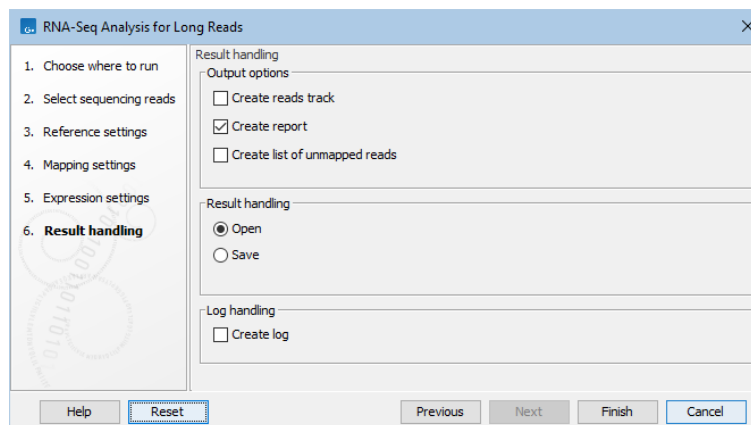


Figure 33.52: The output options for RNA-Seq using long reads.

The main results of the RNA-Seq analysis are Expression Tracks. A track with a name ending with **(GE)** summarizes expression at the gene level. If the "Genome annotated with genes and transcripts" option was selected, a second track, with a name ending in **(TE)**, is produced, which summarizes expression at the transcript level.

In addition, the following outputs are available:

- **Create reads track.** This track contains the mapping of the reads to the references.
- **Create report.** A summary report.
- **Create list of unmapped reads.** This list is made of reads that did not map to the reference, or that were non-specific matches with more placements than specified.

Expression tracks

Expression tracks can be shown in a **Table** (📊) and a **Track** (📈) view.

The expression track table view has the following options (figure 33.53).

- The "Filter to selection" only displays pre-selected rows in the table.
- The "Create track from Selection" will create a new Track with the selected rows.
- The "Select Genes/Transcripts in Other Views" button finds and selects the currently selected genes or transcripts in all other open expression track table views.
- The "Copy Gene/Transcript Names to Clipboard" button copies the currently selected gene or transcript names to the clipboard.

For detailed information on expression tracks, see section 33.4.1.

Rows: 33,538 Table view: Genome					
Filter to Selection...					
Filter					
Name	TPM	Exons	Biotype	Transcripts annota...	Uniquely identified ...
gene07104-v2.0.a2-hybrid	7,737.70		11 Gene	1	1
gene08023-v2.0.a2-hybrid	7,175.25		14 Gene	3	3
gene26994-v2.0.a2-hybrid	7,114.26		5 Gene	1	1
gene19551-v2.0.a2-hybrid	6,864.27		7 Gene	3	3
gene22974-v1.0-hybrid	6,122.62		12 Gene	6	6
gene22465-v1.0-hybrid	6,032.70		3 Gene	1	1
gene01495-v1.0-hybrid	5,845.13		5 Gene	2	2
gene21573-v1.0-hybrid	5,819.01		8 Gene	3	3
gene18240-v2.0.a2-hybrid	5,668.02		5 Gene	2	2
gene30512-v1.0-hybrid	5,413.05		11 Gene	1	1
gene13212-v1.0-hybrid	5,192.39		1 Gene	1	1
gene01608-v1.0-hybrid	4,871.66		7 Gene	2	2
gene20927-v1.0-hybrid	4,863.98		6 Gene	2	2
gene03809-v1.0-hybrid	4,535.41		6 Gene	3	3
gene09659-v2.0.a2-hybrid	4,431.43		1 Gene	1	1
gene26083-v2.0.a2-hybrid	4,253.76		11 Gene	3	1
gene25105-v2.0.a2-hybrid	4,126.40		11 Gene	2	2
gene28639-v1.0-hybrid	4,080.65		5 Gene	2	2
gene26908-v1.0-hybrid	3,939.64		2 Gene	1	1
gene23293-v1.0-hybrid	3,799.04		6 Gene	2	2
gene06563-v1.0-hybrid	3,686.90		6 Gene	2	2
gene11464-v1.0-hybrid	3,648.93		7 Gene	1	1
gene20983-v1.0-hybrid	3,528.43		5 Gene	1	1

Figure 33.53: A gene-level expression track shown in Table view.

RNA-Seq reads track

A track containing the mapped reads can be generated by the tool if the option to do so is enabled. The graphical view of the mapped reads can be shown together with the expression tracks in a **Track list**. For additional information on this output, see section 33.4.1.

RNA-Seq Analysis for Long Reads report

An example of first part of the RNA-Seq report is shown in figure 33.54.

For a detailed description of the report sections, see section 33.4.1.

33.4.3 Detect and Refine Fusion Genes

Detect and Refine Fusion Genes uses information from unaligned read ends in a read mapping (Figure 33.54) to detect potential fusions between gene pairs. Candidate fusions are selected among these, and are refined by mapping the input reads (Figure 33.55) to both the original wildtype reference genome and an artificial fusion genome (figure 33.55). The tool produces comprehensive outputs (see section 33.4.3), enabling detailed analysis of the evidence for fusions.

Known limitations

The tool is not suitable for detection of fusions involving:

- CircRNAs, as it disregards evidence of back-splicing.
- A mix of sense and antisense exons.
- More than two genes.

1 Selected input sequences

1.1 Sequence reads

Name	Number of reads	Longest read	Paired
PacBio SRR3497430	442,601	11,501	no

For paired data, there are two reads in a pair.

1.2 Reference sequences

References	Length	Genes	Transcripts
8	211,673,467	33,538	50,732

2 References

2.1 Transcripts per gene

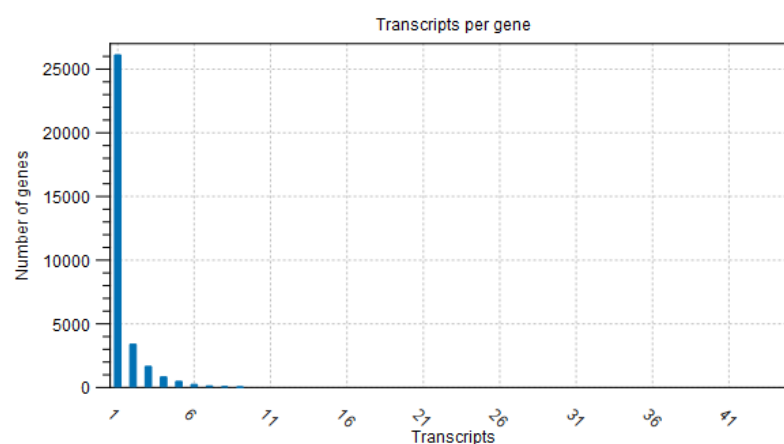


Figure 33.54: Report of an RNA-Seq Analysis for Long Reads.

- A region before the first annotated exon or after the last annotated exon.

Running Detect and Refine Fusion Genes

To run the tool, go to:

Tools | RNA-Seq and Small RNA Analysis (📁) | **RNA-Seq Tools** (🔧) | **Detect and Refine Fusion Genes** (🔍)

The tool takes as input one or more sequence lists (📄) containing reads.

Selecting references

The following options can be configured in the **References** dialog (figure 33.56):

- **RNA-Seq reads track.** A reads track (📄) produced by **RNA-Seq Analysis**, see section 33.4.1, where the same input reads (📄) were used.
- **Reference sequence.** A sequence track (📄) with the wildtype (WT) reference genome.
- **Gene track.** An annotation track (📄) with the WT gene annotations.

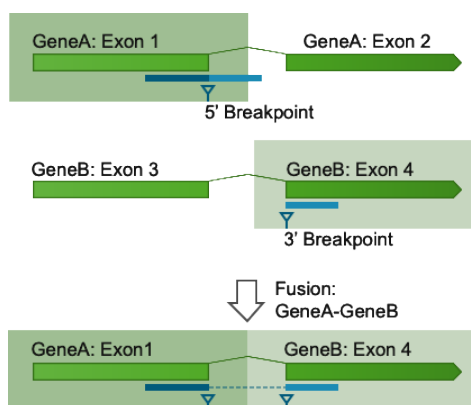


Figure 33.55: A read is mapped to GeneA in the input read mapping, with an unaligned end. The unaligned end is then remapped to GeneB. An artificial fusion chromosome is created for a candidate fusion between GeneA and GeneB. This fusion has its 5' breakpoint at the exon boundary of GeneA and its 3' breakpoint at the exon boundary of GeneB.

- **mRNA track.** An annotation track (📄) with the WT transcript annotations. The same tracks as used for **RNA-Seq Analysis** should be provided.
- **CDS track.** An optional annotation track (📄) with WT coding sequence annotations.
- **Primer track.** An optional annotation track (📄) with the WT primer regions.

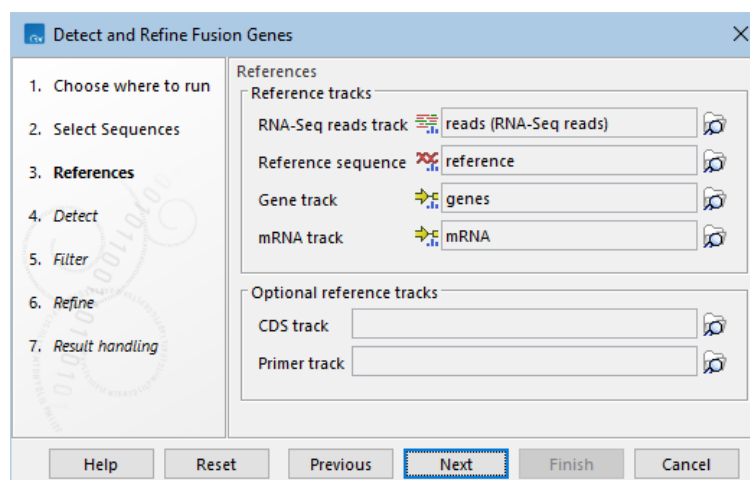


Figure 33.56: Reference tracks for Detect and Refine Fusion Genes.

Detecting fusions

Potential fusions are detected by:

- Identifying unaligned ends in gene regions from the input 'RNA-Seq reads track'.
- Mapping the unaligned ends to the WT reference genome using 'RNA-Seq Analysis' with default options.

- Determining potential fusions using the unaligned ends that map within gene regions.
 - A potential fusion consists of a 5' and 3' gene.
 - A fusion between a 5' and 3' gene is defined by a pair of breakpoints: the 5' breakpoint is located at the beginning of the unaligned end in the input 'RNA-Seq reads track', while the 3' breakpoint is at the beginning of the unaligned end mapped to the WT reference genome (figure 33.55).
 - For a given 5' and 3' gene pair, multiple fusions with distinct breakpoint pairs may be identified.

The following options can be configured in the **Detect** dialog (figure 33.57):

- **Minimum length of unaligned ends.** Only unaligned ends longer than this are used to detect fusions.
- **Maximum distance to known exon boundary.** Potential fusions are built using unaligned ends that:
 - Are located in the input 'RNA-Seq reads track' within this distance of a known exon boundary.
 - Map to the WT reference genome within this distance of another known exon boundary.

The fusion breakpoints are placed at the respective exon boundaries.

The signal for fusions can be increased by using larger maximum distances, as unaligned ends farther from known exon boundaries are then also considered. However, this also decreases the resolution, as transcripts with exon boundaries within this distance of each other would not be differentiated. This can reduce the number of mapped reads in the refinement step.

- **Detect exon skipings.** When checked, fusions involving the same gene, corresponding to transcripts that have missing exons compared to the WT transcript annotation, are detected.
- **Detect novel exon boundaries.** When checked, fusions with one breakpoint located farther than 'Maximum distance to known exon boundary' from an exon boundary are also detected.
 - **Detect novel exon boundaries in both genes.** When checked, fusions with both breakpoints farther than 'Maximum distance to known exon boundary' from an exon boundary are also detected.

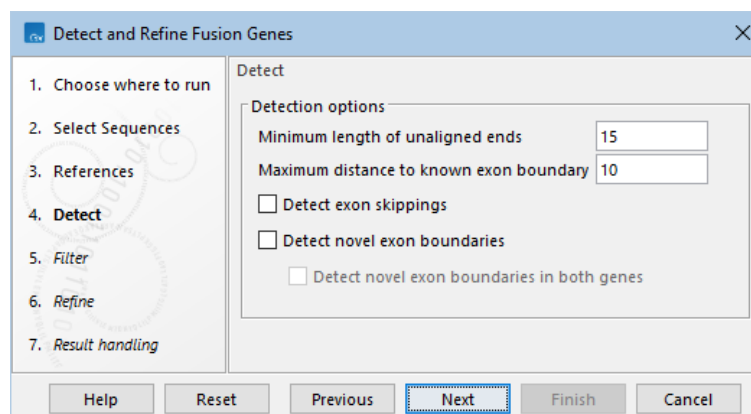


Figure 33.57: Default options for detecting fusion genes.

Filtering fusions

Candidate fusions are identified among the detected potential fusions by:

- Optionally filtering using lists of genes and/or fusions.
- Applying a series of filters:
 - The best scoring fusions, based on p-value and Z-score, are selected.
 - The number of breakpoint pairs for one gene pair is capped at 20.
 - Multiple breakpoint pairs for the same gene pair are treated as one candidate fusion.

The following options can be configured in the **Filter** dialog (figure 33.58):

- **Gene filter action.** How detected fusions should be filtered based on the 5' and 3' genes. The following filter actions are available:
 - **None.** Detected fusions are not filtered.
 - **Exclude.** Detected fusions with a 5' or 3' gene in the provided track or list are removed. This can be useful for removing unwanted fusions.
 - **Include.** Only detected fusions with a 5' or 3' gene in the provided track or list are kept. This can be useful for restricting fusion detection to only genes of interest.


The genes used for filtering can be provided using at least one of the following options:

- **Genes for filtering (tracks).** A gene track (📊) that is a subset of the reference gene track. The gene IDs are used for filtering. Can be left empty or multiple tracks can be provided.
Several tracks suitable for excluding genes are available via the Reference Data Manager, see section 33.4.3 for details.
- **Genes for filtering (names).** A list of case-sensitive gene IDs or names separated by comma, semicolon, or any white-space character. Can be left empty.

Additionally, the following can be configured:

- **Require both genes.** When checked, both 5' and 3' genes must be in the provided track or list in order for the fusion to be filtered.
- **Fusion filter action.** How detected fusions should be filtered. The following filter actions are available:
 - **None.** Detected fusions are not filtered.
 - **Exclude.** Detected fusions in the provided table or list are removed. This can be useful for removing fusions that are frequently detected as false positives.
 - **Include.** Only detected fusions in the provided table or list are kept. This can be useful for restricting fusion detection to only fusions of interest.

The fusions used for filtering can be provided using at least one of the following options:

- **Fusions for filtering (tables).** A table  with case-sensitive gene IDs or names. Can be left empty or multiple tables can be provided. The following formats are supported:
 - * The first two columns contain one gene each.
 - * The first column has a header with the word 'fusion' (case-insensitive) and contains gene pairs in the format 'gene1-gene2'.
 - * The first column has a header with the word 'genes' (case-insensitive) and contains a group of genes in the format 'gene1 gene2 gene3'. Each group results in all possible gene combinations, such as 'gene1-gene2', 'gene1-gene3', and 'gene2-gene3'.

When **Detect exon skipplings** is checked, exon skipplings are only filtered for genes that appear twice within the same group.

All other columns are ignored.

See section 7.1 for details on how to import tables.

Several tables suitable for excluding fusions are available via the Reference Data Manager, see section 33.4.3 for details.

- **Fusions for filtering (names).** A list of case-sensitive gene IDs or names in the format 'gene1-gene2' separated by comma, semicolon, or any white-space character. Can be left empty.

If the IDs or names of genes in the fusions used for filtering contain a '-', the 'Fusions for filtering (tables)' using the second format ('fusion' header) and 'Fusions for filtering (names)' cannot be used. We recommend choosing one of the alternative options instead.

The fusions used for filtering are non-directional: a gene pair 'gene1-gene2' applies to fusions where gene1 is the 5' gene and gene2 is the 3' gene, as well as fusions where gene2 is the 5' gene and gene1 is the 3' gene.

- **Assumed error rate.** The probability of the binomial distribution used to calculate p-values. For example, an assumed error rate of 0.001 indicates that, on average, 1 in 1000 reads covering a breakpoint supports the fusion by chance.

The default value is deliberately conservative, which may result in low-frequency fusions not being selected as candidate fusions. By decreasing the value, more candidate fusions can be identified.

- **Maximum number of fusions.** The maximum number of candidate fusions.

- **Minimum unaligned end count.** The minimum number of unaligned ends that must support a candidate fusion.
- **Promiscuity threshold.** The maximum number of candidate fusions for a given gene.
- **Include all fusions in the WT track output.** If checked, detected potential fusions that were not selected as candidate fusions are included in the 'Fusion genes, WT' track output. See *Filtered fusions* (section 33.4.3) for details.

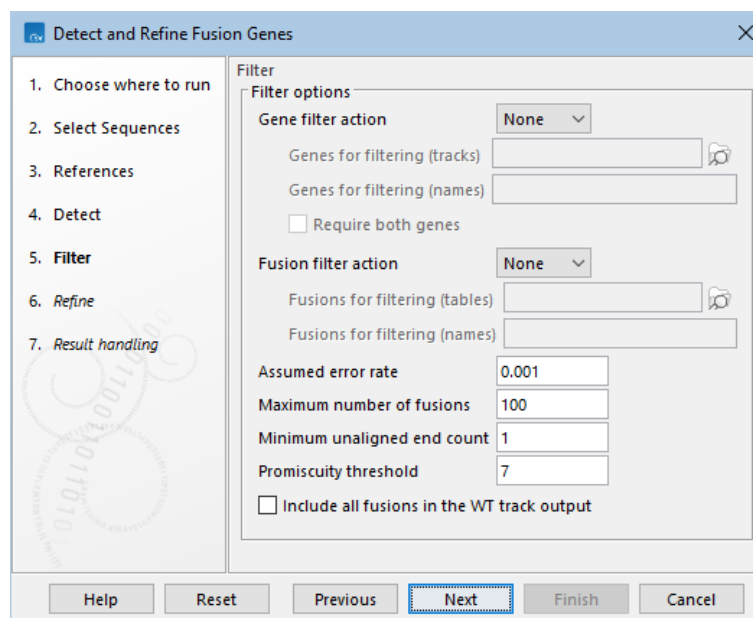


Figure 33.58: Default options for filtering fusion genes.

Refining fusions

Once candidate fusions have been identified, their support is further refined by:

- Creating an artificial fusion genome with one chromosome for each candidate fusion. Corresponding transcripts for all detected breakpoints are created, along with tracks for genes, and, if available, CDS and primers.
- Mapping the input reads simultaneously to the WT and artificial fusion genomes, allowing them to be assigned to the genome (WT or artificial fusion) where they map most accurately.
- Counting the number of reads supporting the fusion vs the WT genes, and recalculating p-values and Z-scores.
- Evaluating the support for candidate fusions for each identified breakpoint pair, and assigning a PASS status to breakpoint pairs if they are found to be adequately supported.

The following options can be configured in the **Refine** dialog (figure 33.59):

- **Mapping options.** Options for mapping the input reads to the WT reference genome and artificial fusion genome. See section 33.4.1 for details.

- **Minimum number of supporting reads.** The minimum number of reads required for a PASS status.
- **Maximum p-value.** The maximum allowed p-value for a PASS status.
- **Minimum Z-score.** The minimum required Z-score for a PASS status.
- **Breakpoint distance.** The minimum number of bases that a read must map within the two exons flanking the breakpoint pair for it to be counted as supporting the fusion. This value should be lower than the length of the shortest flanking exon. Note that setting this value too low may lead to reads mapping by chance being counted as supporting a fusion.
- **Report only significant breakpoints.** When checked, only breakpoint pairs with a PASS status are added to the report.

The screenshot shows a software window titled "Detect and Refine Fusion Genes". On the left is a vertical list of steps: 1. Choose where to run, 2. Select Sequences, 3. References, 4. Detect, 5. Filter, 6. Refine (selected), and 7. Result handling. The main area is divided into two sections: "Mapping options" and "Refinement options".

Mapping options:

- Mismatch cost: 2
- Insertion cost: 3
- Deletion cost: 3
- Length fraction: 0.8
- Similarity fraction: 0.8
- ☐ Global alignment

Refinement options:

- Minimum number of supporting reads: 2
- Maximum p-value: 0.005
- Minimum Z-score: 2.5
- Breakpoint distance: 10
- ☒ Report only significant breakpoints

At the bottom are buttons: Help, Reset, Previous, Next (highlighted with a blue border), Finish, and Cancel.

Figure 33.59: Default options for refining fusion genes.

The default values for 'Maximum p-value' and 'Minimum Z-score' are deliberately conservative, which may result in low-frequency fusions not being assigned a PASS status. By increasing 'Maximum p-value' and/or decreasing 'Minimum Z-score', more fusions can be identified.

Exclude lists

Several gene and fusion exclude lists are available via the **QIAGEN Sets** tab, see section 11.2, of the Reference Data Manager.

The gene exclude lists are annotation tracks (📁) that contain genes known to cause false positive fusions or that are often not of interest for fusion detection:

- **HLA genes.** Human leukocyte antigen genes, also known as major histocompatibility complex genes.
- **Ig and TCR genes.** Immunoglobulin and T cell receptor genes.

- **LncRNA genes.** Long non-coding RNA genes.
- **MT genes.** Genes encoded by the mitochondrial DNA.
- **Other ncRNA genes.** Various short, non-coding RNA genes, including miRNAs, rRNAs, tRNAs, snRNAs, and snoRNAs.
- **Pseudogenes.** Genes that are non-functional copies of functional genes.
- **Readthrough genes.** Genes that produce transcripts containing exons from several distinct neighboring genes. Also known as conjoined genes.
- **TEC genes.** Genes that are yet to be experimentally confirmed.





The fusion exclude lists are tables (📄) that contain gene pairs known to cause false positive fusions or that are often not of interest:

- **Antisense fusions.** Pairs of antisense genes and their sense genes.
- **Paralog fusions.** Groups of genes that are all paralogous to each other.
- **Pseudogene fusions.** Pairs of pseudogenes and their parent genes.
- **Readthrough fusions.** Groups of genes that are all part of the same readthrough gene.
- **Healthy tissue fusions.** Pairs of genes that are recurrent fusion partners in healthy tissue, provided by FusionCatcher (<https://github.com/ndaniel/fusioncatcher>).


Output from Detect and Refine Fusion Genes

The tool produces the following outputs:

- **Fusion genes, WT.** An annotation track (📄) with the fusion breakpoints relative to the wildtype (WT) reference genome. See *Fusion tracks* below for more details.
- **Reads, WT.** A reads track (📄) with the input reads mapped to the WT reference genome after refinement.
- **Unaligned ends, WT.** A reads track (📄) with the unaligned ends mapped to the WT reference genome. This can be useful in choosing optimal values for 'Minimum unaligned end count', 'Minimum length of unaligned ends', and 'Maximum distance to known exon boundary'.
- **Fusion genes, fusions.** An annotation track (📄) with the candidate fusion breakpoints relative to the artificial fusion genome. See *Fusion tracks* below for more details.
- **Reads, fusions.** A reads track (📄) with the input reads mapped to the artificial fusion genome after refinement.
- **Reference (fusions).** A sequence track (📄) with the artificial fusion genome reference sequences.
- **Genes (fusions).** An annotation track (📄) with the gene regions for the candidate fusions on the artificial fusion genome.

- **mRNA (fusions)**. An annotation track  with the transcripts for the breakpoints of the candidate fusions on the artificial fusion genome.
- **CDS (fusions)**. An annotation track  with the CDS regions for the breakpoints of the candidate fusions on the artificial fusion genome. This is only produced when a CDS track was provided as input.
- **Primer (fusions)**. An annotation track  with the primer regions on the artificial fusion genome. This is only produced when a primer track was provided as input.
- **Fusion genes report**. A report  summarizing candidate fusions that have at least one breakpoint pair with a PASS status. See *Detect and Refine Fusion Genes report* below for more details.

Fusion tracks

The fusion annotation track outputs contain the fusion breakpoints. Fusions are named in the format 5' gene–3' gene. The table view , see section 27.3.2, shows one breakpoint per row. In addition to standard information, it contains the following:

- **Fusion plot**. A link to the fusion plot. See *Fusion plots* below for more details.
- **IPA gene view**. A link to QIAGEN Ingenuity Pathway Analysis with additional information about the fusion, if available.
- **Fusion number**. A unique identifier for the breakpoints associated with the same gene pair.
- **Fusion pair**. For each fusion number, a unique identifier for the corresponding breakpoint pairs.
- **5'/3' gene**. The name of the 5'/3' gene.
- **Breakpoint type**. 5' or 3'.
- **Filter**. If the breakpoint pair is assigned a PASS status, it is marked as such; otherwise, the reasons for not passing are provided. See *Filtered fusions* below for more details.
- **P-value**. The probability that an observation indicating a fusion event occurred randomly, in the absence of an actual fusion. This is calculated using a binomial test.
- **Z-score**. Converted from the p-value using the inverse distribution function for a standard Gaussian distribution.
- **Fusion crossing reads** and **Fusion spanning reads**. The number of reads supporting the breakpoint pair. Specifically:
 - Fusion crossing reads are reads that map across the breakpoint.
 - Fusion spanning reads are paired reads that map to both sides of the breakpoint but without crossing it. If paired reads span multiple breakpoints, the counts are distributed proportionally to the number of fusion crossing reads.
- **5'/3' read coverage** and **5'/3' spanning read coverage**. The number of reads mapping in the vicinity of the breakpoint pair. Specifically:

- Read coverage consists of reads that map on the WT reference genome across the breakpoint, in addition to the fusion crossing reads, as described above.
- Spanning read coverage consists of paired reads that map on the WT reference genome on both sides of the breakpoint but without crossing it, in addition to the fusion spanning reads, as described above. If paired reads span multiple breakpoints, the counts are distributed proportionally to the read coverage.
- **Translocation name.** The fusion description in COSMIC format (<https://cancer.sanger.ac.uk/cosmic/help/fusion/summary>). If the input mRNA track has transcript priorities, the transcript from 'Compatible transcripts' with the highest priority is used; otherwise the first transcript from 'Compatible transcripts' is used.
- **Compatible transcripts.** All transcripts from the input mRNA track that have an exon boundary within 'Maximum distance to known exon boundary' from the breakpoint. For novel exon boundaries, a name such as '10-gene27693-32015547-BEGINNING-0' is provided, indicating that the breakpoint at position 32015547 is associated with gene27693 on chromosome 10, near the start of an existing exon. The final '0' is a counter.
- **Exon skipping.** Indicates whether the fusion involves a single gene, corresponding to transcripts that have exons missing compared to the WT transcript annotation.
- **Fusion with novel exon boundaries.** Indicates whether one or both fusion breakpoints are at a novel exon boundary.
- **Found in-frame CDS.** Contains 'Yes' if at least one of the identified CDSs for the breakpoint pair is in frame. This considers only the last included exon of the 5' gene and the first included exon of the 3' gene, disregarding complex factors like frameshift mutations or stop codons from variants near the breakpoints. This column is only present when a CDS track was provided as input.
- **Breakpoint distance.** The physical distance between the breakpoints when on the same chromosome, otherwise -1. If the distance is small (e.g., <200,000), the fusion may be a false positive due to readthrough transcription.
- **Original chromosome** and **Original region.** Only present in tracks based on the artificial fusion genome.

Filtered fusions

The 'Filter' column contains the reasons why breakpoint pairs were not assigned a PASS status.

Breakpoint pairs for candidate fusions selected during filtering contain **Filtered during refinement**, and at least one of the following:

- **Few supporting reads.** The pair did not meet the 'Minimum number of supporting reads' threshold.
- **No support.** The pair was not supported by any read.
- **High p-value.** The pair exceeded the 'Maximum p-value' threshold.
- **Low Z-score.** The pair did not meet the 'Minimum Z-score' threshold.

If 'Include all fusions in the WT track output' was checked, the 'Fusion genes, WT' track additionally contains the detected potential fusions that were not selected as candidate fusions during filtering. These fusions contain **Filtered before refinement**, and at least one of the following:

- **Excluded by fusion filter (<table name>), Excluded by fusion filter (names), or Not included by fusion filter.** The fusion was:
 - Excluded based on a table named '<table name>' provided in 'Fusions for filtering (tables)'.
 - Excluded based on 'Fusions for filtering (names)'.
 - Not included based on either a table provided in 'Fusions for filtering (tables)' or 'Fusions for filtering (names)'.

Note that fusions filtered by the 'Gene filter action' are not included in the fusion track.

- **Too few unaligned ends.** The fusion did not meet the 'Minimum unaligned end count' threshold.
- **Exceeds number of breakpoints.** The fusion was excluded due to the limitation of 20 breakpoint pairs between the same two genes.
- **Exceeds promiscuity threshold.** The fusion exceeded the 'Promiscuity threshold'.
- **Exceeds number of fusions.** The fusion exceeded the 'Maximum number of fusions' threshold.
- **No fusion crossing reads.** Due to the absence of fusion crossing reads, the precise breakpoint location could not be identified. However, the fusion was supported by paired reads that mapped as broken pairs. For such fusions, the breakpoint region and 5'/3' gene are ill-defined:
 - The breakpoint 'Region' encompasses the entire region of the 5'/3' gene.
 - The 5' gene is the one that the first read in the pair maps to, if the read is mapped in the direction of the gene. Otherwise, the 5' gene is the one that the second read maps to.

Fusion plots

The fusion plot displays identified breakpoint pairs between one 5' and one 3' transcript for a given candidate fusion (figure 33.60). Only the breakpoint pairs compatible with the selected 5' and 3' transcripts are shown.

The 5' and 3' transcripts in the plot are selected as follows:

- They have an exon boundary at the breakpoint with the highest number of fusion crossing/spanning reads. If there are multiple such transcripts:
- They have the most exons with boundaries at breakpoints (i.e., the most purple connections). If there are multiple such transcripts:

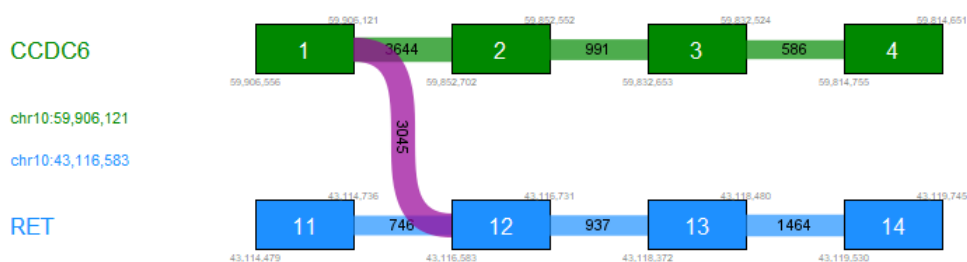


Figure 33.60: *Fusion plot showing a fusion between exon 1 of CCDC6 and exon 12 of RET.*

- They have the highest number of fusion crossing/spanning reads summed across the breakpoints.

The plot contains the following types of elements:

- **Green box.** An exon in the 5' transcript.
- **Blue box.** An exon in the 3' transcript.
- **Gray box.** An exon that is not in the selected transcript. The exon may be present in other transcripts, or may represent a novel exon.
- **Purple connection.** Fusion crossing/spanning reads supporting the fusion.
- **Green/blue connection.** Reads supporting the WT 5'/3' transcript.
- **Gray connection.** Reads supporting alternative splicing.
- **White vertical line within green or blue box.** Novel exon boundary.

Detect and Refine Fusion Genes report

The report has three sections: 'Summary', 'Unaligned Ends', and 'Fusions'.

The 'Summary' section contains a table with the sample name.

The 'Unaligned Ends' section contains statistics on the unaligned ends used to detect fusions:

- **Unaligned ends.** Number of identified unaligned ends.
- **Mapped unaligned ends.** Number of unaligned ends that could be mapped.
- **Unmapped unaligned ends.** Number of unaligned ends that could not be mapped.
- **Discarded breakpoints.** Number of discarded breakpoints. When breakpoints are adjacent because of two overlapping transcripts, one of the breakpoints is discarded.

The 'Fusions' section provides details for all candidate fusions that have at least one breakpoint pair with a PASS status (figure 33.61): an overview of the most supported breakpoint pair, a summary for up to 10 breakpoint pairs, sorted by support, and the fusion plot (see *Fusion plots* above).

3.3 CCDC6-RET

Fusion name	CCDC6-RET
5' gene	CCDC6
5' chromosome	10
3' gene	RET
3' chromosome	10
Reported transcript 5'	NM_005436.5
Reported transcript 3'	NM_020975.6
Translocation name	CCDC6{NM_005436.5}:r.1_435_RET{NM_020975.6}:r.2327_5617
P-value	0.00
Z-score	1,114.27
Fusion crossing reads	3,045
Fusion spanning reads	0
5' read coverage	3,866
5' spanning read coverage	2,823
3' read coverage	3,742
3' spanning read coverage	49

5' position	3' position	Translocation Name	Count
59,906,121	43,116,583	CCDC6{NM_005436.5}:r.1_435_RET{NM_020975.6}:r.2327_5617	3,045

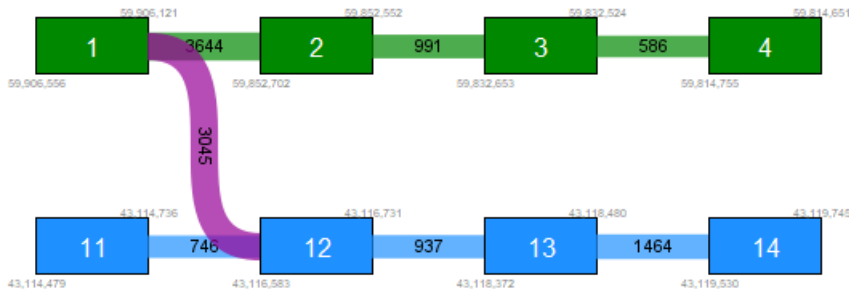


Figure 33.61: Example of a candidate fusion in the 'Fusions' section of the report.

Interpretation of fusion results

To evaluate the support for identified candidate fusions, the following outputs produced by the tool (see section 33.4.3) are most useful:

- The **Fusion genes, fusions** track.
- The **Reads, fusions** track.
- The *Fusion plots* (see section 33.4.3).

For a better overview of the results, we recommend making a **Track List** (📊) (see section 27.2) that contains both the **Fusion genes, fusions** track and the **Reads, fusions** track (figures 33.62 and 33.63).

Key aspects when evaluating the fusions are:

1. Reads should map uniquely across the fusion breakpoints

A high proportion of yellow reads mapping to the artificial fusion chromosome may indicate a false positive fusion (figure 33.62). Reads are yellow if they are not uniquely mapped and such reads are excluded when counting fusion supporting reads.

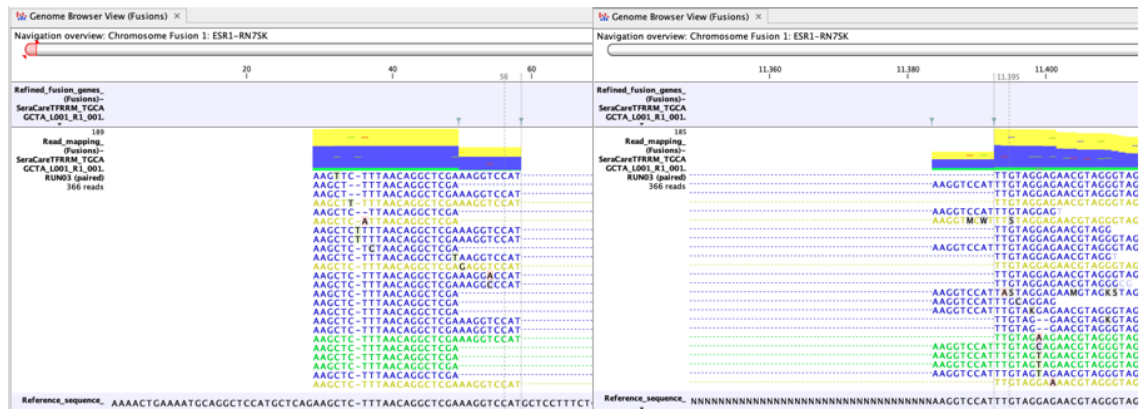


Figure 33.62: An example of a possible false positive fusion with many yellow reads.

2. There should be no sign of incomplete poly-A trimming

A high proportion of A-rich reads may indicate a false positive fusion (figure 33.63). Incomplete poly-A trimming can lead to reads with normal complexity on one side of the breakpoint and A-rich reads on the other side.

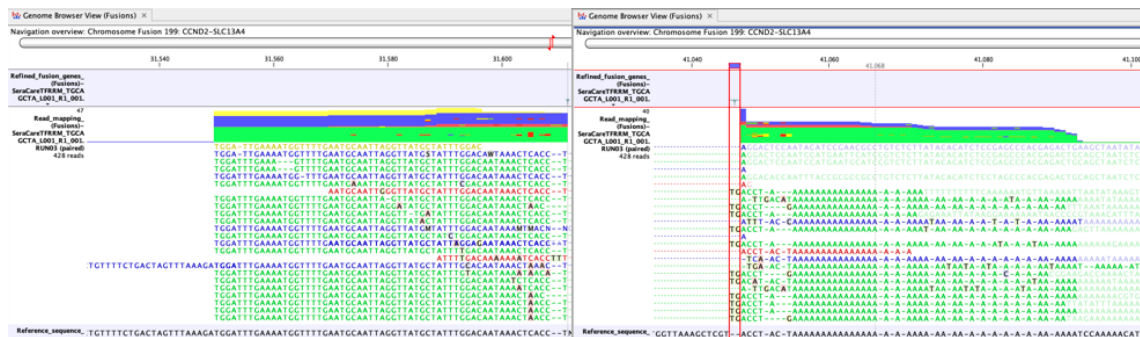


Figure 33.63: An example of a false positive fusion caused by incomplete poly-A trimming.

3. Fusions involving insertion of intronic sequences should be well supported

Fusions that involve insertion of intronic sequences may be false positives, especially if they have poor support. Insertion of an intronic sequence appears as a 'novel exon' indicated with a gray box in the fusion plot. Figure 33.64 shows an example of a possible false positive fusion where the novel exon has no support apart from the fusion crossing reads. Although there are 431 such reads, they do not fuse into a known exon boundary, but instead into the middle of an exon (white vertical line).

In contrast, figure 33.65 shows a true positive fusion that is supported by 48 fusion crossing reads, where the novel exon is supported by an additional 3 reads splicing into a known exon.

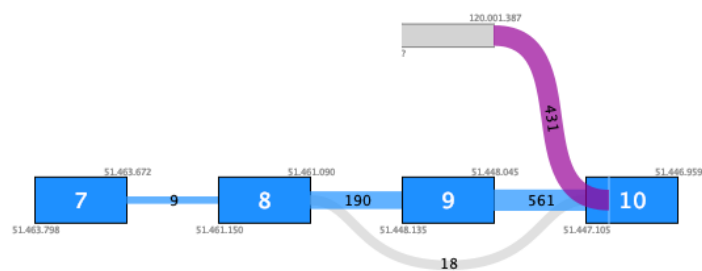


Figure 33.64: An example of a possible false positive fusion where the insertion of an intronic sequence is only supported by fusion crossing reads that are not at a known exon boundary.

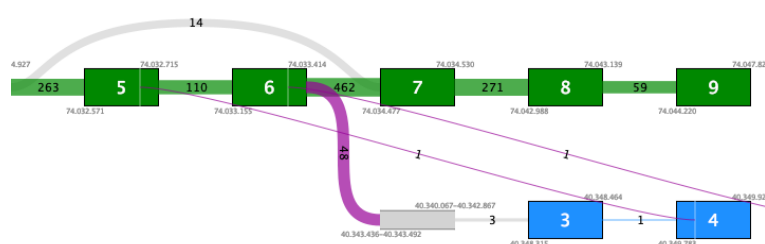


Figure 33.65: An example of a true positive fusion where the insertion of an intronic sequence is supported by reads other than fusion crossing reads.

33.4.4 Import Expression Data

Import Expression Data imports expression data from a tabular file. The importer produces one expression track per sample.

To run the importer, go to:

Tools | RNA-Seq and Small RNA Analysis (📁) | **RNA-Seq Tools** (🔧) | **Import Expression Data** (🇺🇸)

The following options can be configured (figure 33.66):

- **Table file** An Excel, CSV or TSV file containing the expression data (figure 33.67) where:
 - Columns represent samples and rows represent genes.
 - Feature (gene or transcript) names/IDs are in the first column.
 - Feature IDs represent a unique type of identifier, such as Ensembl or geneID.
 - Expression values are non-negative and of the same type, such as raw counts or TPM.
- **Table has sample names** When checked, sample names are read from the first row in the file and are used as the names for the output expression tracks. Otherwise, the expression tracks are named based on the table file name.
- **Expression type** The type of expression values contained in the file:
 - **Counts** The raw counts. This is recommended when available.

Figure 33.66: Configurable options for "Import Expression Data".

- **TPM** Transcripts per million.
- **RPKM** Reads Per kilobase of exon model per million mapped reads.
- **Minimum count** The smallest raw count found in the original expression data, typically 1 for unfiltered data. It is used for calculating the counts in the expression tracks when the expression type is TPM or RPKM.
 If the minimum count is unknown, it should ideally be set to be in the same order of magnitude as the original raw counts. Setting the minimum count too low can lead to a loss of precision, while setting it too high can create a false sense of precision, causing genes with equal expression levels to appear different due to small variations in exon lengths.
- **Feature types** The types of features for which expression values are contained in the file:
 - **Genes with transcripts** The feature names/IDs are matched against the gene track. The corresponding transcripts are used to calculate exon lengths when converting between counts and TPM/RPKM. The importer outputs gene expression tracks.
 - **Genes** The feature names/IDs are matched against the gene track. The gene lengths are used when converting between counts and TPM/RPKM. The importer outputs gene expression tracks.

- **Transcripts** The feature names/IDs are matched against the mRNA track. The exon lengths are used when converting between counts and TPM/RPKM. The importer outputs transcript expression tracks.
- **Gene track** The gene track used for matching the genes names/IDs in the file.
- **mRNA track** The mRNA track used for converting between counts and TPM/RPKM or for matching the transcript names/IDs in the file.
- **Calculate expression for genes without transcripts** When checked, the gene lengths are used when converting between counts and TPM/RPKM for genes without a corresponding transcript. This option is only applicable when the feature type is set to **Genes with transcripts**.
- **Unmatched features** When matching feature names/IDs from the file against the gene or mRNA track, some features may remain unmatched either because they are not found in the track or because they are ambiguous, matching multiple features in the track. The following options are available:
 - **Include** Unmatched features are included without information on chromosome or genomic position (region). This option can only be used when importing raw counts.
 - **Ignore** Unmatched features are ignored.
 - **Fail** Unmatched features cause the importer to fail.

	A	B	C	D	E
1	Name	23T	23N	26T	26N
2	DDX11L1	0	0.09587955	0	0
3	WASH7P	0.53207257	0.73096019	0.67368009	0.21937677
4	MIR1302-10	0	0	0	0
5	FAM138A	0.12064385	0.06905845	0.904455	0
6	OR4G4P	0	0	0	0
7	OR4G11P	0	0	0	0
8	OR4F5	0	0	1.12094619	0
9	RP11-34P13.7	0.14422163	0.02358707	1.66816069	0.02548434
10	RP11-34P13.8	0.61323481	0	0.50153024	0
11	CICP27	0	0	0	0
12	AL627309.1	1.18823541	1.5185073	0.80103692	3.2129422
13	RP11-34P13.15	0	0	0	0
14	RP11-34P13.16	0	0	0	0

Figure 33.67: Expression data for four samples. The first column contains Ensembl gene names and the expressions are RPKM values.

Importing sample metadata

Certain types of analysis, such as **Differential Expression for RNA-Seq**, see section 33.6.4, require sample metadata. The sample metadata can be imported into a CLC Metadata Table from an Excel, CSV or TSV file, see section 13.1.1 for details. Once the metadata is imported, it can be associated with the imported expression tracks, see section 13.2 for details.

When used in a workflow, **Import Expression Data** can both import the metadata and associate the resulting CLC Metadata Table with the expression tracks. The CLC Metadata Table is used by downstream tools that require metadata, such as **Differential Expression for RNA-Seq**, see section 33.6.4. The following additional options can be configured:

- **Import metadata** Enables import of sample metadata.
- **Metadata file** An Excel, CSV or TSV file containing the metadata (figure 33.68).

	A	B	C
1	Sample	group	Treatment
2	26T	Tumor	0
3	23T	Tumor	1
4	26N	Normal	0
5	23N	Normal	0

Figure 33.68: Sample metadata for the expression data from figure 33.67.

33.5 Expression Plots

A number of tools are available for visualizing the outputs from the RNA-Seq and miRNA Analysis.

33.5.1 PCA for RNA-Seq

Principal Component Analysis makes it possible to project a high-dimensional dataset (where the number of dimensions equals the number of genes or transcripts) onto two or three dimensions. This helps in identifying outlying samples for quality control, and gives a feeling for the principal causes of variation in a dataset. The analysis proceeds by transforming a large set of variables (in this case, the counts for each individual gene or transcript) to a smaller set of orthogonal principal components. The first principal component specifies the direction with the largest variability in the data, the second component is the direction with the second largest variation, and so on.


The **PCA for RNA-Seq** tool clusters samples in 2D or 3D. Known metadata about each sample is added as an overlay. In addition, the following filtering and normalization are performed:

- 'log CPM' (Counts per Million) values are calculated for each gene. The CPM calculation uses the effective library sizes as calculated by the TMM normalization.
- After this, a Z-score normalization is performed across samples for each gene: the counts for each gene are mean centered, and scaled to unit variance.
- Genes or transcripts with zero expression across all samples or invalid values (NaN or +/- Infinity) are removed.

For more detail about these steps, see section 33.1.

To start the analysis:

Tools | RNA-Seq and Small RNA Analysis | Expression Plots | PCA for RNA-Seq

Select a number of expression tracks  and click **Next**. The tool will generate a PCA plot that can be visualized in 2D and 3D. The plot has two table views, each with a column per principal component.

The first table shows the loadings of each gene. These are the unit eigenvectors multiplied by the square root of the eigenvalues (note that in some contexts "loadings" is instead used to describe the unscaled unit eigenvectors). The genes with large positive and negative loadings contribute most to the direction of the principal component. Loadings can also be compared across principal components: the genes with largest positive and negative loadings are those that explain most variation in the data.

The second table shows the coordinates of the samples in the plot. This allows the plot to be redrawn in other software.

Principal component analysis plot (2D)

The default view is a two-dimensional principal component plot as shown in figure 33.69.

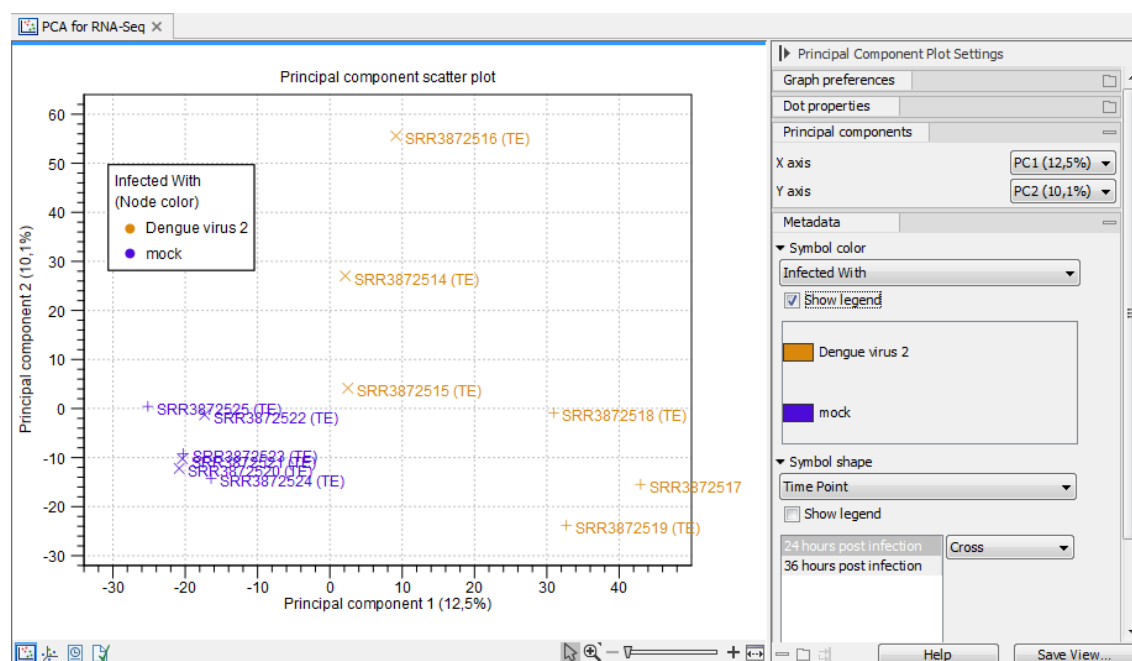


Figure 33.69: A principal component plot.

The plot shows the projection of the samples onto the two-dimensional space spanned by the first and second principal components of the covariance matrix. The expression levels used as input are normalized log CPM values, see section 33.1.

The view settings can be adjusted using the **Side Panel**. Under **Graph preferences**, you can adjust the general properties of the plot.

- **Lock axes** This will always show the axes even though the plot is zoomed to a detailed level.

- **Frame** Shows a frame around the graph.
- **Tick type** Determines whether tick lines should be shown outside or inside the frame.
- **Tick lines at** Choosing Major ticks will show a grid behind the graph.
- **Horizontal axis range** Sets the range of the horizontal axis (x axis). Enter a value in **Min** and **Max**, and press Enter. This will update the view. If you wait a few seconds without pressing Enter, the view will also be updated.
- **Vertical axis range** Sets the range of the vertical axis (y axis). Enter a value in **Min** and **Max**, and press Enter. This will update the view. If you wait a few seconds without pressing Enter, the view will also be updated.
- **y = 0 axis** Draws a line where $y = 0$ with options for adjusting the line appearance.

Below the general preferences, you find the **Dot properties**:


- **Drop down menu** In this you select the expression tracks to which following choices apply.
- **Dot type** Allows you to choose between different dot types.
- **Dot color** Click the color box to select a color.
- **Show name** This will show a label with the name of the sample next to the dot.

Note that the Dot properties may be overridden when the Metadata options are used to control the visual appearance (see below).

The **Principal Components** group determines which two principal components are used in the 2D plot. By default, the first principal component is shown for the X axis and the second principal component is shown for the Y axis. The value after the principal component identifier (for example "PC1 (72.5 %)") displays the amount of variance explained by this particular principal component.

The **Metadata** group allows metadata associated with the Expression tracks to be visualized in a number of ways:

- **Symbol color** Colors are assigned based on a categorical factor in the metadata table.
- **Symbol shape** Shape is assigned based on a categorical factor in the metadata table.
- **Label text** Dots are labeled according to the values in a given metadata column.
- **Legend font settings** contains options to adjust the display of labels.

The graph and axes titles can be edited simply by clicking them with the mouse. These changes will be saved when you **Save** () the graph - whereas the changes in the **Side Panel** need to be saved explicitly (see section 4.6).

Principal component analysis plot (3D)

The principal component plot may also be displayed in 3D. The 3D view is accessible through the view buttons at the bottom of the panel.

If you have problems viewing the 3D plot, please check your system matches the requirements for 3D viewers. See section [1.3](#).

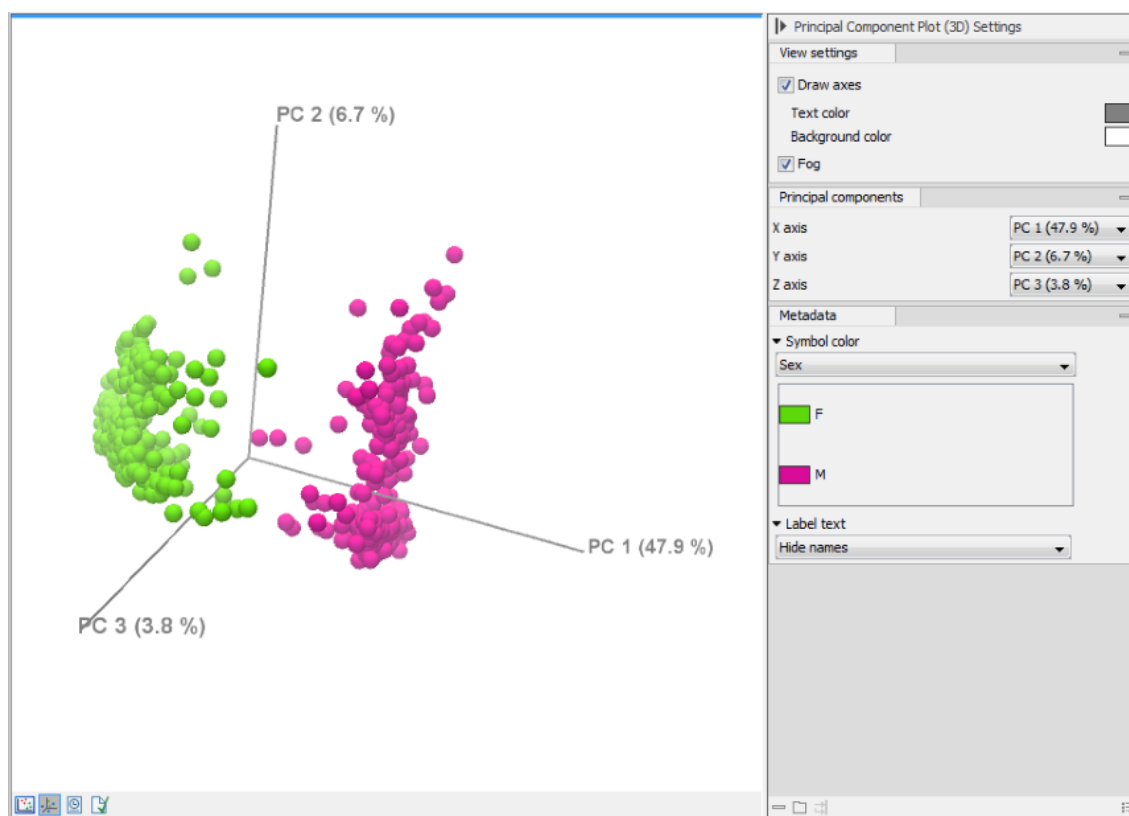


Figure 33.70: A principal component 3D plot.

The 3D view may be rotated by dragging on the view with the left mouse button pressed. It is possible to pan the view by dragging with the right mouse button pressed. Zooming can be done either using the mouse scroll wheel, or by dragging with both left and right mouse button pressed. It is also possible to center and zoom to a sample simply by clicking on it. Clicking outside any sample (or clicking with the right mouse button) restores the zoom and centering.

The **Side Panel** offers a number of options to change the appearance of the 3D principal component plot:

The **View settings** group makes it possible to toggle the coordinate system on and off, and adjust the text and background color. It is also possible to enable **Fog**, which dims distant objects in order to improve the depth perception.

The **Principal Components** group determines which principal components are used in the 3D plot. The value after the principal component identifier (for example "PC 1 (72.5 %)") displays the amount of variance explained by this particular principal component.


The **Metadata** group allows metadata associated with the Expression tracks to be visualized using color or as text:

- **Symbol color** Colors are assigned based on a categorical factor in the metadata table.
- **Label text** Samples are labeled according to the values in a given metadata column. If 'Show names' is selected, the samples will be labeled according to their name (as shown in the **Navigation Area**).

To save the current view as an image, press the **Graphics** button in the Workbench toolbar. Next, select the location where you wish to save the image, select file format (PNG, JPEG, or TIFF), and provide a name, if you wish to use another name than the default name.

It is possible to save the current view settings (including camera settings) using the **Side Panel** view settings options, see section 4.6.

33.5.2 Create Sample Level Heat Map for RNA-Seq

Create Sample Level Heat Map for RNA-Seq takes at least two **Expression tracks**  and creates a heat map that displays the distances between samples with respect to their feature expression profiles. Each track represents a sample in the heat map. See section 33.5.2 for details.

To run the tool, go to:

Tools | RNA-Seq and Small RNA Analysis  | **Expression Plots**  | **Create Sample Level Heat Map for RNA-Seq** 

The following options for performing hierarchical clustering for the samples (see section 33.5.3) can be configured in the **Distances** dialog (figure 33.71):

- **Distance measure** for calculating distances between samples:
 - **Euclidean distance**
 - **1 - Pearson correlation**
 - **Manhattan distance**
- **Cluster linkage** for calculating distances between clusters:
 - **Single linkage**
 - **Average linkage**
 - **Complete linkage**

The following options for filtering the features used to create the heat map can be configured in the **Feature filters** dialog (figure 33.72):

- **No filtering.** All features are kept.
- **Fixed number of features.** A specified number of features are kept.

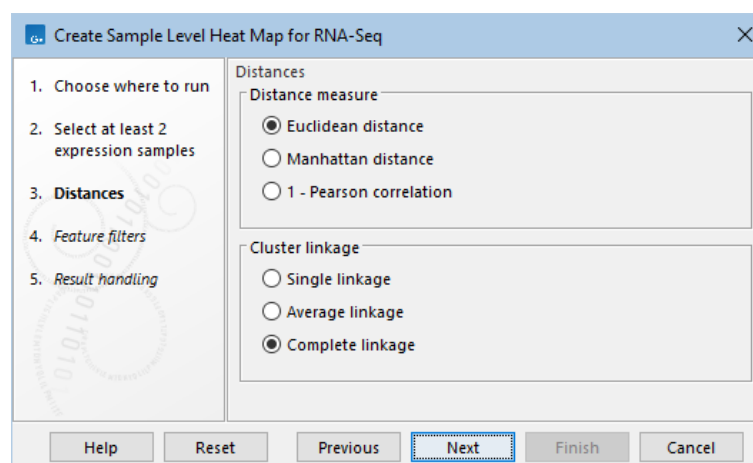





Figure 33.71: Distance options used to create the heat map.

- **Number of features.** This number of features with the highest index of dispersion (variance to mean ratio) are kept. Raw count values (not normalized) are used to calculate this index.
- **Minimum counts in at least one sample.** Only features with at least this raw count value in one sample are kept.
- **Filter features by statistics.** Features that are differentially expressed according to the specified thresholds are kept.
 - **Statistical comparison.** A **Statistical Comparison Track** () see section 33.6.5.
 - **Minimum absolute fold change.** Only features with at least this absolute fold change are kept.
 - **Threshold.** Only features with at most this p-value are kept. The type of the p-value can be set to **P-value**, **FDR p-value**, or **Bonferroni**.
- **Specify features.** A set of specified features are kept.
 - **Feature track.** An **Annotation Track** () All features in this track are kept.
 - **Feature names.** Features with names in the list of case-sensitive names are kept. Any white-space characters, comma, and semicolon are accepted as separators.

The sample level heat map view

In the heat map (figure 33.73), each column and row represents a sample. The distance between two samples dictates the color of the corresponding cell and can be seen in the table view (). See section 33.5.3 for details on how expressions are normalized and distances are calculated.

The following can be configured in the Side Panel, see section 2.1.6:

Heat map. General options for configuring layout and coloring.

- **Lock width/height to window.** When checked, the width/height of the heat map is locked to the window width/height when zooming in, ensuring that all columns/rows remain visible. Note that if both options are checked, zooming is not possible.

Figure 33.72: Feature filtering options.

- **Lock headers and footers.** When checked, the shown names and trees will be visible when zooming in.
- **Colors.** The gradient depicts the distances, with low values on the left side and high values on the right side. See section 2.1.6 for how to customize the gradient.

Columns. Options for configuring the display of the heat map columns.

- **Show above/below.** When checked, the names/colors legend/tree are shown above and/or below the heat map.
- **Tree size.** The size of the tree. Selecting **Full tree** adjusts the size automatically according to the tree depth.
- **Order by.** The order of the heat map columns. The following options are available:
 - **Tree.** Uses the hierarchical tree structure to determine the order.
 - **Samples.** Uses the order of the inputs to the tool.
 - **Active Metadata layers.** Uses the order from the selected metadata layers.
 - **Metadata table column.** Uses the chosen metadata table column to determine the order.

Choosing an order other than **Tree** hides the tree.

Rows. Options for configuring the display of the heat map rows.

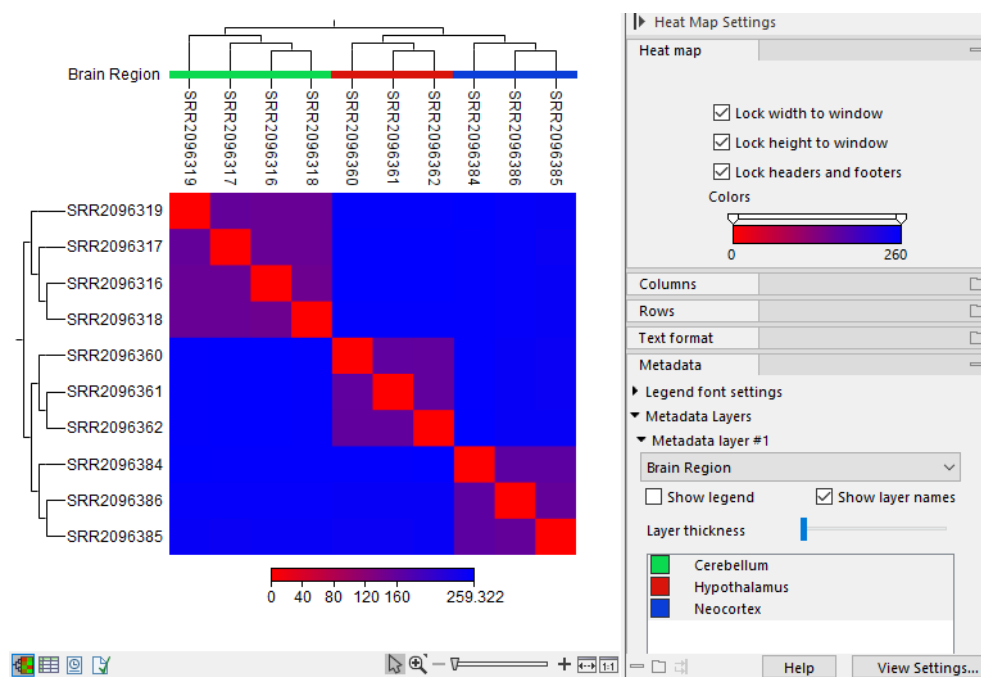


Figure 33.73: The sample level heat map.

- **Show left/right.** When checked, the names/colors legend/tree are shown above and/or below the heat map.
- **Tree size.** The size of the tree. Selecting **Full tree** adjusts the size automatically according to the tree depth.
- **Optimize tree layout.** If checked, the rows are reordered, to the extent possible, such that the highest distances form a diagonal line from the top-left to the bottom-right of the heat map.

Metadata. Options for visualizing metadata associated with the **Expression tracks**

- **Legend font settings.** Options for configuring the legend font.
- **Metadata layer #1.** Selecting a metadata table column adds a color bar above the heat map, indicating the corresponding values. The values in the metadata layer can be reordered using drag-and-drop. This impacts the order in the legend and/or columns, as relevant. As many metadata layers can be added as needed. For each metadata layer, the following can be adjusted:
 - **Show legend.** When checked, a legend is shown. The legend can be resized and it can be moved using drag-and-drop.
 - **Show layer names.** When checked, the names of the metadata layers are shown to the left of the color bar.

33.5.3 Create Feature Level Heat Map for RNA-Seq

Create Feature Level Heat Map for RNA-Seq takes at least two **Expression tracks** and creates a heat map that displays feature expression levels across these tracks. Each track

represents a sample in the heat map. See section 33.5.3 for details.

To run the tool, go to:

Tools | RNA-Seq and Small RNA Analysis (📄) | **Expression Plots** (📊) | **Create Feature Level Heat Map for RNA-Seq** (🔥)

The following options for performing hierarchical clustering for the samples and features (see section 33.5.3) can be configured in the **Distances** dialog (figure 33.74):

- **Distance measure** for calculating distances between samples and features:
 - **Euclidean distance**
 - **1 - Pearson correlation**
 - **Manhattan distance**
- **Cluster linkage** for calculating distances between clusters:
 - **Single linkage**
 - **Average linkage**
 - **Complete linkage**

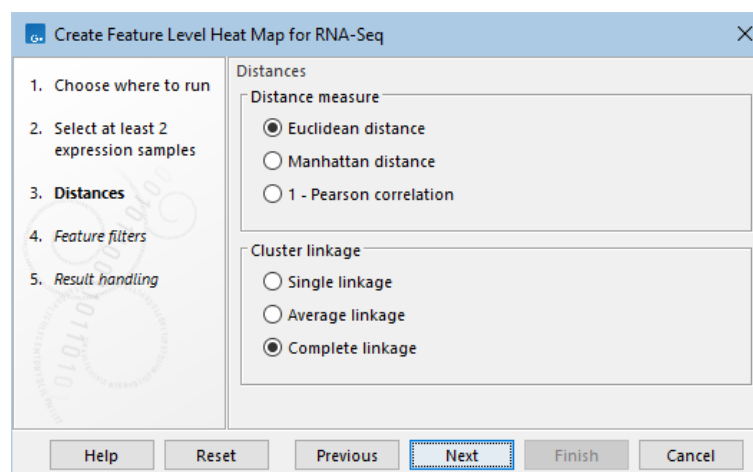




Figure 33.74: Distance options used to create the heat map.

Genomes often have too many features to visualize all of them meaningfully, and clustering such a large number is time-consuming. Therefore, it is recommended to reduce the number of features by configuring the options in the **Feature filters** dialog (figure 33.75):

- **No filtering.** All features are kept.
- **Fixed number of features.** A specified number of features are kept.
 - **Number of features.** This number of features with the highest index of dispersion (variance to mean ratio) are kept. Raw count values (not normalized) are used to calculate this index.
 - **Minimum counts in at least one sample.** Only features with at least this raw count value in one sample are kept.

- **Filter features by statistics.** Features that are differentially expressed according to the specified thresholds are kept.
 - **Statistical comparison.** A **Statistical Comparison Track** () see section 33.6.5.
 - **Minimum absolute fold change.** Only features with at least this absolute fold change are kept.
 - **Threshold.** Only features with at most this p-value are kept. The type of the p-value can be set to **P-value**, **FDR p-value**, or **Bonferroni**.
- **Specify features.** A set of specified features are kept.
 - **Feature track.** An **Annotation Track** () All features in this track are kept.
 - **Feature names.** Features with names in the list of case-sensitive names are kept. Any white-space characters, comma, and semicolon are accepted as separators.

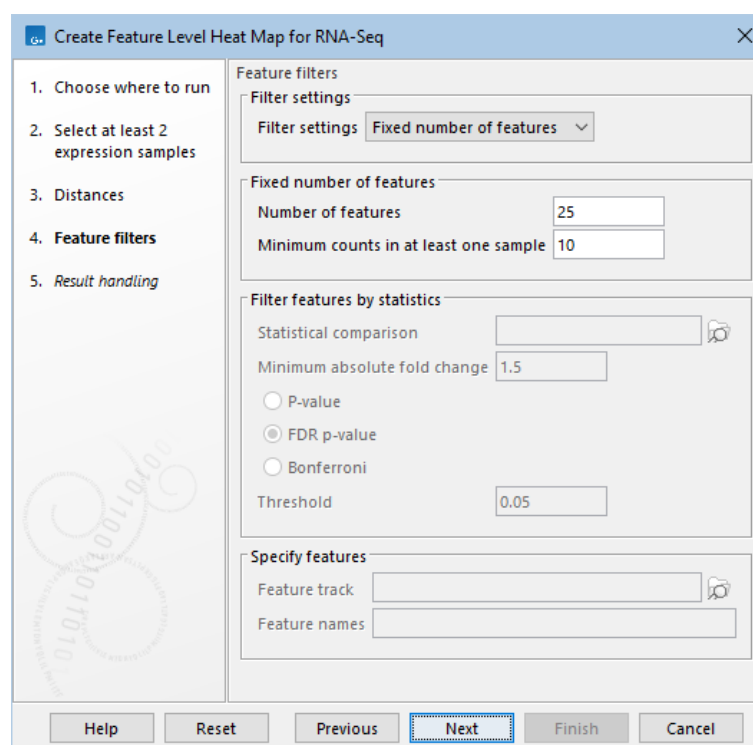



Figure 33.75: Feature filtering options.

The heat map view

In the heat map (figure 33.76), each column represents a sample, and each row represents a feature (gene or transcript). The normalized expression level for a sample and feature dictates the color of the corresponding cell and can be seen in the table view () See section 33.5.3 for details on how expressions are normalized.

The following can be configured in the Side Panel, see section 2.1.6:

Heat map. General options for configuring layout and coloring.

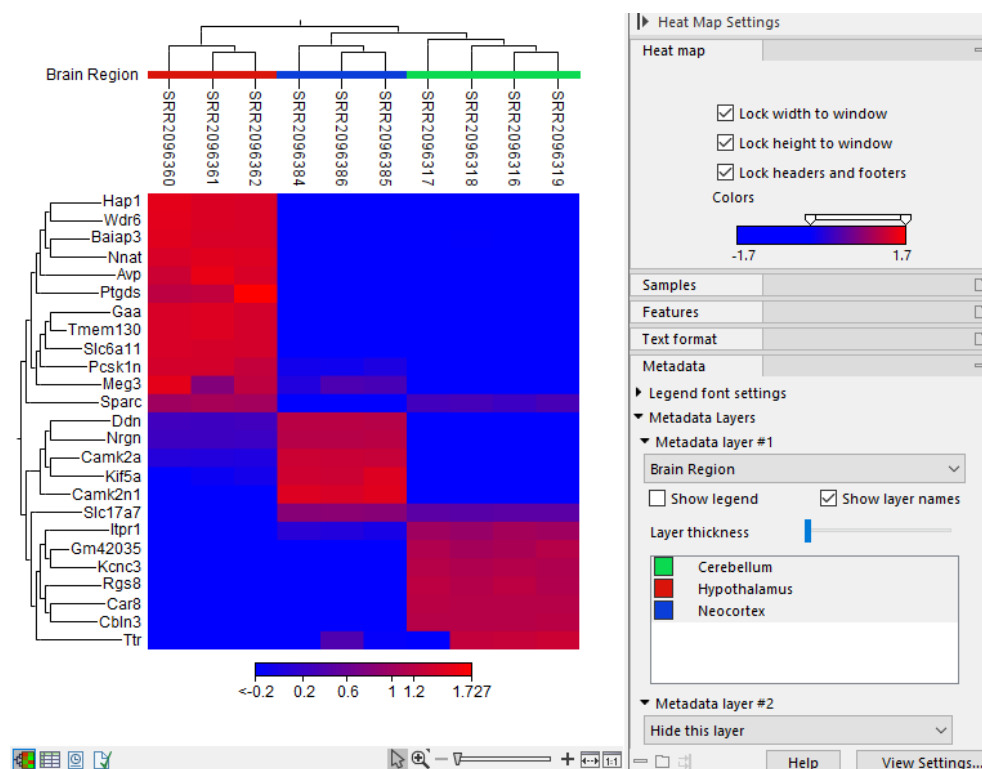


Figure 33.76: The feature level heat map.

- **Lock width/height to window.** When checked, the width/height of the heat map is locked to the window width/height when zooming in, ensuring that all columns/rows remain visible. Note that if both options are checked, zooming is not possible.
- **Lock headers and footers.** When checked, the shown names and trees will be visible when zooming in.
- **Colors.** The gradient depicts the expression levels, with low values on the left side and high values on the right side. See section 2.1.6 for how to customize the gradient.


Samples. Options for configuring the display of the heat map columns.

- **Show above/below.** When checked, the names/colors legend/tree are shown above and/or below the heat map.
- **Tree size.** The size of the tree. Selecting **Full tree** adjusts the size automatically according to the tree depth.
- **Order by.** The order of the heat map columns. The following options are available:
 - **Tree.** Uses the hierarchical tree structure to determine the order.
 - **Samples.** Uses the order of the inputs to the tool.
 - **Active Metadata layers.** Uses the order from the selected metadata layers.
 - **Metadata table column.** Uses the chosen metadata table column to determine the order.

Choosing an order other than **Tree** hides the tree.

Features. Options for configuring the display of the heat map rows.

- **Show left/right.** When checked, the names/colors legend/tree are shown above and/or below the heat map.
- **Tree size.** The size of the tree. Selecting **Full tree** adjusts the size automatically according to the tree depth.
- **Optimize tree layout.** If checked, the rows are reordered, to the extent possible, such that the highest expression levels form a diagonal line from the top-left to the bottom-right of the heat map.

Metadata. Options for visualizing metadata associated with the **Expression tracks** .

- **Legend font settings.** Options for configuring the legend font.
- **Metadata layer #1.** Selecting a metadata table column adds a color bar above the heat map, indicating the corresponding values. The values in the metadata layer can be reordered using drag-and-drop. This impacts the order in the legend and/or columns, as relevant. As many metadata layers can be added as needed. For each metadata layer, the following can be adjusted:
 - **Show legend.** When checked, a legend is shown. The legend can be resized and it can be moved using drag-and-drop.
 - **Show layer names.** When checked, the names of the metadata layers are shown to the left of the color bar.

Normalization and clustering

The expression values are filtered and normalized as follows:

- Features with zero expression across all samples, or invalid values (NaN or +/- Infinity), are removed.
- logCPM values are calculated for each feature.
- A Z-Score normalization is applied on the logCPM values.

See section [33.1](#) for more details.

The samples and features, as relevant, are hierarchically clustered based on the similarity of their expression profiles, as follows:

1. Create clusters containing one sample/feature.
2. Calculate the distances between all clusters.
3. Merge the two closest clusters into one.

4. Repeat until only one cluster remains, containing all the samples/features.

The hierarchical cluster forms tree structures displayed along the rows and columns of the heat map. The tree branch lengths represent the distances between clusters.

The distance between two clusters is determined using one of the following linkage types:

- **Single linkage.** The distance between the two closest samples/features in the two clusters.
- **Average linkage.** The average distance between samples/features in the first cluster and samples/features in the second cluster.
- **Complete linkage.** The distance between the two farthest samples/features in the two clusters.

The distance between two samples/features is calculated using one of the following distance measures:

- **Euclidean distance.** The length of the segment connecting two points. If $u = (u_1, u_2, \dots, u_n)$ and $v = (v_1, v_2, \dots, v_n)$, then the Euclidean distance between u and v is

$$|u - v| = \sqrt{\sum_{i=1}^n (u_i - v_i)^2}.$$

- **Manhattan distance.** The distance between two points measured along axes at right angles. If $u = (u_1, u_2, \dots, u_n)$ and $v = (v_1, v_2, \dots, v_n)$, then the Manhattan distance between u and v is

$$|u - v| = \sum_{i=1}^n |u_i - v_i|.$$

- **1 - Pearson correlation.** The Pearson correlation coefficient between $x = (x_1, x_2, \dots, x_n)$ and $y = (y_1, y_2, \dots, y_n)$ is defined as

$$r = \frac{1}{n-1} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{s_x} \right) \cdot \left(\frac{y_i - \bar{y}}{s_y} \right)$$

where \bar{x}/\bar{y} and s_x/s_y are the average and sample standard deviation, respectively, of the values in x/y values.

The Pearson correlation coefficient ranges from -1 to 1, with high absolute values indicating strong correlation, and values near 0 suggesting little to no relationship between the elements.

Using $1 - |\text{Pearson correlation}|$ as the distance measure ensures that highly correlated elements have a shorter distance, while elements with low correlation are farther apart.

33.5.4 Create K-medoids Clustering for RNA-Seq

In a k-medoids clustering, features are clustered into k separate clusters. The procedure seeks to assign features to clusters such that distances between features of the same cluster are small, while distances between clusters are large.

The output of the tool is a Clustering Collection (🌐). The clusters in the Clustering Collection can be viewed together as a Sankey plot (📊) or individually as graphs (📈).

To perform a k-medoids clustering:

Tools | RNA-Seq and Small RNA Analysis (📄) | Expression Plots (📊) | Create K-medoids Clustering for RNA-Seq (📈)

Select at least two expression tracks (📊), or miRNA expression tables (📄)/ (📄).

Click **Next** to display a dialog as shown in figure 33.77.



Figure 33.77: Parameters for *k*-medoids clustering.

The parameters are:

- **Number of clusters.** The maximum number of clusters to cluster features into: the final number of clusters will be smaller than this if there are fewer features than clusters.
- **Metadata table** (Optional) The metadata table describing the factors for the selected inputs.
- **Perform a separate clustering for each** (Optional) one of the factors from the metadata table. A separate *k*-medoids clustering is performed for each group in this factor. The clusters for each group form separate columns in the Sankey plot. This is useful when looking for genes whose expression pattern changes in a certain way between groups. The groups could, for example, represent different treatments.
- **Group samples by** (Optional) One of the factors from the metadata table. The distances between samples for a feature are calculated using the group means. If this is left blank, then distances will be calculated using all the individual values of the samples.


- **Order groups** (Optional) For the chosen **Group samples by**, specify the order of the groups. The ordering controls the x-axis of the expression graphs. This is useful when the data has a natural ordering, such as a time series. If only some groups are ordered here, then these will come first, and the remaining groups will be added at the end.

Genomes often have too many features to visualize all of them meaningfully, and clustering such a large number is time-consuming. Therefore, it's recommended to reduce the number of features by using the **Feature filters**:

- **No filtering.** All features are kept.
- **Fixed number of features.** A specified number of features are kept.
 - **Number of features.** This number of features with the highest index of dispersion (variance to mean ratio) are kept. Raw count values (not normalized) are used to calculate this index.
 - **Minimum counts in at least one sample.** Only features with at least this raw count value in one sample are kept.
- **Filter features by statistics.** Features that are differentially expressed according to the specified thresholds are kept.
 - **Statistical comparison.** A **Statistical Comparison Track** () , see section 33.6.5.
 - **Minimum absolute fold change.** Only features with at least this absolute fold change are kept.
 - **Threshold.** Only features with at most this p-value are kept. The type of the p-value can be set to **P-value**, **FDR p-value**, or **Bonferroni**.
- **Specify features.** A set of specified features are kept.
 - **Feature track.** An **Annotation Track** () . All features in this track are kept.
 - **Feature names.** Features with names in the list of case-sensitive names are kept. Any white-space characters, comma, and semicolon are accepted as separators.

We only recommend using **Fixed number of features** for exploratory analysis. This is because, while the chosen features have the most variable expression among all the samples, the variation may not be of interest: for example, maybe there is a large variability across different time points in a time series, but this is the same in both treatment and control groups.

Viewing the result of k-medoids clustering

The result of the clustering is a Clustering Collection (). When opened, this shows a Sankey plot view with one column for each condition chosen in **Perform a separate clustering for each** and one row for each cluster, see the top part of figure 33.78. A thumbnail in each cluster shows the average expression for the features in that cluster.

The Sankey plot view is linked to a graph view. Clicking on a thumbnail shows the expression of all the features in the cluster in the linked graph view. Clicking two thumbnails while holding down the Ctrl key (⌘ on Mac) shows the features in common to the two selected clusters, see the bottom part of figure 33.78.

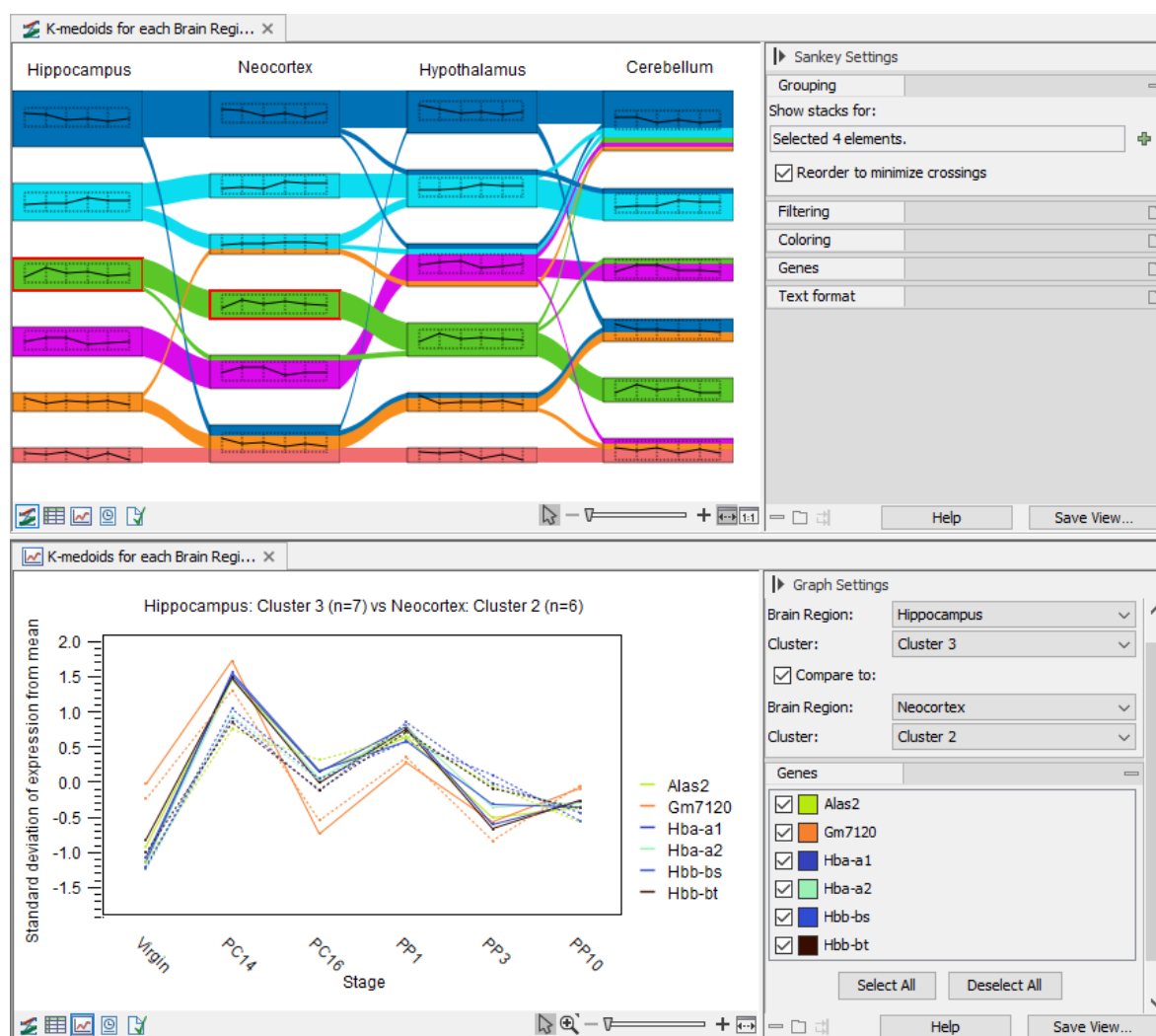


Figure 33.78: Sankey plot example. The data set contains four mouse brain tissues and 6 time points, from virgin to postpartum (GEO accession GSE70732). Top: Features in each brain tissue has been divided into the same number of clusters and the flows indicate how the features change the clusters they belong to in the different tissues. Two clusters are selected, as indicated by the red border. Bottom: The line graph of the clusters to be compared shows the feature expression across the time points for the features found in both selected clusters. Here, cluster 3 in Hippocampus and cluster 2 in Neocortex are compared.

Thumbnails and flows have a right-click menu which, for example, allows selecting the corresponding features in other views, such as an expression browser, heat map, or the volcano plot of a statistical comparison track.

If you close one of the two views, you can re-open it by holding down the Ctrl key (⌘ on Mac) and clicking on the icon for the view you wish to re-open.

Side panel settings

Adjustments to the Sankey plot and line graph can be done from the side panel.

It is possible to re-order or remove the columns in the Sankey plot, to remove clusters from a column, to select where flows start (defaults to the first column), and to remove all flows except for those originating in specified clusters of the start column. Furthermore, selected features can be highlighted.

- To **re-order or remove columns**, click the (+) button in the side panel in "Show stacks for:" under "Grouping" and use the arrow buttons to order and select columns, see figure 33.79.

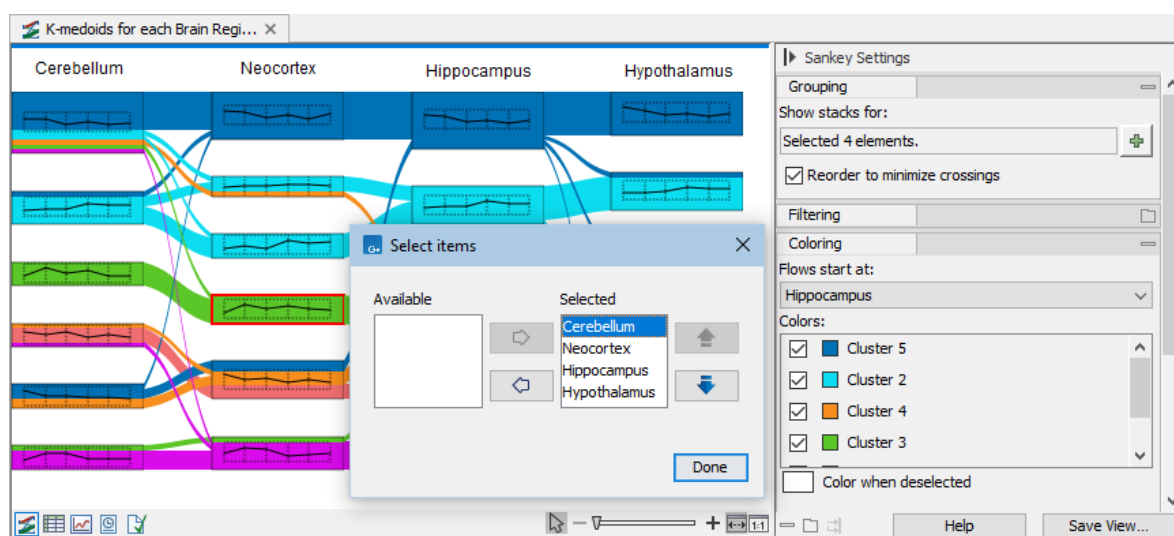


Figure 33.79: Changing the columns order. Note that colors are determined by the column selected in "Flows start at:" in the side panel.

- To **select where flows start**, select the desired column in "Flows start at" in the side panel under "Coloring".
- To **remove clusters from a column**:
 - Click the cluster to select it, right-click and choose "Remove Selected".
 - In the side panel, under "Filtering", click the (+) button for the relevant column, and use the arrow buttons to choose the clusters.
- To **remove flows**:
 - Click a cluster in the first column to select it, right-click and choose "Color only from selected".
 - In the side panel, under "Coloring", tick the clusters for which to retain coloring. Flows from the other clusters are colored using "Color when deselected", defaulting to white and making the deselected flows invisible.

Figure 33.80 show an example using filters and colors to highlight features from specific clusters.

- To **highlight features**, use the "Select genes to trace in plot" under "Genes" in the side panel. Use space to get the full list of available features. Selected features will be highlighted in the Sankey plot and represented in bold in the line graphs, see figure 33.81

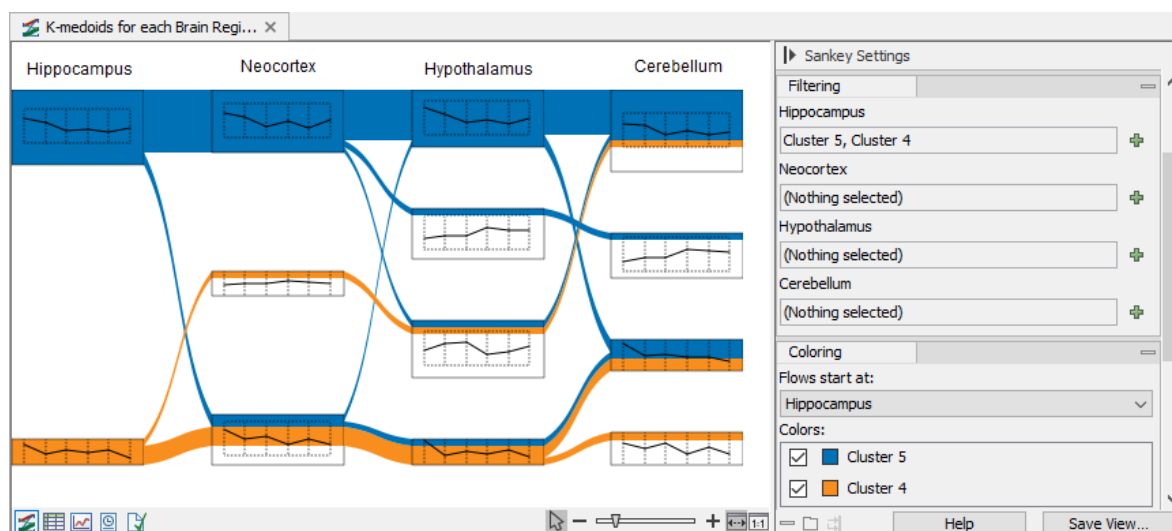


Figure 33.80: Filtering and coloring can help highlighting features flow across clusters.

The line graph allows two types of adjustments from the side panel, selection of "Cluster" and "Genes". Only features present in the selected clusters are available in the features pick list and they can be added and removed from the line graph using the tick boxes.

Line graph representation

The line representations are dependent on selections:

- **One cluster is selected:** all lines are solid.
- **Two clusters are compared:** the expression from the first cluster is represented by solid lines, and the expression from the second cluster is represented by dashed lines.
- **Features are selected:** the line style (solid or dashed) is as described above for all features. The selected features are highlighted in bold the side panel under "Genes" and their corresponding lines, from both clusters if two clusters are compared, are thicker, see figure 33.81.

The K-Medoids algorithm

K-medoids clustering is computed using the PAM-algorithm (PAM is short for Partitioning Around Medoids). The PAM-algorithm is based on the search for k representatives (called medoids) among all elements of the dataset. When having found k representatives, k clusters are generated by assigning each element to its nearest medoid. The algorithm first looks for a good initial set of medoids (the BUILD phase). Then it finds a local minimum for the objective function:

$$V = \sum_{i=1}^k \sum_{x_j \in S_i} (x_j - c_i)^2$$

where there are k clusters $S_i, i = 1, 2, \dots, k$ and c_i is the medoid of S_i . This solution implies that there is no single switch of an object with a medoid that will decrease the objective (this is called the SWAP phase). The PAM-algorithm is described in [Kaufman and Rousseeuw, 1990].

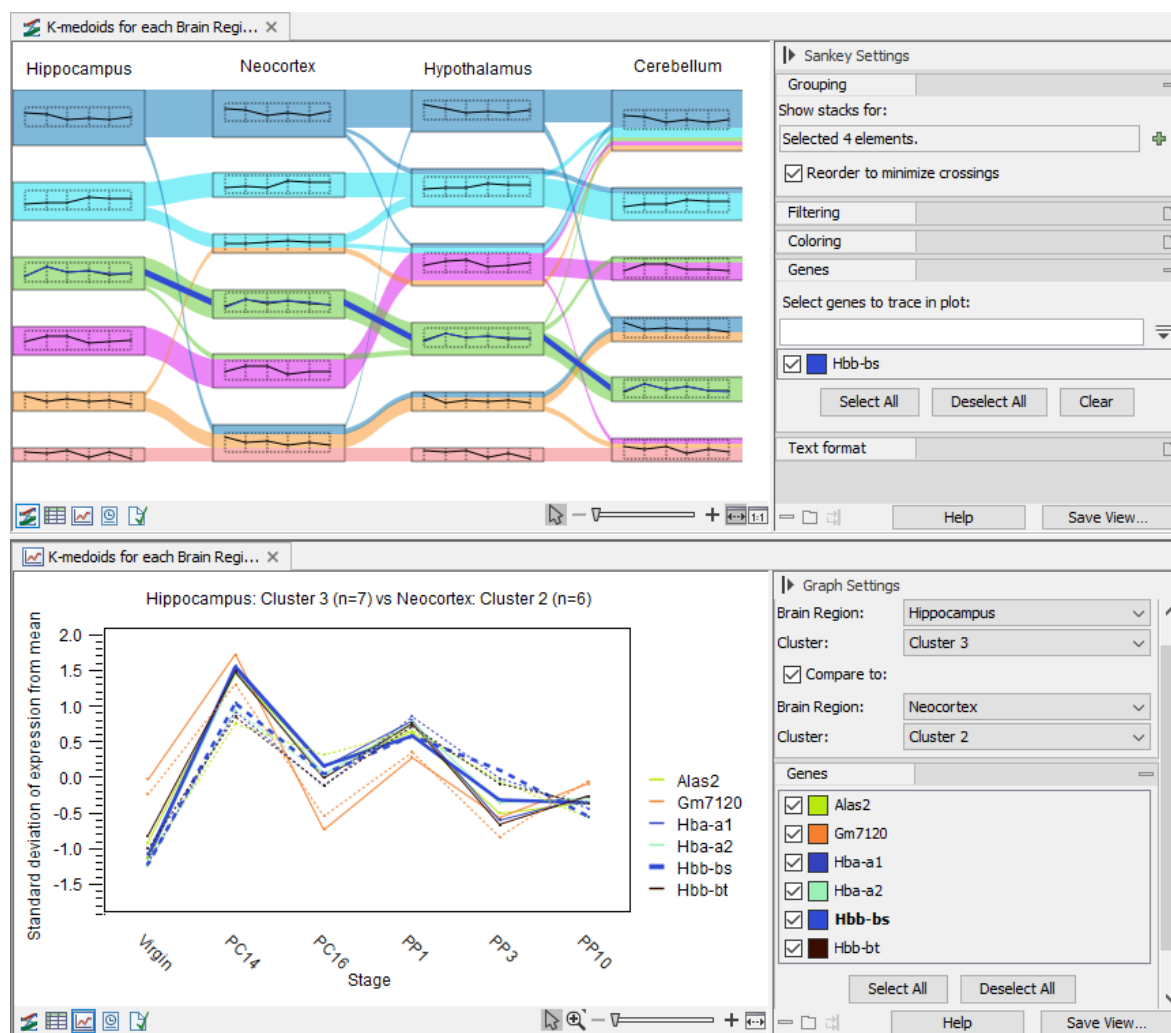


Figure 33.81: Top: The selected feature is highlighted in the plot and its expression is added to the thumbnail. Two clusters are selected. Bottom: The line graph of the clusters to be compared. The selected feature is represented in bold.

Features are z-score normalized prior to clustering: they are rescaled such that the mean expression value over all input samples for the clustering is 0, and the standard deviation is 1.

33.6 Differential Expression

In this section tools are described for calculating, visualizing and annotating differential expression. The tools are compatible both with the Expression Tracks created by the RNA-Seq Analysis tool and the tables created by the miRNA quantification tool. Two tools are available in the Workbench for calculating differential expressions. The **Differential Expression in Two Groups** tool performs a statistical differential expression test for a set of Expression Tracks and a set of control tracks. The **Differential Expression for RNA-Seq** tool performs a statistical differential expression test for a set of Expression Tracks with associated metadata. Both tools use multi-factorial statistics based on a negative binomial Generalized Linear Model (GLM).

How many replicates do I need? The Differential Expression for RNA-Seq tool is capable of running without replicates, but this is not recommended and the results should be treated with caution. In general it is desirable to have as many biological replicates as possible – typically at least 3. Replication is important in that it allows the 'within group' variation to be accurately estimated for a gene. In the absence of replication, the Differential Expression for RNA-Seq tool assumes that genes with similar average expression levels have similar variability.

Technical or biological replicates? Auer and Doerge, 2010 illustrate the importance of *biological replicates* with the example of an alien visiting Earth. The alien wishes to know if men are taller than women. It abducts one man and one woman, and measures their heights several times i.e. performs several *technical replicates*. However, in the absence of *biological replicates*, the alien would erroneously conclude that women are taller than men if this was the case in the two abducted individuals.

The use of the GLM formalism allows us to fit curves to expression values without assuming that the error on the values is normally distributed. Similarly to edgeR and DESeq2, we assume that the read counts follow a Negative Binomial distribution as explained in McCarthy et al., 2012. The Negative Binomial distribution can be understood as a 'Gamma-Poisson' mixture distribution i.e., the distribution resulting from a mixture of Poisson distributions, where the Poisson parameter λ is itself Gamma-distributed. In an RNA-Seq context, this Gamma distribution is controlled by the **dispersion** parameter, such that the Negative Binomial distribution reduces to a Poisson distribution when the dispersion is zero.

To learn more about the performance of the Differential Expression Analysis tool in comparison to well-accepted protocols like DESeq2 and EdgeR, read our benchmark results here: <https://digitalinsights.qiagen.com/news/blog/discovery/lasting-expressions/>.

33.6.1 Pre-filtering data for Differential Expression

It is not generally recommended to pre-filter data for differential expression analysis. Instead, by default, a post-processing step filters low expression features to increase power. This is described in more detail in section 33.6.4 and is also a feature of the commonly-used DESeq2 package <https://bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html#pre-filtering>.

EdgeR - another commonly-used algorithm for differential expression analysis - recommends filtering features with low expression before performing differential expression analysis. This is because 1) features with very low expression are unlikely to be biologically important, 2) the discreteness of counts can interfere with EdgeR's approximations <https://www.bioconductor.org/packages/release/bioc/vignettes/edgeR/inst/doc/edgeRUsersGuide.pdf>. To remove features unlikely to be of biological importance, we instead recommend post-filtering results on their "max group means" column.

In rare cases, pre-filtering can be desirable to speed up calculations and reduce memory consumption. It is possible to pre-filter Gene Expression data as follows (similar steps can be used for other types of expression data):

- Run **Create Expression Browser** on all the samples, see section 33.2.

- Export the Expression Browser to a format you can work with, for example Excel 2010 (.xlsx). It is easiest to deselect "Export all columns" and then in the next wizard step choose to "Export table as currently shown". For more details see section 8.1.6.
- Perform the filtering outside the workbench. For example, in Excel one might calculate the sum of the CPM values of all the samples for each feature, then filter the rows to show only those with a total CPM of at least 10. Copy the names of the retained features.
- Switch back to the workbench - open all the samples. For the first sample, filter such that "Name" "is in list" and then paste in the filtered names and click Filter. This may take a couple of minutes.
- Select all the rows, and choose to "Select Genes in Other Views". This might take a little time, but typically less than 1 minute. Now these rows are selected in all the open samples.
- Go through each sample and choose to "Create Track from Selection" - then save the new element.

Pre-filtering may also be desirable to remove extreme outliers. However, in most cases, the "Downweight outliers" option described in section 33.6.4 is preferable, because a gene can be differentially expressed and also have an outlier measurement.

33.6.2 The GLM model

Fitting a GLM to expression data

It is easiest to understand how the GLM model works through an example. Imagine an experiment looking at the effect of two drug treatments while controlling for gender:

- **Test differential expression due to** Treatment with three groups: drugA, drugB, placebo
- **While controlling for** Gender with groups: Male, Female

In an abuse of mathematical notation, the underlying GLM for each gene looks like

$$\log y_i = (\text{placebo and Male}) + \text{drugA} + \text{drugB} + \text{Female} + \text{constant}_i$$

where y_i is the expression level for the gene in sample i ; the combined term (placebo and Male) describes an arbitrarily chosen baseline expression level (of males being given a placebo); and the other terms drugA, drugB and Female are numbers describing the effect of each group with respect to this baseline. The constant_i accounts for differences in the library size between samples. For example, if a subject is male and given a placebo we predict the expression level to be

$$\log y_i = (\text{placebo and Male}) + \text{constant}_i.$$

If instead he had been given drug B, we would predict the expression level y_i to be augmented with the drugB coefficient, resulting in

$$\log y_i = (\text{placebo and Male}) + \text{drugB} + \text{constant}_i.$$

We assume that the expression levels y_i follow a Negative Binomial distribution. This distribution has a free parameter, the dispersion. The greater the dispersion, the greater the variation in expression levels for a gene.

The most likely values of the dispersion and coefficients, drugA, drugB and Female, are determined simultaneously by fitting the GLM to the data. To see why this simultaneous fitting is necessary, imagine an experiment where we observe counts {15,10,4} for Males and {30,20,8} for Females. The most natural fit is for the coefficient Female to have a two-fold change and for the dispersion to be small, but an alternative fit has no fold change and a larger dispersion. Under this second fit the variation in the counts is greater, and it is just by chance that all three Female values are larger than all three Male values.

Refining the estimate of dispersion

Much research has gone into refining the dispersion estimates of GLM fits. One important observation is that the GLM dispersion for a gene is often too low, because it is a *sample* dispersion rather than a *population* dispersion. We correct for this using the Cox-Reid adjusted likelihood, as in the multi-factorial edgeR method [Robinson et al., 2010]. To understand the purpose of the correction, it may help to consider the analogous situation of calculation of the variance of normally distributed measurements. One approach would be to calculate $\frac{1}{n} \sum (x_i - \bar{x})^2$, but this is the *sample* variance and often too low. A commonly used correction for the *population* variance is: $\frac{1}{n-1} \sum (x_i - \bar{x})^2$.

A second observation that can be used to improve the dispersion estimate, is that genes with the same average expression often have similar dispersions. To make use of this observation, we follow Robinson et al., 2010 in estimating gene-wise dispersions from a linear combination of the likelihood for the gene of interest and neighboring genes with similar average expression levels. The weighting in this combination depends on the number of samples in an experiment, such that the neighbors have most weight when there are no replicates, and little effect when the number of replicates is high.

When estimating dispersion, we use the following strategy:

1. We sort the genes from lowest to highest average logCPM (CPM is defined also by the edgeR authors). For each gene, we calculate its log-likelihood at a grid of known dispersions. The known dispersions are $0.2 * 2^i$, where $i = -6, -4.8, -3.6 \dots 6$ such that there are 11 values of i in total. You can imagine the results of this as being stored in a table with one column for each gene and one row for each dispersion, with neighboring columns having similar average logCPM (because of the sorting in the previous step).
2. We now calculate a weighted log-likelihood for each gene at these same known dispersions. This is the original log-likelihood for that gene at that dispersion plus a "weight factor" multiplied by the average log-likelihood for genes in a window of similar average logCPM. The window includes the 1.5% of genes to the left and to the right of the gene we are looking at. For example, if we had 3000 genes, and were calculating values for gene 500, then $0.015 * 3000 = 45$, so we would average the values for genes 455 to 545. The "weight factor" = $20 / (\text{numberOfSamples} - \text{number of parameters in the factor being tested})$. This

means that the greater the number of samples, the lower the weight, and the fewer free parameters to fit, the lower the weight.

3. We fit an FMM spline for each gene to its 11 weighted log-likelihoods. Our implementation of the FMM spline is translated from the Fortran code found here <https://www.netlib.org/fmm/spline.f>. This spline allows us to estimate the weighted log-likelihood at any dispersion within the grid i.e., from $0.2 * 2^{-6}$ to $0.2 * 2^6$
4. We find the dispersion on the spline that maximizes the log-likelihood. This is the dispersion we use for that gene.

Downweighting outliers

It is often difficult to detect outliers. Therefore, instead of removing putative outliers completely from the GLM fit, we give them "observation weights" that are smaller than 1. These weights reduce the effect of the outliers on the GLM fit.

Outliers can have two undesirable effects. First, genes with outliers are more likely to be called as differentially expressed because a sample with anomalously high expression is hard to reconcile with no fold change. Second, outliers can prevent differentially expressed genes from being detected. This is because, when a gene has an anomalously high expression, the dispersion for that gene will typically be overestimated. Consequently the dispersion of genes with similar expression will be overestimated, such that they are less likely to be called as differentially expressed.

When downweighting outliers, the above dispersion estimation procedure is repeated 5 more times, and per-gene weights are estimated for the samples at each iteration. Note that these weights are not related to the weights in the weighted log-likelihood of the dispersion estimation procedure.

The following iterative procedure is performed to estimate the observation weights. For each gene, one iteration proceeds as follows:

1. The Cook's distance is calculated for each sample. The Cook's distance for sample i is

$$D_i = \frac{1}{p} z_i^2 \frac{h_i}{(1 - h_i)^2}$$

where

- p is the number of parameters in the fit.
- z_i is the Pearson residual for sample i , a measure of how well the fitted model predicts the expression of sample i . The Pearson residual is calculated as the difference between the measured expression y_i and the modeled expression \hat{y}_i , divided by the standard deviation σ of the negative binomial distribution with dispersion γ that has been fitted to the gene:

$$z_i = \frac{y_i - \hat{y}_i}{\sigma} = \frac{y_i - \hat{y}_i}{\sqrt{\hat{y}_i(1 + \gamma\hat{y}_i)}}$$

- h_i is the leverage of sample i on the fit, a measure of how much sample i affects the fitted parameters. The leverage is the i^{th} entry on the diagonal of the projection matrix H which relates measured expressions to modeled expressions: $\hat{y}_i = Hy_i$.

Outliers are expected to have large Cook's distances because they tend to be poorly fitted, resulting in large z_i , and they are likely to disproportionately influence the fit compared to other points, leading to large h_i .

2. Cook's distances are assumed to follow an F distribution $F(p, n - p)$, where n is the number of samples. For each sample, this distribution is used to calculate the probability $p(D_i)$ of obtaining a Cook's distance as extreme as or more extreme than the observed D_i . A first approximation for the sample observation weight is w_i

$$w_i = \begin{cases} p(D_i) & \text{if } p(D_i) < 0.5 \\ 1 & \text{otherwise} \end{cases}$$

The probability cut-off of 0.5 is set empirically from experiments on real data.

3. This approximation is not effective in practice when outliers are present because a single outlier in a condition can influence the fit for all other samples within that condition, causing them to also exhibit large Cook's distances and appear as outliers. As a result, this approximation leads to reduced observation weights of many non-outlier samples, especially during the initial iteration of the procedure. We address this in two parts:
 - (a) We limit how much a group of samples is downweighted. For the example at the start of this section, the groups are "drug A female", "drug A male", "drug B female" etc. The allowed downweighting for a group g containing n_g samples is:

$$d_g = \min \left(\sum_{i \in g} 1 - w_i, \min(n_g - 1, \max(1, 0.1n_g)) \right)$$

We hence disregard at most one sample or up to 10% of the samples from the group, whichever is larger. However, at least one sample from the group must be kept. In practice, the weights are spread across all samples: while some samples may be more heavily downweighted than others, it is uncommon for any sample to be entirely ignored, i.e., receive an observation weight w_i of 0.

- (b) We calculate a normalized skewed weight

$$s_i = \begin{cases} \frac{1}{s_t} \frac{w_{min}}{w_i} & \text{if } p(D_i) < 0.5 \\ 0 & \text{otherwise} \end{cases}$$

to penalize the most extreme outliers, where

- w_{min} is the smallest w_i across all samples, regardless of their group.
- s_t is the normalization factor such that $\sum_k s_k = 1$.

If the true outlier has a w_i that is 10 times smaller than another sample in the same group, then s_i for the true outlier will also be 10 times smaller than the one for the other sample.

4. The final observation weights are:

$$o_i = 1 - \min(1, \gamma d_g s_i)$$

where γ is a normalization factor used to ensure that at most one sample, or 10% of the total number of samples n are disregarded from the entire dataset:

$$\gamma = \min \left(1, \frac{\max(1, 0.1n)}{\sum_i \min(1, d_g s_i)} \right)$$

5. The GLM is re-fit, using these weights.

The above steps are repeated until no observation weight changes by more than 0.01, or a weight that was < 0.01 becomes 1 in the following round, or a weight that decreased by more than 0.01 in the previous round increases by more than 0.01 in the next round. These stopping criteria are needed because the weights do not converge: when a weight becomes sufficiently small, the leverage component of the Cook's distance tends to zero and so the Cook's distance also becomes zero and the weight in the next iteration will be 1.

Once the observation weights have been determined, the re-fitted GLM has a different likelihood than before. This is partly because the weighting is applied directly to the likelihood, and partly because the weighting affects the fitted coefficients.

The use of observation weights in the GLM is based on [Zhou et al., 2014](#). The use of Cook's distance is motivated by DESeq2, which ignores genes with extreme Cook's distances when the number of replicates is small, rather than downweighting them. The form of the weights is novel to *CLC Genomics Workbench*.

Statistical testing

The final GLM fit and dispersion estimate allows us to calculate the total likelihood of the model given the data, and the uncertainty on each fitted coefficient. The two statistical tests each make use of one of these values.

Wald test Tests whether a given coefficient is non-zero. This test is used in the **All group pairs** and **Against control group** comparisons. For example, to test whether there is a difference between subjects treated with a placebo, and those treated with drugB, we would use the Wald test to determine if the drugB coefficient is non-zero.

Likelihood Ratio test Fits two GLMs, one with the given coefficients and one without. The more important the coefficients are, the greater the ratio of the likelihoods of the two models. The likelihood of both models is computed using the dispersion estimate and observation weights of the larger model. This test is used in the Across groups (ANOVA-like) comparison. If we wanted to test whether either drug had an effect, we would compare the likelihoods of the GLM described with equation

$$\log y_i = (\text{placebo and Male}) + \text{drugA} + \text{drugB} + \text{Female} + \text{constant}_i$$

with those in the reduced GLM with equation

$$\log y_i = (\text{Male}) + \text{Female} + \text{constant}_i.$$

33.6.3 Differential Expression in Two Groups

The **Differential Expression in Two Groups** tool performs a statistical differential expression test for a set of Expression Tracks and a control. It uses multi-factorial statistics based on a negative

binomial GLM as described in section 33.6.2. Differential Expression in Two Groups only handles one factor and two groups, as opposed to the Differential Expression for RNA-Seq tool that can handle multiple factors and multiple groups.

To run **Differential Expression in Two Groups**, go to:

Tools | RNA-Seq and Small RNA Analysis (🇺🇸) | Differential Expression (🇺🇸) | Differential Expression in Two Groups (🇺🇸)

In the first dialog (figure 33.82), select a number of Expression tracks (🇺🇸) (GE or TE) and click **Next**. For Transcripts Expression Tracks (TE), the values used as input are "Total transcript reads". For Gene Expression Tracks (GE), the values used depend on whether an eukaryotic or prokaryotic organism is analyzed, i.e., if the option "Genome annotated with Genes and transcripts" or "Genome annotated with Genes only" was used. For Eukaryotes the values are "Total Exon Reads", whereas for Prokaryotes the values are "Total Gene Reads".

Note that the tool can be run in batch mode, albeit with the same control group expression for all selected batch units.

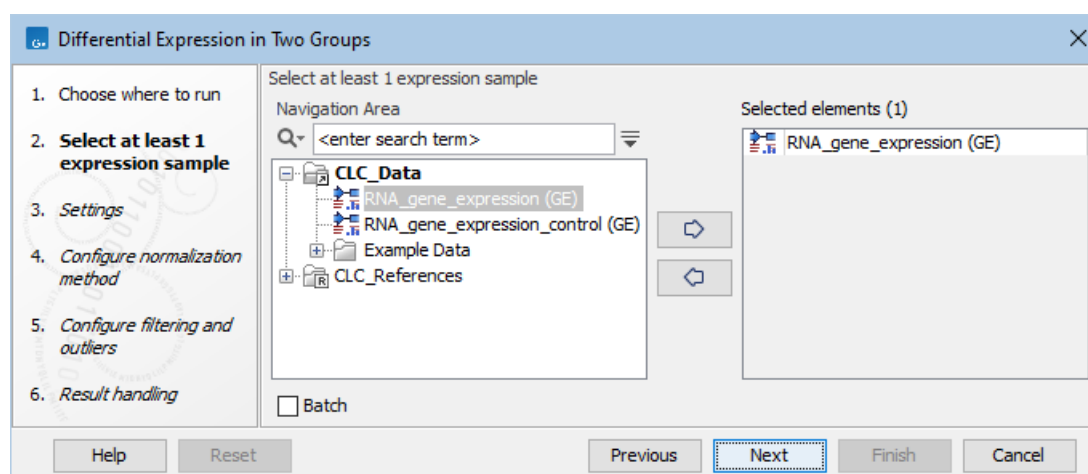


Figure 33.82: Select expression tracks for analysis.

In the **Settings** dialog, select a number of control Expression tracks (🇺🇸) (GE or TE). A warning message (as seen in figure 33.83) appears if only one track is selected for either the input or the control group: such a setting does not provide replicates, thus does not ensure sufficient statistical power to the analysis.

The available normalization options can be seen in figure 33.84.

First, choose the application that was used to generate the expression tracks: Whole transcriptome RNA-Seq, Targeted RNA-Seq, or Small RNA. For Targeted RNA-Seq and Small RNA, you can choose between two normalization methods: TMM and Housekeeping genes, while Whole transcriptome RNA-Seq will be normalized by default using the TMM method. For more detail on the methods see section 33.1.

TMM Normalization (Trimmed Mean of M values) calculates effective libraries sizes, which are then used as part of the per-sample normalization. TMM normalization adjusts library sizes based on the assumption that most genes are not differentially expressed.

Normalization with Housekeeping genes can be done when a set of housekeeping genes to use

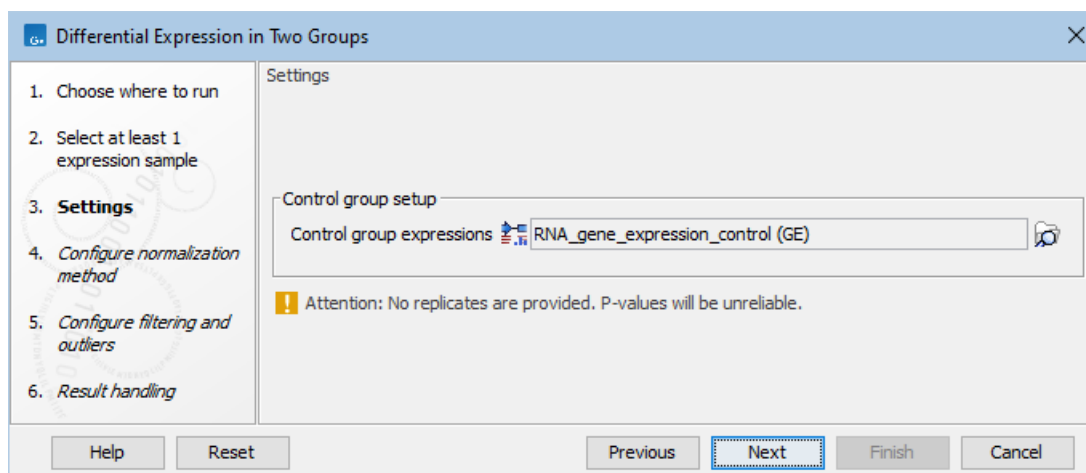


Figure 33.83: Select enough control expression tracks to ensure that replicates are provided.

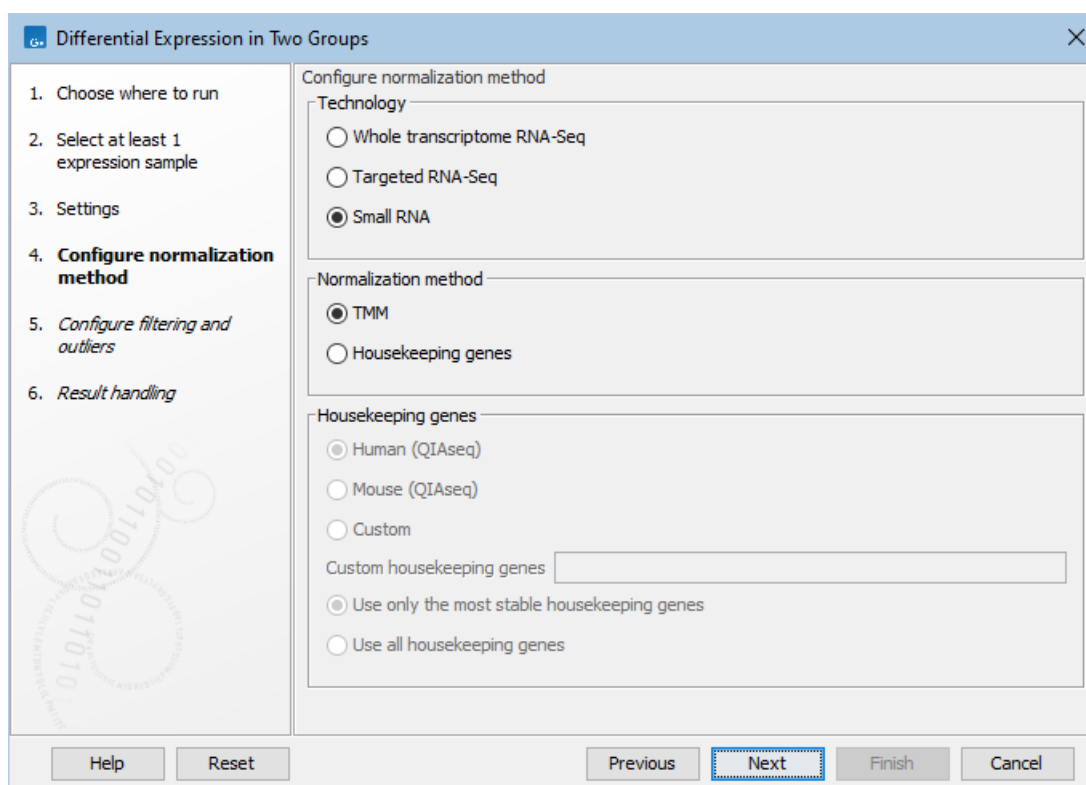


Figure 33.84: Normalization methods.

is available: in the "Custom housekeeping genes" field, type the name of the genes separated by a space. Finally choose between these two options:

- **Use only the most stable housekeeping genes** will use a subset (at least three) of the most stable genes for normalization, these being defined using the GeNorm algorithm [Vandesompele et al., 2002].
- **Use all housekeeping genes** keep all housekeeping genes listed for normalization.

When working with Targeted RNA Panels, we recommend that normalization is done using the Housekeeping genes method rather than TMM. Predefined list of housekeeping genes are available for samples generated using Human and Mouse QIAseq panels (hover with the mouse on the dialog to find the list of genes included in the set). If you are working with a custom panel, you can also provide the corresponding set of housekeeping genes in the "Custom housekeeping genes" as described above.

In the final dialog, choose whether to downweight outlier expressions, and whether to filter on average expression prior to FDR correction.


Downweighting outliers is appropriate when a standard differential expression analysis is enriched for genes that are highly expressed in just one sample. These genes do not fit the null hypothesis of no change in expression across samples. Downweighting comes at a cost to precision and so is not recommended generally. For more details, see section 33.6.4.

Filtering maximizes the number of results that are significant at a target FDR threshold, but at the cost of potentially removing significant results with low average expression. For more details, see section 33.6.4.

The tool outputs a statistical comparison table, see section 33.6.5.

33.6.4 Differential Expression for RNA-Seq

Differential Expression for RNA-Seq performs a statistical differential expression test for a set of Expression Tracks. It uses multi-factorial statistics based on a negative binomial GLM. The tool supports paired designs and can control for batch effects. The statistical analysis is described in more detail in section 33.6.2.

To run the **Differential Expression for RNA-Seq** tool, you need Expression Tracks  and a CLC Metadata Table that provides, at minimum, information about the conditions relevant for the statistical testing. The Expression Tracks provided as input must already have associations to this CLC Metadata Table (see chapter 13).

The **RNA-Seq and Differential Gene Expression Analysis** template workflow includes **Differential Expression for RNA-Seq** and illustrates an approach where metadata can be provided in an Excel, CSV or TSV format file, avoiding the need to create a CLC Metadata Table before starting the analysis. See section 14.5.5 for details.

Running the Differential Expression for RNA-Seq tool To run **Differential Expression for RNA-Seq**, go to:

Tools | RNA-Seq and Small RNA Analysis  | **Differential Expression**  | **Differential Expression for RNA-Seq** 

Select a number of Expression tracks  and click **Next** figure 33.85.

For Expression Tracks (TE), the values used as input are "Total transcript reads". For Gene Expression Tracks (GE), the values used depend on whether a eukaryotic or prokaryotic organism is analyzed, i.e., if the option "Genome annotated with Genes and transcripts" or "Genome annotated with Genes only" is used. For Eukaryotes the values are "Total Exon Reads", whereas for Prokaryotes the values are "Total Gene Reads".

The order of comparisons can be controlled by changing the order of the inputs.

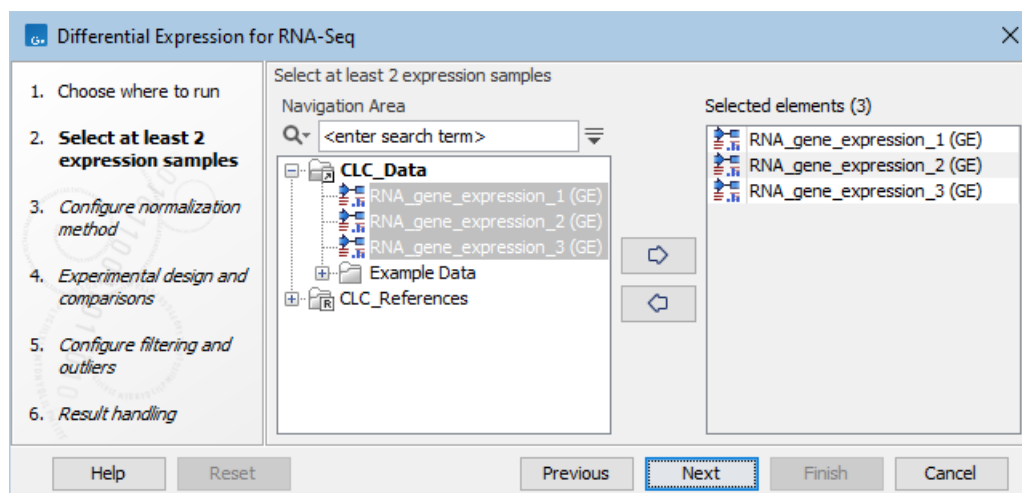


Figure 33.85: Select a number of Expression Tracks.

Normalization options are provided in the "Configure normalization method" step of the wizard (figure 33.86).

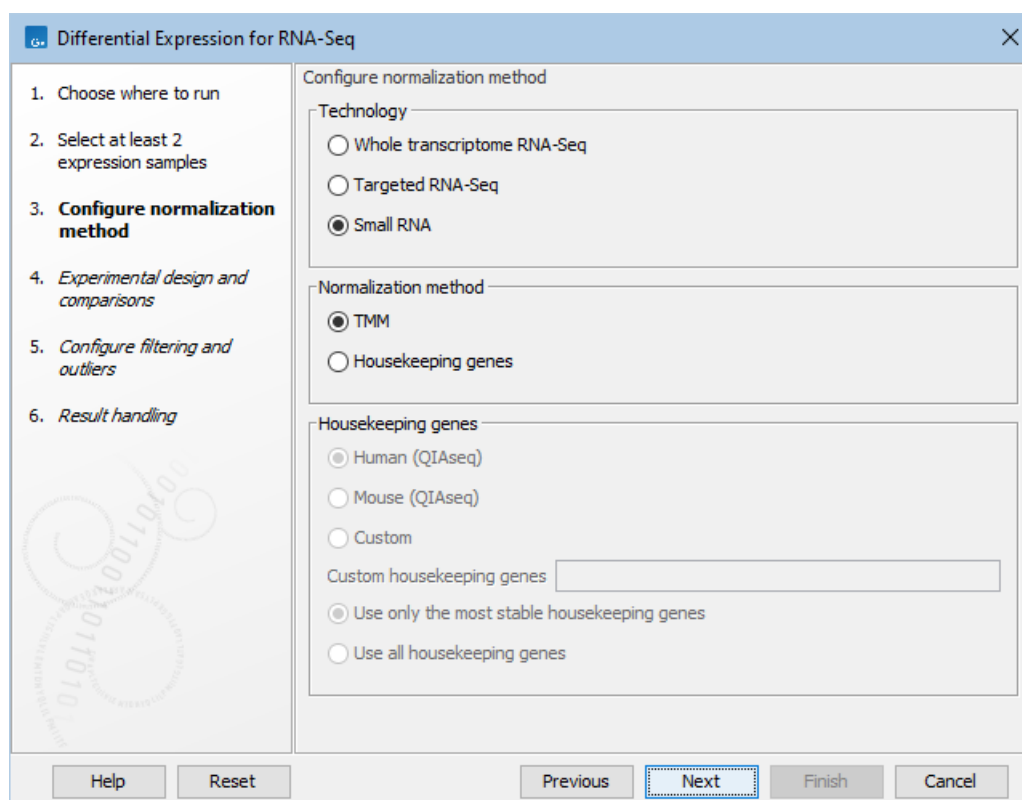


Figure 33.86: Normalization methods.

First, choose the application that was used to generate the expression tracks: Whole transcriptome RNA-Seq, Targeted RNA-Seq, or Small RNA. For Targeted RNA-Seq and Small RNA, you can choose between two normalization methods: TMM and Housekeeping genes, while Whole transcriptome RNA-Seq will be normalized by default using the TMM method. For more detail on the methods see section 33.1.

TMM Normalization (Trimmed Mean of M values) calculates effective libraries sizes, which are then used as part of the per-sample normalization. TMM normalization adjusts library sizes based on the assumption that most genes are not differentially expressed.

Normalization with Housekeeping genes can be done when a set of housekeeping genes to use is available: in the "Custom housekeeping genes" field, type the name of the genes separated by a space. Finally choose between these two options:

- **Use only the most stable housekeeping genes** will use a subset (at least three) of the most stable genes for normalization, these being defined using the GeNorm algorithm [Vandesompele et al., 2002].
- **Use all housekeeping genes** keep all housekeeping genes listed for normalization.

When working with Targeted RNA Panels, we recommend that normalization is done using the Housekeeping genes method rather than TMM. Predefined list of housekeeping genes are available for samples generated using Human and Mouse QIAseq panels (hover with the mouse on the dialog to find the list of genes included in the set). If you are working with a custom panel, you can also provide the corresponding set of housekeeping genes in the "Custom housekeeping genes" as described above.

In the "Experimental design and comparison" wizard step, you are asked to provide information about the samples, test conditions, and the type of testing to carry out (figure 33.87).

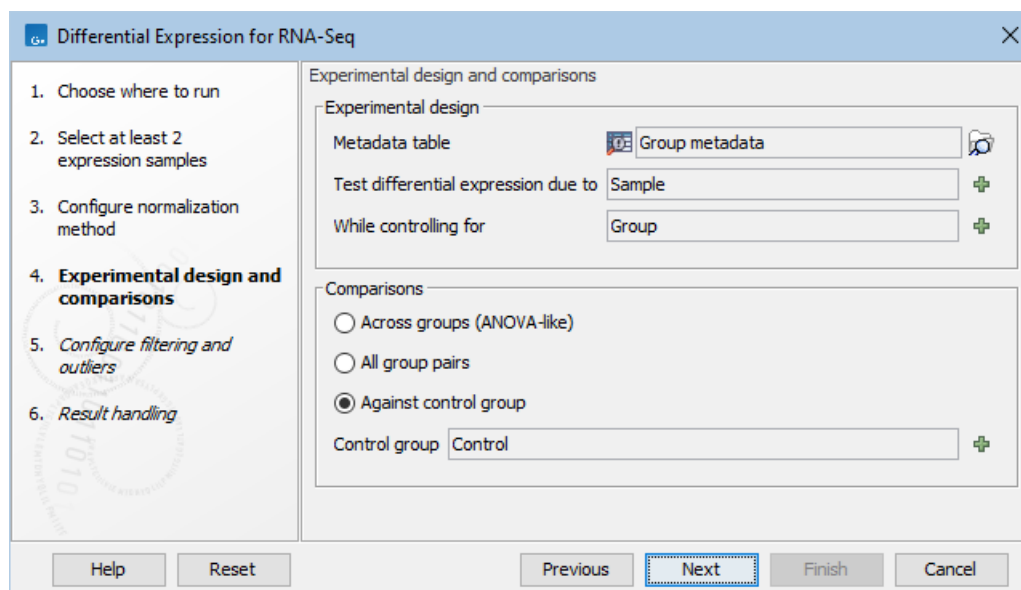


Figure 33.87: Setting up the experimental design and comparisons.

In the Experimental design panel, the following information must be provided:

- **Metadata table** Specify a CLC Metadata Table containing information about the selected Expression Tracks relevant for the statistical testing, i.e. the factors. The Expression Tracks must already have associations to the selected CLC Metadata Table.
- **Test differential expression due to** Specify the factor to test for differential expression.

- **While controlling for** Specify confounding factors, i.e., factors that are not of primary interest, but may affect gene expression.

In the **Comparisons** panel, the type of test(s) to be run is specified. This affects the number and type of statistical comparison outputs generated (see section 33.6.5 for more details).

Depending on the type of comparison chosen, a Wald test or a Likelihood Ratio test will be used. For example, assume that we test a factor called 'Tissue' with three groups: skin, liver, brain.

- **Across groups (ANOVA-like)** This mode tests for the effect of a factor across all groups.
 - *Outputs produced:* "Due to Tissue"
 - *Test used:* Likelihood ratio test
 - *Fold change reports:* The maximum pairwise fold change between any two of the three tissue types.
 - *Max of group means reports:* The maximum of the average group TPM values among any of the tissue types for a gene.
- **All group pairs** tests for differences between all pairs of groups in a factor.
 - *Outputs produced:* "skin vs. liver", "skin vs. brain", "liver vs. brain"
 - *Test used:* Wald test
 - *Fold change reports:* The fold change in the defined order between the named pair of tissue types.
 - *Max of group means reports:* The maximum of the average group TPM values between the two named tissue types.
- **Against control group** This mode tests for differences between all the groups in a factor and the named reference group. In this example the reference group is skin.
 - *Outputs produced:* "liver vs. skin", "brain vs. skin"
 - *Test used:* Wald test
 - *Fold change reports:* The fold change in the defined order between the named pair of tissue types.
 - *Max of group means reports:* The maximum of the average group TPM values between the two named tissue types.

Note: Fold changes are calculated from the GLM, which corrects for differences in library size between the samples and the effects of confounding factors. It is therefore not possible to derive these fold changes from the original counts by simple algebraic calculations.

In the "Configure filtering and outliers" wizard step, you choose whether to downweight outlier expressions, and whether to filter on average expression prior to FDR correction.

Downweighting outliers is appropriate when a standard differential expression analysis is enriched for genes that are highly expressed in just one sample. These genes do not fit the null hypothesis of no change in expression across samples. Downweighting comes at a cost to precision and so is not recommended generally. For more details, see section 33.6.4.

Filtering maximizes the number of results that are significant at a target FDR threshold, but at the cost of potentially removing significant results with low average expression. For more details, see section 33.6.4.

The outputs from Differential Expression for RNA-Seq are described in section 33.6.5.

Downweighting outliers

Downweighting outliers reduces the chance that a single outlier for a feature is sufficient to reject the null hypothesis and to call a feature differentially expressed. In the presence of outliers, downweighting can increase both sensitivity and precision. When no outliers are present, downweighting typically leads to similar sensitivity and reduced precision. Tests on simulated data show that downweighting leads to a modest loss of control of the false discovery rate (FDR) regardless of whether outliers are present. For example, among features with FDR p-value below 0.05, 5% should be false positives, but when downweighting outliers a higher proportion will be false positives.

Because downweighting is only advantageous when outliers are actually present, we recommend using it only when a standard analysis is enriched for genes that are highly expressed in just one sample. This is often easiest to see in a heat map.

Note that downweighting outliers is not a way of handling low quality samples. If a single sample behaves very differently from others, consider removing it from the analysis.

The implementation of outlier downweighting is described in more detail in section 33.6.2.

Filtering on average expression

The FDR p-value is a multiple-testing correction for all tests that are performed. Sometimes power can be improved if genes are filtered prior to FDR correction. The filtering approach used is similar that of DESeq2 (Love et al., 2014, see section "Automatic Independent Filtering").

An example of the results of this procedure is shown in figure 33.88. The left side of the figure shows results with the option disabled, and the right side shows the same results with the option enabled. *Loxhd1* is filtered away prior to the FDR correction, and so has "FDR p-value = NaN". All other genes have lower FDR p-values because fewer tests were performed as a result of the filtering. The total number of genes detected as significantly differentially expressed at a target FDR of 0.1 has been increased.

Note that only the values in the **FDR p-value** column are changed. When filtering is enabled, low expression genes are filtered away prior to FDR correction. The exact threshold for low expression is determined by the tool and may be 0, in which case filtering has no effect. The threshold is chosen so as to maximize the number of significant tests at a target FDR of 0.1.

In detail, the determination of the filtering threshold works as follows:

1. Genes are ordered by average counts, where the average includes all samples across all conditions.
2. FDR corrections are run on the most expressed 1%, 2%... 100% of the genes, and the number of significant differentially expressed (DE) genes at a target FDR of 0.1 in each case is plotted.

Name	Max group mean	Log2 fold change	Fold change	P-value	FDR p-value	Bonferroni
Aqp4	58.59	-0.40	-1.32	1.87E-19	4.14E-15	4.14E-15
Gfap	72.64	-0.67	-1.59	8.60E-15	9.51E-11	1.90E-10
Tle2	20.38	-0.30	-1.23	6.25E-11	4.61E-7	1.38E-6
Acot1	9.35	-0.53	-1.44	5.14E-10	2.45E-6	1.14E-5
Dcn	9.40	0.50	1.42	5.54E-10	2.45E-6	1.22E-5
Tll3	12.02	0.28	1.21	2.19E-8	8.06E-5	4.83E-4
Tmem145	60.25	-0.25	-1.19	5.84E-8	1.84E-4	1.29E-3
Gja1	79.10	-0.24	-1.18	1.04E-7	2.86E-4	2.29E-3
Ogn	1.16	0.99	1.98	1.75E-7	4.29E-4	3.86E-3
A2m	25.83	0.34	1.27	3.17E-7	7.00E-4	7.00E-3
Peg10	4.41	0.29	1.22	3.67E-7	7.38E-4	8.12E-3
Cams1	3.28	-0.42	-1.34	9.07E-7	1.67E-3	0.02
Mbd5	17.67	-0.23	-1.18	1.10E-6	1.88E-3	0.02
Srebf1	12.68	0.23	1.18	1.65E-6	2.61E-3	0.04
Prf	3.48	4.09	17.00	2.02E-6	2.98E-3	0.04
Hsp90	98.72	-0.31	-1.24	2.33E-6	3.21E-3	0.05
Hsp91	32.62	0.22	1.16	2.90E-6	3.77E-3	0.06
Loxhd1	0.10	-0.22	-2.53	4.14E-3	4.62E-3	0.09
Maf	45.42	-0.21	-1.16	4.29E-6	4.99E-3	0.09
4932438H23Rk	0.49	1.03	2.04	4.59E-6	5.07E-3	0.10
Casq2	13.64	0.25	1.19	5.16E-6	5.43E-3	0.11
Spag5	4.78	0.26	1.20	6.39E-6	6.42E-3	0.14
Hsp90b1	87.12	0.29	1.22	1.05E-5	0.01	0.23
Mt2	413.35	-0.29	-1.22	1.22E-5	0.01	0.27
Them4	6.54	0.25	1.19	1.31E-5	0.01	0.29
Rgs7	20.78	0.22	1.16	1.49E-5	0.01	0.33
Fam81a	4.67	0.36	1.28	1.49E-5	0.01	0.33
Wls	6.73	0.23	1.17	1.57E-5	0.01	0.35

Figure 33.88: Results of the same test performed without (left) and with (right) filtering on average expression. Only the FDR p-values are changed. More genes are found significant at a target FDR of 0.1, but at a cost that genes with low average expression, such as *Loxhd1*, are filtered away.

3. A smoothed line is fit to these data using local regression.
4. An estimate is made of the variation in the number of DE genes around the line.
5. The final filtering threshold is that which keeps most genes while being at most 1 standard deviation below the maximum number of DE genes.

33.6.5 Output of the Differential Expression tools

Differential expression tools output one or more statistical comparison tracks or tables.

It can often be useful to collect all samples in an expression browser (section 33.2), and visualize it alongside the statistical comparison, see section 2.1.4.

Statistical comparisons have a number of views (figures 33.89 and 33.90), displaying information about the performed test in different formats:

- Table
- Track
- Volcano plot

Using tracks differential expression results can be stacked with data of different types based on a compatible genome, and linked viewed opened. See section 27.2. See also section 27.3 for details about working with individual tracks.

Note that the track view is not available for statistical comparison tables.

The statistical comparison table offers, for each feature, the following

- **Max group mean.** For each group in the statistical comparison, the average TPM is calculated. This value is the maximum of the average TPM's.

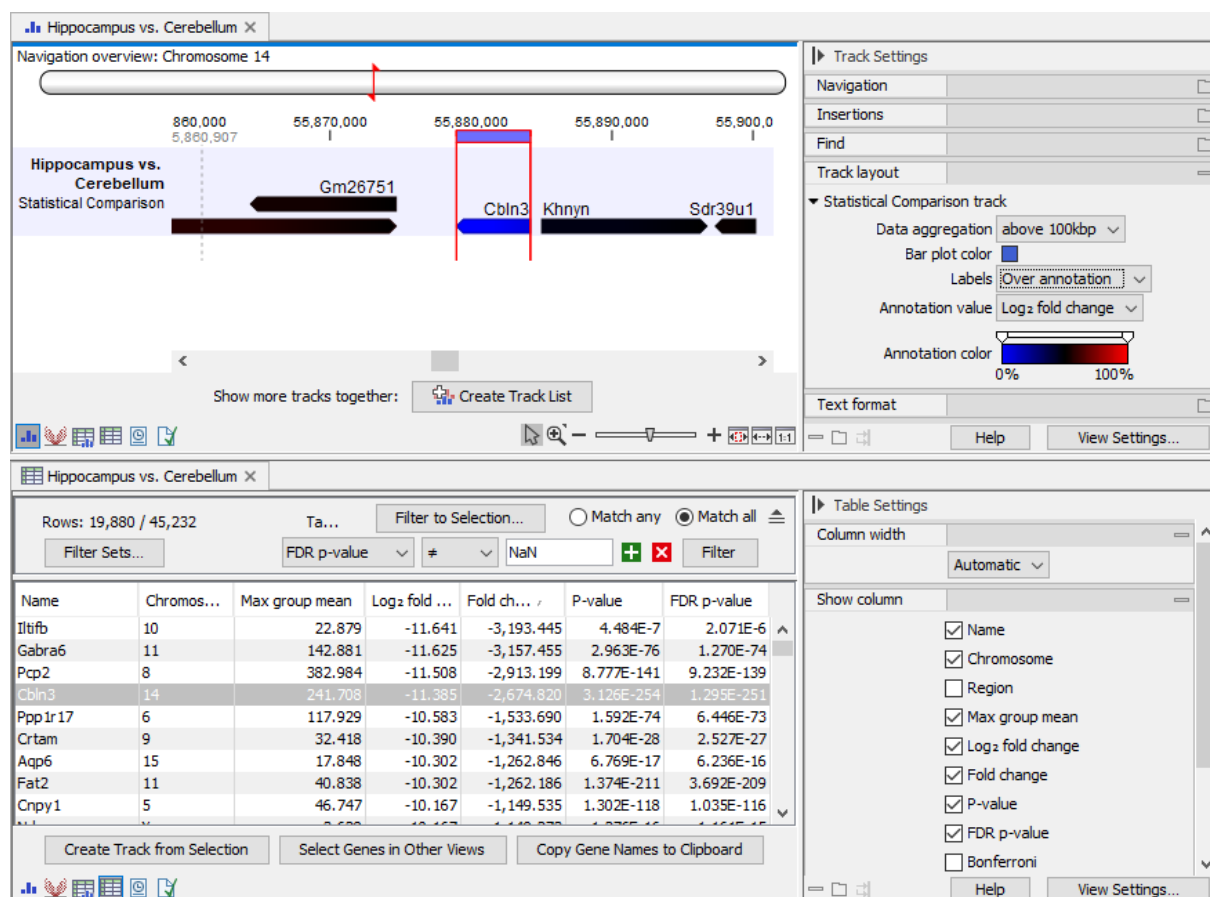


Figure 33.89: Views of a statistical comparison. Top: Track view. Bottom: Table view. The views are linked: selecting a feature in one view, also selects the feature in the other view.

- **Log₂ fold change.** The logarithmic fold change.
- **Fold change.** The (signed) fold change: the relative expression between the groups. Features that are not observed in any sample have undefined fold changes and are reported as NaN (not a number).
The fold change is estimated using the GLM model, see section 33.6.2. It is not possible to derive the fold change from the expression values by simple algebraic calculations.
- **P-value.** Standard p-value. Features that are not observed in any sample have undefined p-values and are reported as NaN (not a number).
- **FDR p-value.** The false discovery rate corrected p-value. This is calculated directly from the values in the P-value column.
- **Bonferroni.** The Bonferroni corrected p-value. This is calculated directly from the values in the P-value column.

Note that NaN p-values are not considered when calculating the FDR and Bonferroni corrected p-values.

At the bottom of the table view, the following buttons are available:

- **Create Track/Table from Selection.** Creates a track/table using the selected features.
- **Select Genes/Transcripts in Other Views.** Selects the currently selected features in all other opened views containing the same features, for example another view of the statistical comparison, or an expression browser (section 33.2).
- **Copy Gene/Transcript Names to Clipboard.** Copies the names of the selected features to the clipboard.

Volcano plots

The volcano plot (figure 33.90) shows the relationship between the fold changes and p-values. The \log_2 fold changes and $-\log_{10}$ p-values are plotted on the x- and y-axis, respectively. Features of interest are typically those with large fold changes (far from $x = 0$) that are statistically significant, i.e. have small p-values (far from $y = 0$). They are located in the upper left (down-regulated) and upper right (up-regulated) hand corners of the volcano plot.

Volcano plots can exhibit unexpected patterns looking like "wings", such as the one in orange in the bottom left corner in figure 33.90. These patterns reflect the mathematical relationship between fold changes and p-values, which can be exposed when there are few replicates and expression is low in one condition. For example, expression counts for two genes might be (5,5) vs (0,0) and (5,6) vs (0,1). These two genes would appear in the same "wing". Two other genes with expression counts (5,5) vs (0,1) and (5,6) vs (0,1) would be in another "wing".

The following can be configured in the Side Panel, see section 2.1.6:

Volcano plot. General options for configuring the content and coloring.

- **P-value type.** The standard p-value, FDR p-value or Bonferroni can be displayed in the plot.
- **Lower limit.** All p-values smaller than this number are rounded to this number. This allows very small values to be visualized on a logarithmic scale volcano plot. The limit can be input as linear or logarithmic.
- **Coloring.** The features in the volcano plot can be colored in different ways:
 - **Fixed color.** Down-regulated features have one color, and up-regulated features have another.
 - **Gradient.** Down-regulated features have one gradient, and up-regulated features have another gradient. Gradients reflect the fold change. The gradient for up-regulated features can be applied to the down-regulated features by checking **Mirror up-regulated**.
 - **By annotation.** Features are colored using the annotations selected in the **Annotations** group, see below.
See section 2.1.6 for how to change the colors and/or gradient.
- **Show faded points.** If checked, faded features are shown in the plot. Features are faded when
 - They have small fold change and/or non-significant p-values, as configured in the **Thresholds** group, see below.

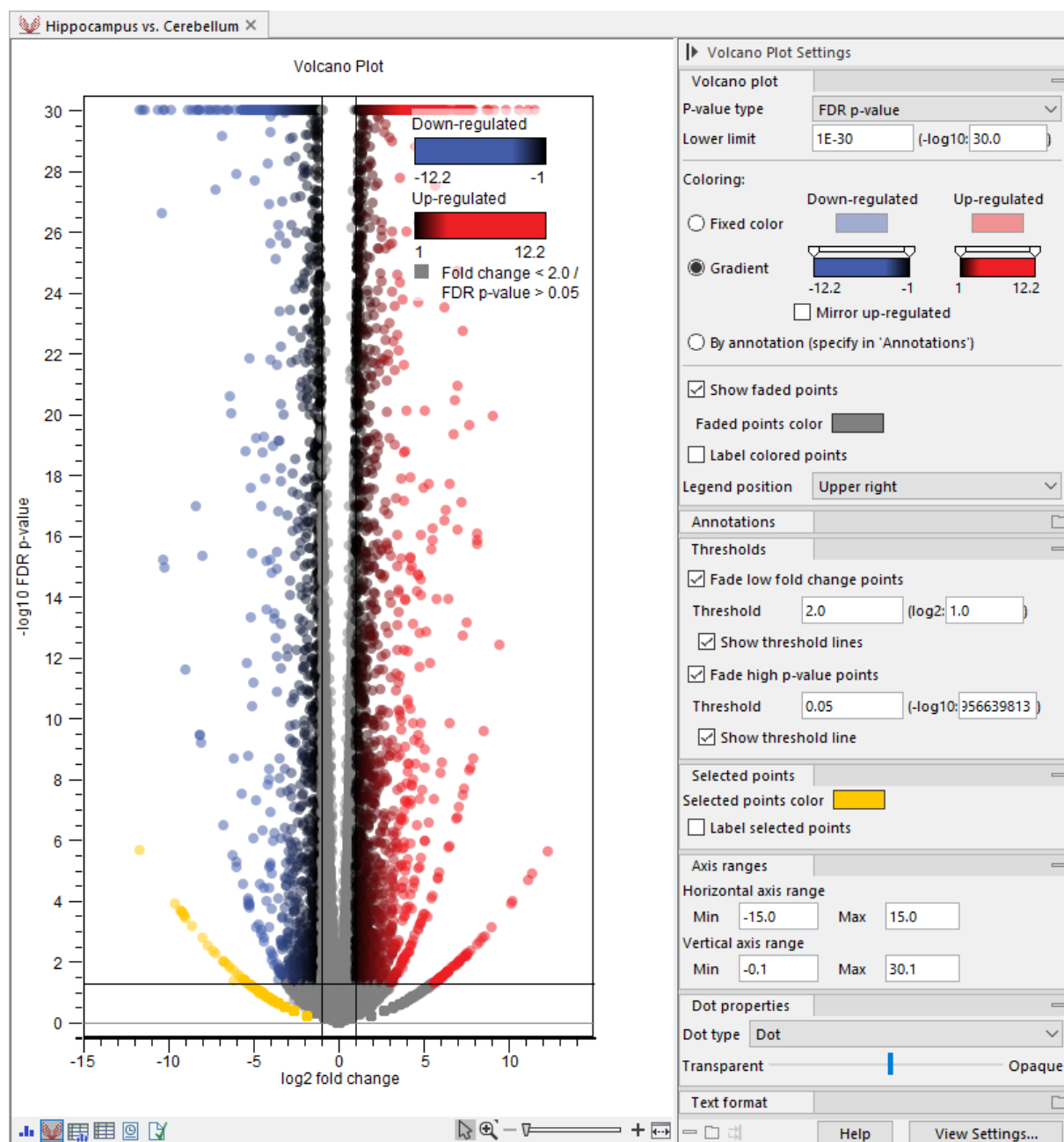


Figure 33.90: Customized volcano plot. Features are colored using gradients and those with low fold changes or high p-values are faded. The legend is shown in the upper right corner. The plot uses transparency for better visualization of overlapping points. Features in the bottom left corner are selected to highlight a "wing" pattern. The horizontal axis range is adjusted to center the plot.

- They have an annotation that has not been chosen for coloring when coloring **By annotations**.
- **Faded points color.** The color of the faded points.
- **Label colored points.** If checked, points that are not faded are labeled using the feature name.

- **Legend position.** Choose whether to display a color legend and where. Note that the legend does not contain the selected points (see below).

Annotations. Options for coloring features using annotations (figure 33.91).

- **Annotation type.** The type of annotation to be used for coloring.
- **Select annotations to color by.** The search filed is used for finding relevant annotations. Adding a space lists all available annotations. Double-clicking or hitting enter on a selected search result adds the annotation for coloring at the bottom.

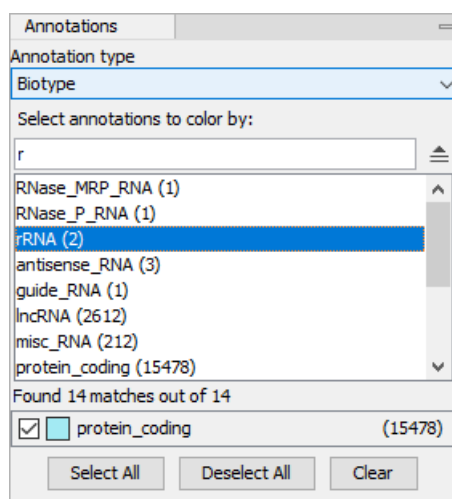


Figure 33.91: The Annotations group in the Side Panel. All biotype annotations containing "r" are shown, and protein coding features are colored using turquoise.

Thresholds. Options for fading features with small fold changes or non-significant p-values.

- **Fade low fold change points.** Fade features with absolute fold changes that are lower than the selected threshold.
- **Threshold.** The threshold can be input as linear or logarithmic.
- **Show threshold lines.** If checked, two vertical lines are drawn indicating the selected threshold and the corresponding negative value.
- **Fade high p-value points.** Fade features with p-values that are larger than the selected threshold.
- **Threshold.** The threshold can be input as linear or logarithmic.
- **Show threshold line.** If checked, a horizontal line is drawn indicating the selected threshold.

Selected points. Options for configuring selections.

Points in the volcano plot can be selected by

- Clicking on an individual point. Note that if there are multiple, overlapping points under the mouse cursor, just one of these points will be selected.
- Using the lass tool. With the left mouse button depressed, drag the cursor around the area of interest. Release the button to create the selection.

Hold down the Ctrl key (⌘ on Mac) to perform an additional selection.

- **Selected points color.** The color of the selected points.
- **Label selected points.** If checked, selected points are labeled using the feature name.

Dot properties.

- **Dot type.** Each point is drawn using the chosen dot type.
- **Transparency.** The slider sets the transparency of the points and labels from opaque (right) to invisible (left). This can help visualize overlapping points.

Axis ranges. Options for configuring the the ranges of the two axes.

- **Horizontal axis range.** Change the range of the horizontal axis (\log_2 fold change) by setting the **Min** and **Max** values.
- **Vertical axis range.** Change the range of the vertical axis ($-\log_{10}$ p-value) by setting the **Min** and **Max** values.

Note: When the potential number of labels is high, only a subset is shown. Zooming in and out may affect the labels shown.

The plot right-click menu offers the following options, for both **Colored points** and **Selected points**:

- **Select Genes/Transcripts in Other Views.** Selects the relevant features in all other opened views containing the same features, for example another view of the statistical comparison, or an expression browser (section 33.2).
- **Copy Gene/Transcript Names to Clipboard.** Copies the names of the relevant features to the clipboard.

33.6.6 Create Venn Diagram for RNA-Seq

The **Create Venn Diagram** tool makes it possible to compare the overlap of differentially expressed features, such as genes or transcripts, in two or more statistical comparisons. The features considered to be differentially expressed can be controlled by setting appropriate p-value and fold change thresholds.

Note that it is possible to create filtered Venn diagrams from existing ones - for example, showing only genes that are up-regulated in all statistical comparisons - by using the **Create from selection** button on the Venn diagram table view.

To create the Venn diagram:

Tools | RNA-Seq and Small RNA Analysis (📁) | **Differential Expression** (📁) | **Create Venn Diagram for RNA-Seq** (🔗)

Select a number of statistical comparison tracks (📊) or statistical comparison tables (📄) and click **Next** (see figure 33.92 and figure 33.93).

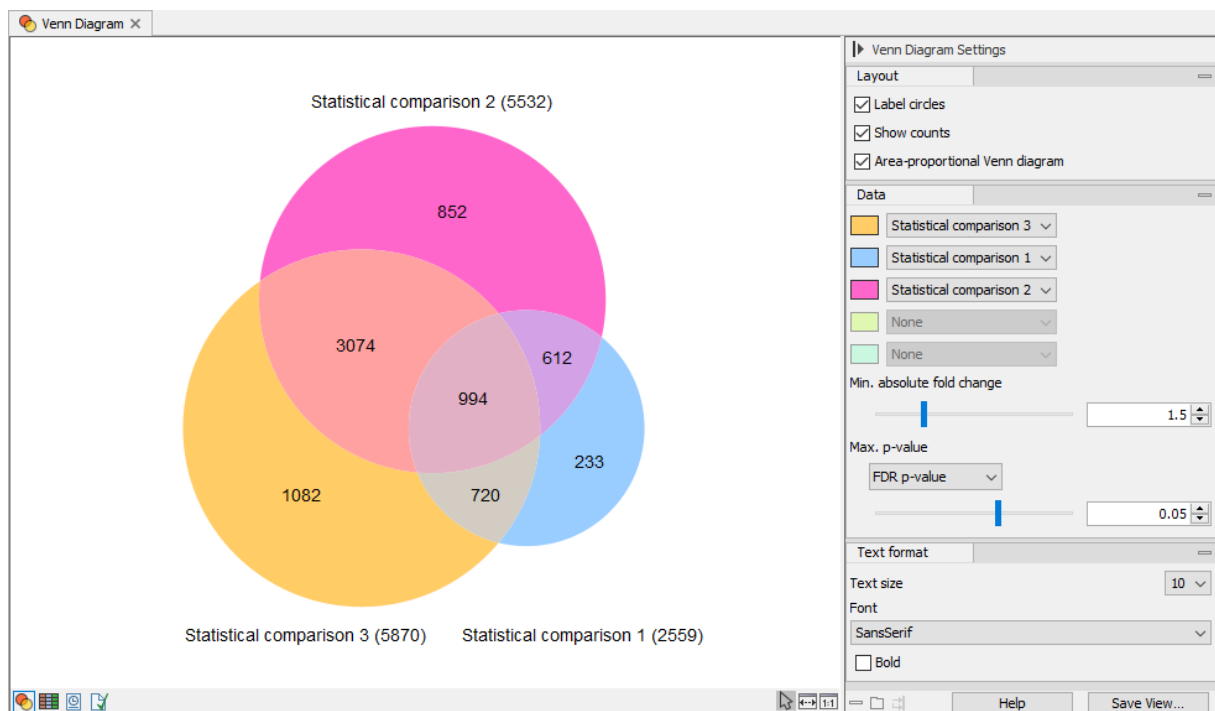


Figure 33.92: A Venn diagram with 3 groups. The circle sizes and overlaps are proportional to the number of overlapping features.

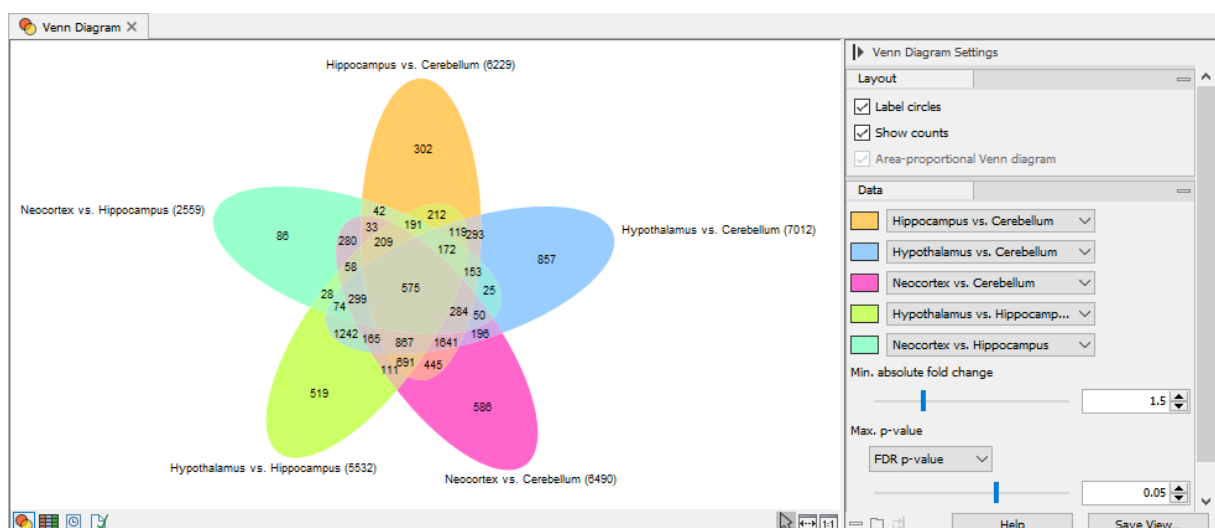


Figure 33.93: A Venn diagram with 5 groups.

In the **Side Panel** to the right, it is possible to adjust the Venn Diagram settings. Under **Layout**, you can adjust the general properties of the plot.

- **Label circles** Toggles the names of the statistical comparisons.
- **Show counts** Toggles the display of feature counts.
- **Area-proportional Venn Diagram** When selected, sizes and positions of the circles are adjusted in proportion to the number of overlapping features. This is only supported for Venn diagrams for two or three groups. Otherwise, ellipses are drawn with fixed positions and identical size.

The **Data** side panel group makes it possible to choose the differentially expressed features of interest. The set of statistical comparisons to be compared can be selected using the drop down menus at the top of the group. The color of a given statistical comparison can be customized using the color picker next to the drop down menu. Note that when comparing two or three samples at a time the circles behave dynamically while selection of four or five samples for comparison provides static visualizations.

- **Min. absolute fold change** Only features with an absolute fold change higher than the specified threshold are taken into account.
- **Max. p-value** Only features with a p-value less than the specified threshold will be taken into account. It is possible to select which p-value measure to use.

Finally, the **Text format** group makes it possible to adjust the settings for the count and statistical comparison labels.



Venn diagram table view

The p-values and fold changes for each feature can be inspected in the Venn diagram table view (see figure 33.94).

Clicking a circle segment in the Venn Diagram plot will select the features in the table view. You can then use the "Filter to selection" button in the Table view to only see the selected rows. It is also possible to create a subset Venn diagram using the **Create from selection** button.

In the **Side Panel** to the right it is possible to adjust the column layout, and select which columns should be included in the table.

33.6.7 Gene Set Test

Gene Set Test takes a Statistical comparison track () and a Gene Ontology Annotation (GOA) table () as input. The tool outputs a GO enrichment analysis table, summarizing the results of hypergeometric tests that evaluate whether terms from the GOA table are over-represented in the set of differentially expressed features from the input track.

To run the tool, go to:

Tools | RNA-Seq and Small RNA Analysis () | **Differential Expression** () | **Gene Set Test** ()

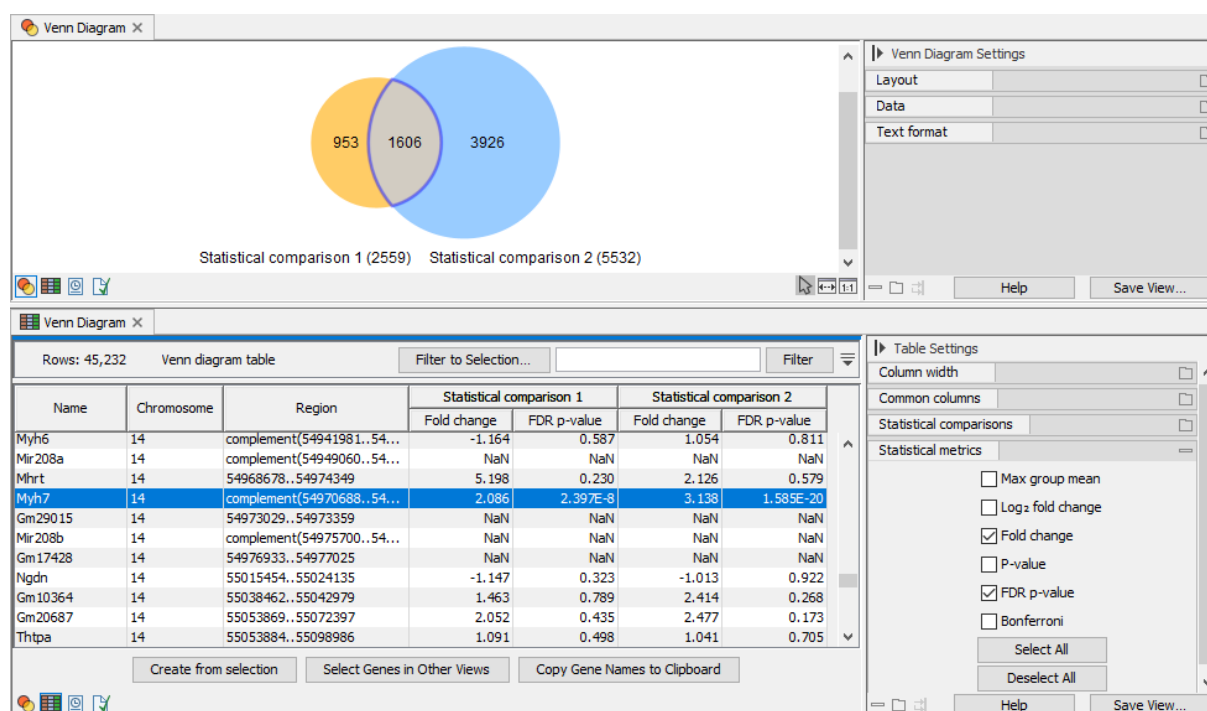


Figure 33.94: The Venn diagram table view.

The following options can be configured in the **Annotation testing parameters** dialog (figure 33.95):

- **GOA**. A Gene Ontology Annotation (GOA) table (📄). The values in the "Feature ID" column must match exactly the feature names in the input track.

Several GOA tables are available via the **QIAGEN Sets** tab, see section 11.2, of the Reference Data Manager. GO annotation files are provided by various sources, including Blast2Go and Gene Ontology, see section K.4. It is also possible to import GOA tables from a tabular file with custom annotations, see section K.5.

- Enriched terms within a specific GO aspect are tested. The following are available:
 - **GO biological process**: the larger processes, or 'biological programs' accomplished by multiple molecular activities. Examples: lipid storage, chemical homeostasis.
 - **GO molecular function**: molecular-level activities performed by gene products. Examples: retinoic acid receptor activity, transcription regulator activity.
 - **GO cellular component**: a location, relative to cellular compartments and structures, occupied by a macromolecular machine. Examples: nuclear inner membrane, ubiquitine ligase complex.
- **Exclude computationally inferred GO terms**. If checked, GO terms with the [IEA] tag are excluded (figure 33.96). These terms are automatically curated and have not been reviewed by a curator.
- **Allow gene name synonyms**. If checked, matching the "Feature ID" column against the feature names allows synonyms and database identifiers.

- **Ignore gene name capitalization.** If checked, matching the "Feature ID" column against the feature names is case-insensitive.

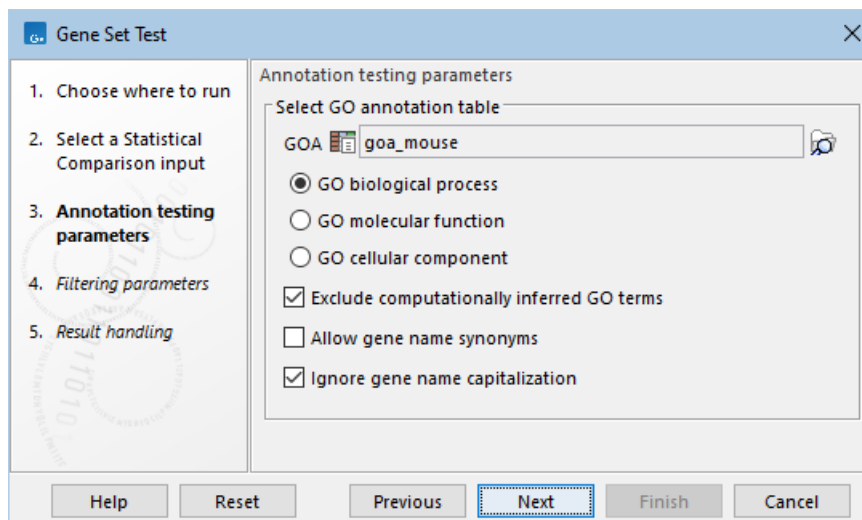


Figure 33.95: Annotation testing parameters.

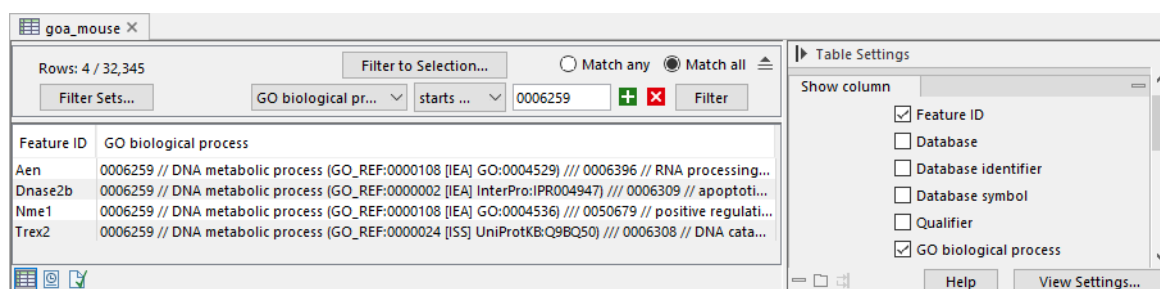


Figure 33.96: GO terms with the [IEA] tag are computationally inferred.

The following options for defining the differentially expressed features can be configured in the **Filtering parameters** dialog (figure 33.97):

- **Ignore features with mean expression values below.** Only features with at least this "Max group mean" are kept.
- **Minimum absolute fold change.** Only features with at least this absolute fold change are kept.
- **Threshold.** Only features with at most this p-value are kept. The type of the p-value can be set to **P-value**, **FDR p-value**, or **Bonferroni**.

The tool outputs a GO enrichment analysis table (figure 33.98). Each row in the table corresponds to a GO term and includes information on the number and names of the detected and differentially expressed (DE) features, as well as the hypergeometric test p-value, FDR, and Bonferroni corrected p-values.

GO terms are organized in a hierarchical structure. For example, the term "GO:0033151 V(D)J recombination" from the Gene Ontology [Ashburner et al., 2000, The Gene Ontology Consortium,

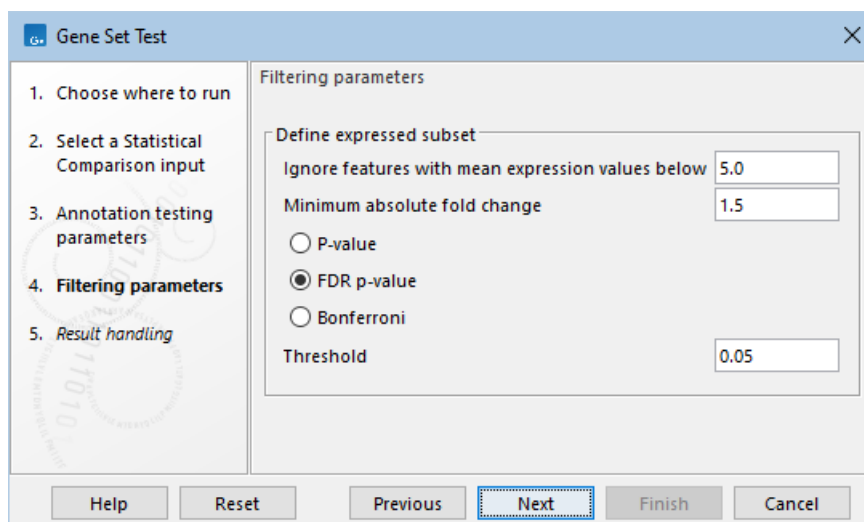


Figure 33.97: Filtering parameters.

The screenshot shows a 'Postpartum vs. Postconception...' window with a 'GO enrichment analysis' tab. The table displays results for 15,201 rows, sorted by FDR p-value. The table has columns for GO term, Description, Detected Genes, DE Genes, and FDR p-value. The right sidebar shows 'Table Settings' with options for column width (Automatic) and column visibility. The 'Show column' section has checkboxes for: GO term (checked), Description (checked), Detected Genes (checked), Detected Genes (Names) (unchecked), DE Genes (checked), DE Genes (Names) (unchecked), P-values (unchecked), FDR p-value (checked), and Bonferroni (unchecked). 'Select All' and 'Deselect All' buttons are also present.

GO term	Description	Detected Genes	DE Genes	FDR p-value
0042744	hydrogen peroxide catabolic process	18	2	0.448
0009713	catechol-containing compound biosynthetic process	21	2	0.448
0042423	catecholamine biosynthetic process	21	2	0.448
0031102	neuron projection regeneration	29	2	0.511
0048821	erythrocyte development	30	2	0.511
0042743	hydrogen peroxide metabolic process	34	2	0.511
0046189	phenol-containing compound biosynthetic process	34	2	0.511
0006584	catecholamine metabolic process	41	2	0.579
0009712	catechol-containing compound metabolic process	41	2	0.579
0003147	neural crest cell migration involved in heart formation	1	1	0.739
0051866	general adaptation syndrome	1	1	0.739
0060648	mammary gland bud morphogenesis	1	1	0.739
0060974	cell migration involved in heart formation	1	1	0.739
0071231	cellular response to folic acid	1	1	0.739
0044763	single-organism cellular process	7815	11	0.923
0061515	myeloid cell development	69	2	0.923
0006835	dicarboxylic acid transport	77	2	1.000
0033555	multicellular organismal response to stress	80	2	1.000
0018958	phenol-containing compound metabolic process	81	2	1.000
0010625	positive regulation of Schwann cell proliferation	2	1	1.000
0031099	regeneration	94	2	1.000

Figure 33.98: GO enrichment analysis table with rows sorted by FDR p-value.

2019] (<https://geneontology.org/>) is a descendant of "GO:0006259 DNA metabolic process".

When testing for the significance of a particular GO term, all features linked to descendant GO terms are included in the test. This can lead to a higher number of detected genes in the output table, compared to the number of genes linked to the tested GO term.


Due to the hierarchical structure, GO terms are not independent of one another, and the p-values provided in the enrichment analysis table should be interpreted with caution.


Chapter 34

Microarray analysis

Contents

34.1 Experimental design	1062
34.1.1 Setting up a microarray experiment	1062
34.1.2 Organization of the experiment table	1065
34.1.3 Adding annotations to an experiment	1071
34.1.4 Scatter plot view of an experiment	1072
34.1.5 Cross-view selections	1073
34.2 Transformation and normalization	1075
34.2.1 Selecting transformed and normalized values for analysis	1075
34.2.2 Transformation	1075
34.2.3 Normalization	1076
34.3 Quality control	1078
34.3.1 Create Box Plot	1079
34.3.2 Hierarchical Clustering of Samples	1082
34.3.3 Principal Component Analysis	1087
34.4 Feature clustering	1090
34.4.1 Hierarchical clustering of features	1090
34.4.2 K-means/medoids clustering	1094
34.5 Statistical analysis - identifying differential expression	1098
34.5.1 Tests on proportions	1098
34.5.2 Gaussian-based tests	1100
34.5.3 Corrected p-values	1101
34.5.4 Volcano plots - inspecting the result of the statistical analysis	1103
34.6 Annotation tests	1105
34.6.1 Hypergeometric Tests on Annotations	1105
34.6.2 Gene Set Enrichment Analysis	1107
34.7 General plots	1111
34.7.1 Histogram	1111
34.7.2 MA plot	1113
34.7.3 Scatter plot	1116

This section focuses on analysing expression data from sources such as microarrays using tools found under the **Microarray Analysis** () folder of the Tools menu. This includes tools for performing quality control of the data, transformation and normalization, statistical analysis to measure differential expression and annotation-based tests. A number of visualization tools such as volcano plots, MA plots, scatter plots, box plots, and heat maps are also available to aid interpretation of the results.

Tools for analysing RNA-Seq and small RNA data are available under the **RNA-Seq and Small RNA Analysis** () folder of the Tools menu and are described in section 33.4.1 and section 33.3.

34.1 Experimental design

Central to expression analysis in the *CLC Genomics Workbench* is the concept of the **sample**. For microarray data, a sample would typically consist of the expression values from one array. Within a sample, there are a number of **features**, usually genes, and their associated expression levels.

Importing expression data into the Workbench as samples is described in appendix section K.

The first step towards analyzing this expression data is to create an **Experiment**, which contains information about which samples belong to which groups.


34.1.1 Setting up a microarray experiment

To analyze differential expression, the Workbench must know which samples belong to which groups. This information is provided in an **Experiment**. Statistical analyses can then be carried out using the information in the Experiment to investigate differential expression. The Experiment element is also where things like t-test results and clustering information are stored.

The statistics created using this tool is for microarray data, RNA-Seq differential expression are better handled by the tools in the folder **Tools | RNA-Seq and Small RNA Analysis** ()

To set up an experiment:

Tools | Microarray Analysis () | **Set Up Microarray Experiment** ()

Select the samples that you wish to use by double-clicking or selecting and pressing the **Add** () button (see figure 34.1).

Clicking **Next** shows the dialog in figure 34.2.

Here you define the experiment type and the number of groups in the experiment.

The options are:

- **Experiment.** At the top you can select a two-group experiment, and below you can select a multi-group experiment and define the number of groups.

Note that you can also specify if the samples are paired. Pairing is relevant if you have samples from the same individual under different conditions, e.g. before and after treatment, or at times 0, 2, and 4 hours after treatment. In this case statistical analysis becomes more efficient if effects of the individuals are taken into account, and comparisons

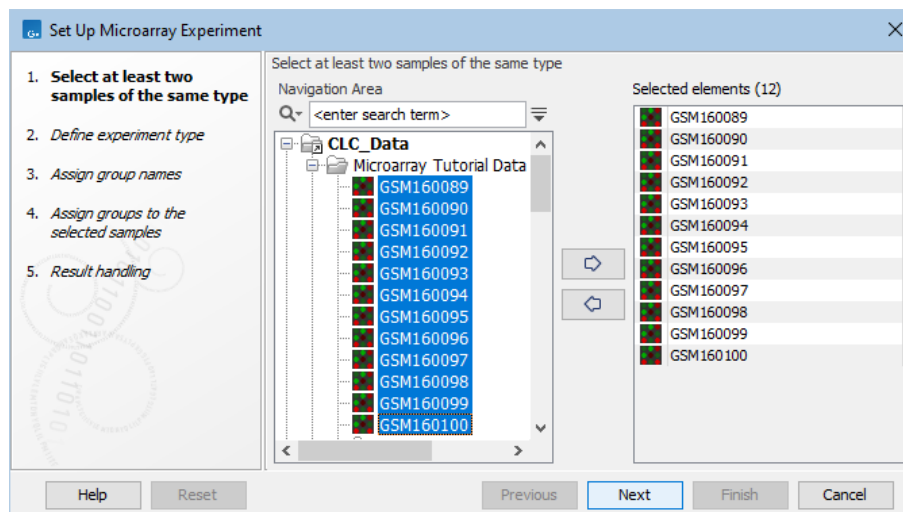


Figure 34.1: Select the samples to use for setting up the experiment.

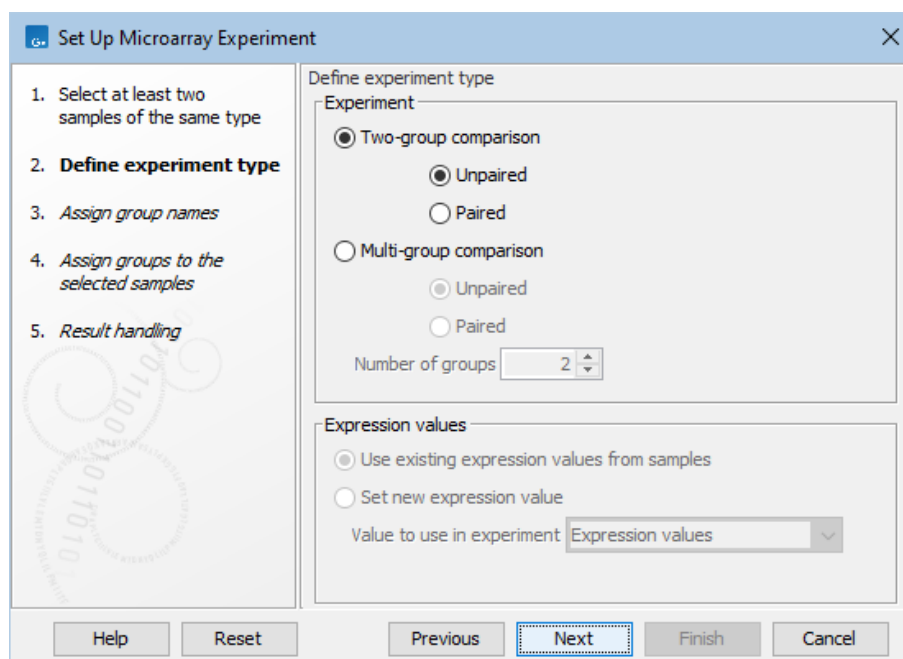


Figure 34.2: Defining the number of groups and expression value type.

are carried out not simply by considering *raw* group means but by considering these *corrected* for effects of the individual. If **Paired** is selected, a paired rather than a standard t-test will be carried out for two group comparisons. For multiple group comparisons a repeated measures rather than a standard ANOVA will be used.

- **Expression values.** If you choose to **Set new expression value** you can choose between the following options depending on whether you look at the gene or transcript level:
 - **Genes: Unique exon reads.** The number of reads that match uniquely to the exons (including the exon-exon and exon-intron junctions).
 - **Genes: Unique gene reads.** This is the number of reads that match uniquely to the gene.
 - **Genes: Total exon reads.** Number of reads mapped to this gene that fall entirely

within an exon or in exon-exon or exon-intron junctions. As for the "Total gene reads" this includes both uniquely mapped reads and reads with multiple matches that were assigned to an exon of this gene.

- **Genes: Total gene reads.** This is all the reads that are mapped to this gene, i.e., both reads that map uniquely to the gene and reads that matched to more positions in the reference (but fewer than the "Maximum number of hits for a read" parameter) which were assigned to this gene.
- **Genes: RPKM.** This is the expression value measured in RPKM [Mortazavi et al., 2008]: $RPKM = \frac{\text{total exon reads}}{\text{mapped reads(millions)} \times \text{exon length (KB)}}$. See exact definition below. Even if you have chosen the RPKM values to be used in the **Expression values** column, they will also be stored in a separate column. This is useful to store the RPKM if you switch the expression measure. See more in section 33.4.1.
- **Transcripts: Unique transcript reads.** This is the number of reads in the mapping for the gene that are uniquely assignable to the transcript. This number is calculated after the reads have been mapped and both single and multi-hit reads from the read mapping may be unique transcript reads.
- **Transcripts: Total transcript reads.** Once the "Unique transcript read's" have been identified and their counts calculated for each transcript, the remaining (non-unique) transcript reads are assigned randomly to one of the transcripts to which they match. The "Total transcript reads" counts are the total number of reads that are assigned to the transcript once this random assignment has been done. As for the random assignment of reads among genes, the random assignment of reads within a gene but among transcripts, is done proportionally to the "unique transcript counts" normalized by transcript length, that is, using the RPKM. Unique transcript counts of 0 are not replaced by 1 for this proportional assignment of non-unique reads among transcripts.
- **Transcripts: RPKM.** The RPKM value for the transcript, that is, the number of reads assigned to the transcript divided by the transcript length and normalized by "Mapped reads" (see below).

Clicking **Next** shows the dialog in figure 34.3.

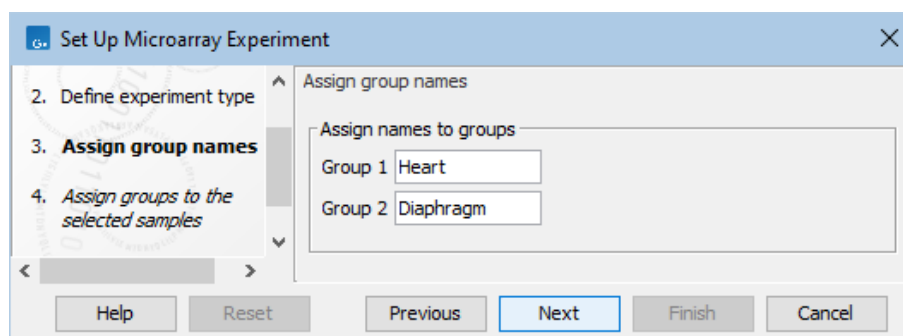


Figure 34.3: Naming the groups.

Depending on the number of groups selected in figure 34.2, you will see a list of groups with text fields where you can enter an appropriate name for that group.

For multi-group experiments, if you find out that you have too many groups, click the **Delete** (X) button. If you need more groups, simply click **Add New Group**.

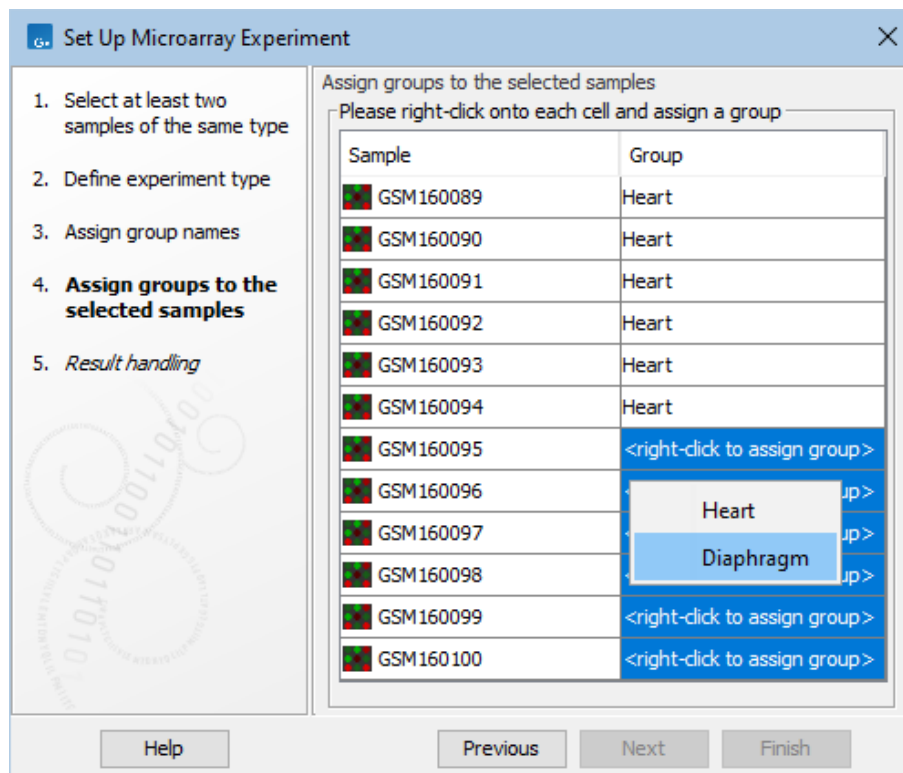


Figure 34.4: Putting the samples into groups.

Click **Next** when you have named the groups, and you will see figure 34.4.

This is where you define which group the individual sample belongs to. Simply select one or more samples (by clicking and dragging the mouse), right-click (Ctrl-click on Mac) and select the appropriate group.

Note that the samples are sorted alphabetically based on their names.

If you have chosen **Paired** in figure 34.2, there will be an extra column where you define which samples belong together. Just as when defining the group membership, you select one or more samples, right-click in the pairing column and select a pair.

Click **Finish** to start the tool.

34.1.2 Organization of the experiment table

The resulting experiment includes all the expression values and other information from the samples (the values are copied - the original samples are not affected and can thus be deleted with no effect on the experiment). In addition it includes a number of summaries of the values across all, or a subset of, the samples for each feature. Which values are included is described in the sections below.

When you open it, it is shown in the experiment table (see figure 34.5).

For a general introduction to table features like sorting and filtering, see section 9.

Unlike other tables in *CLC Genomics Workbench*, the experiment table has a hierarchical grouping of the columns. This is done to reflect the structure of the data in the experiment. The **Side**

The screenshot shows a software interface for microarray analysis. The main window displays a table with 15,923 rows. The table has columns for Feature ID, Total present, Range (original), IQR (original), Difference (original), Fold Change, and two groups of columns for GSM160089 and GSM160090 (Expression values and Presence call). The right-hand side features a 'Experiment Table Settings' panel with options for Column width (Manual), Experiment level (Analysis level, Annotation level, Group level), and checkboxes for Heart, Diaphragm, Group columns, Means, and Present count. There are also buttons for 'Select All' and 'Deselect All'.

Feature ID	Total present...	Range (original)	IQR (original)	Difference (original)	Fold Change...	GSM160089		GSM160090	
						Expression values	Presence call	Expression values	Presence call
1367452_at	12	862.00	470.50	464.82	1.19	2,532.90 P		2,518.60 P	
1367453_at	12	1,231.00	430.80	428.37	1.13	3,464.20 P		3,197.40 P	
1367454_at	12	536.50	349.10	153.85	1.10	1,620.80 P		1,870.50 P	
1367455_at	12	2,196.20	1,352.90	-772.35	-1.17	5,512.50 P		4,103.90 P	
1367456_at	12	2,095.50	1,264.20	-1,205.65	-1.27	6,090.80 P		5,352.20 P	
1367457_at	12	508.20	319.00	-64.73	-1.07	1,093.90 P		1,134.30 P	
1367458_at	12	268.30	148.90	112.30	1.38	347.80 P		223.90 P	
1367459_at	12	3,993.80	2,434.30	2,557.35	1.36	7,665.80 P		7,415.90 P	
1367460_at	12	1,182.80	557.00	-484.73	-1.17	3,155.70 P		2,946.90 P	
1367461_at	12	485.70	280.20	184.35	1.29	507.00 P		610.30 P	
1367462_at	12	1,032.50	309.70	268.23	1.08	3,207.50 P		3,371.30 P	
1367463_at	12	1,621.60	510.00	701.97	1.21	3,510.30 P		3,050.30 P	
1367464_at	12	317.70	202.00	-111.67	-1.13	797.70 P		1,038.90 P	
1367465_at	12	699.00	196.80	257.50	1.22	1,103.10 P		1,281.80 P	
1367466_at	12	265.50	122.50	-54.67	-1.04	1,385.10 P		1,321.30 P	

Figure 34.5: Opening the experiment.

Panel is divided into a number of groups corresponding to the structure of the table. These are described below. Note that you can customize and save the settings of the **Side Panel** (see section 4.6).

Whenever you perform analyses like normalization, transformation, statistical analysis etc, new columns will be added to the experiment. You can at any time **Export** (📄) all the data in the experiment in csv or Excel format or **Copy** (📋) the full table or parts of it.

Column width

There are two options to specify the width of the columns and also the entire table:

- **Automatic.** This will fit the entire table into the width of the view. This is useful if you only have a few columns.
- **Manual.** This will adjust the width of all columns evenly, and it will make the table as wide as it needs to be to display all the columns. This is useful if you have many columns. In this case there will be a scroll bar at the bottom, and you can manually adjust the width by dragging the column separators.

Experiment level

The rest of the **Side Panel** is devoted to different levels of information on the values in the experiment. The experiment part contains a number of columns that, for each feature ID, provide summaries of the values across all the samples in the experiment (see figure 34.6).

Initially, it has one header for the whole **Experiment**:

- **Range (original values).** The 'Range' column contains the difference between the highest and the lowest expression value for the feature over all the samples. If a feature has the value NaN in one or more of the samples the range value is NaN.
- **IQR (original values).** The 'IQR' column contains the inter-quantile range of the values for a feature across the samples, that is, the difference between the 75 %-ile value and the 25 %-ile value. For the IQR values, only the numeric values are considered when percentiles are calculated (that is, NaN and +Inf or -Inf values are ignored), and if there are fewer than

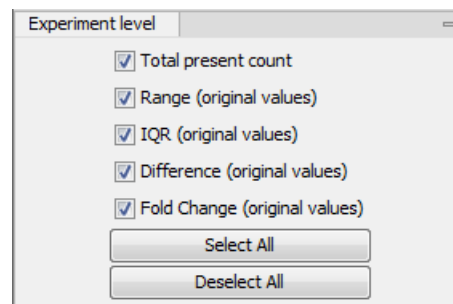


Figure 34.6: *The initial view of the experiment level for a two-group experiment.*

four samples with numeric values for a feature, the IQR is set to be the difference between the highest and lowest of these.

- **Difference (original values).** For a two-group experiment the 'Difference' column contains the difference between the mean of the expression values across the samples assigned to group 2 and the mean of the expression values across the samples assigned to group 1. Thus, if the mean expression level in group 2 is higher than that of group 1 the 'Difference' is positive, and if it is lower the 'Difference' is negative. For experiments with more than two groups the 'Difference' contains the difference between the maximum and minimum of the mean expression values of the groups, multiplied by -1 if the group with the maximum mean expression value occurs before the group with the minimum mean expression value (with the ordering: group 1, group 2, ...).
- **Fold Change (original values).** For a two-group experiment the 'Fold Change' tells you how many times bigger the mean expression value in group 2 is relative to that of group 1. If the mean expression value in group 2 is bigger than that in group 1 this value is the mean expression value in group 2 divided by that in group 1. If the mean expression value in group 2 is smaller than that in group 1 the fold change is the mean expression value in group 1 divided by that in group 2 with a negative sign. Thus, if the mean expression levels in group 1 and group 2 are 10 and 50 respectively, the fold change is 5, and if the and if the mean expression levels in group 1 and group 2 are 50 and 10 respectively, the fold change is -5. Entries of plus or minus infinity in the 'Fold Change' columns of the Experiment area represent those where one of the expression values in the calculation is a 0. For experiments with more than two groups, the 'Fold Change' column contains the ratio of the maximum of the mean expression values of the groups to the minimum of the mean expression values of the groups, multiplied by -1 if the group with the maximum mean expression value occurs before the group with the minimum mean expression value (with the ordering: group 1, group 2, ...).

Thus, the sign of the values in the 'Difference' and 'Fold change' columns give the direction of the trend across the groups, going from group 1 to group 2, etc.

If the samples used are Affymetrix GeneChips samples and have 'Present calls' there will also be a 'Total present count' column containing the number of present calls for all samples.

The columns under the 'Experiment' header are useful for filtering purposes, e.g. you may wish to ignore features that differ too little in expression levels to be confirmed e.g. by qPCR by filtering on the values in the 'Difference', 'IQR' or 'Fold Change' columns or you may wish to ignore features that do not differ at all by filtering on the 'Range' column.

If you have performed normalization or transformation (see sections 34.2.3 and 34.2.2, respectively), the IQR of the normalized and transformed values will also appear. Also, if you later choose to transform or normalize your experiment, columns will be added for the transformed or normalized values.

Note! It is very common to filter features on fold change values in expression analysis and fold change values are also used in volcano plots, see section 34.5.4. There are different definitions of 'Fold Change' in the literature. The definition that is used typically depends on the original scale of the data that is analyzed. For data whose original scale is *not* the log scale the standard definition is the ratio of the group means [Tusher et al., 2001]. This is the value you find in the 'Fold Change' column of the experiment. However, for data whose original *is* the log scale, the difference of the mean expression levels is sometimes referred to as the fold change [Guo et al., 2006], and if you want to filter on fold change for these data you should filter on the values in the 'Difference' column. Your data's original scale will e.g. be the log scale if you have imported Affymetrix expression values which have been created by running the RMA algorithm on the probe-intensities.

Analysis level

The results of each statistical test performed are in the columns listed in this area. In the table, a heading is given for each test. Information about the results of statistical tests are described in the statistical analysis section (see section 34.5).

An example of Analysis level settings is shown in figure 34.7.

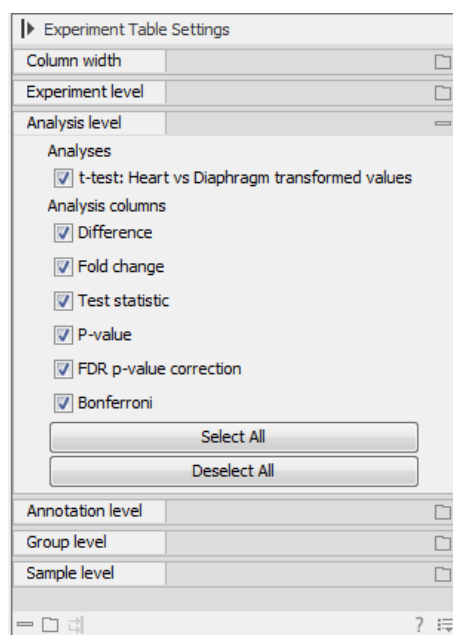


Figure 34.7: An example of columns available under the Analysis level section.

Note: Some column names here are the same as ones under the Experiment level, but the results here are from statistical tests, while those under the Experiment level section are calculations carried out directly on the expression levels.

Annotation level

If your experiment is annotated (see section 34.1.3), the annotations will be listed in the **Annotation level** group as shown in figure 34.8.

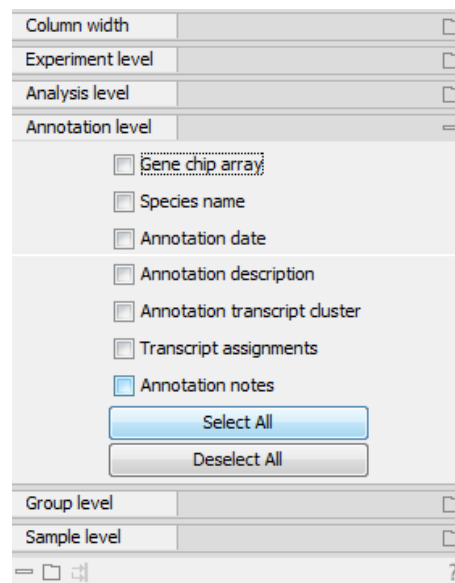


Figure 34.8: An annotated experiment.

In order to avoid too much detail and cluttering the table, only a few of the columns are shown per default.

Note that if you wish a different set of annotations to be displayed each time you open an experiment, you need to save the settings of the **Side Panel** (see section 4.6).

Group level

At the group level, you can show/hide entire groups (*Heart* and *Diaphragm* in figure 34.5). This will show/hide everything under the group's header. Furthermore, you can show/hide group-level information like the group means and present count within a group. If you have performed normalization or transformation (see sections 34.2.3 and 34.2.2, respectively), the means of the normalized and transformed values will also appear.

An example is shown in figure 34.9.

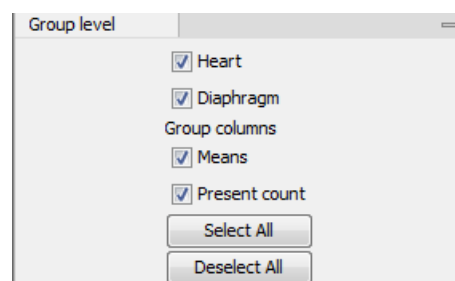


Figure 34.9: Group level .

Sample level

In this part of the side panel, you can control which columns to be displayed for each sample. Initially this is the all the columns in the samples.

If you have performed normalization or transformation (see sections 34.2.3 and 34.2.2, respectively), the normalized and transformed values will also appear.

An example is shown in figure 34.10.

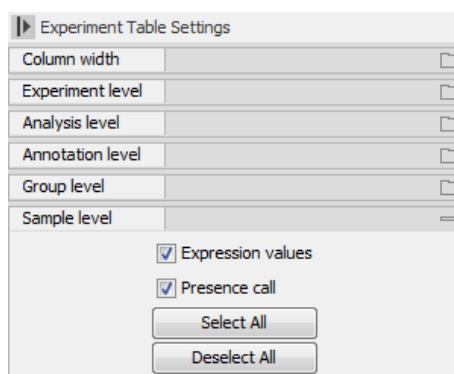


Figure 34.10: Sample level when transformation and normalization has been performed.

Creating a sub-experiment from a selection

If you have identified a list of genes that you believe are differentially expressed, you can create a subset of the experiment. (Note that the filtering and sorting may come in handy in this situation, see section 9).

To create a sub-experiment, first select the relevant features (rows). If you have applied a filter and wish to select all the visible features, press Ctrl + A (⌘ + A on Mac). Next, press the **Create Experiment from Selection** (📊) button at the bottom of the table (see figure 34.11).

1	1160	186	341	175	330
1	1212	100	794	85	767
1	795	506	559	498	549
1	1116	427	438	421	422
1	3732	965	970	930	934
1	1827	68	68	64	64
1	2391	1840	1874	1816	1846
1	1635	28	35	14	14
1	6292	715	740	626	630
1	667	667	667	667	667

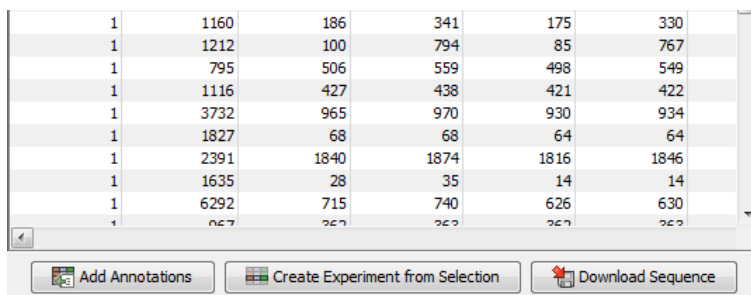
Figure 34.11: Create a subset of the experiment by clicking the button at the bottom of the experiment table.

This will create a new experiment that has the same information as the existing one but with less features.

Downloading sequences from the experiment table

If your experiment is annotated, you will be able to download the GenBank sequence for features which have a GenBank accession number in the 'Public identifier tag' annotation column. To do

this, select a number of features (rows) in the experiment and then click **Download Sequence** (📄) (see figure 34.12).



1	1160	186	341	175	330
1	1212	100	794	85	767
1	795	506	559	498	549
1	1116	427	438	421	422
1	3732	965	970	930	934
1	1827	68	68	64	64
1	2391	1840	1874	1816	1846
1	1635	28	35	14	14
1	6292	715	740	626	630
1	667	267	267	267	267

Figure 34.12: Select sequences and press the download button.

This will open a dialog where you specify where the sequences should be saved. You can learn more about opening and viewing sequences in chapter 15. You can now use the downloaded sequences for further analysis in the Workbench.

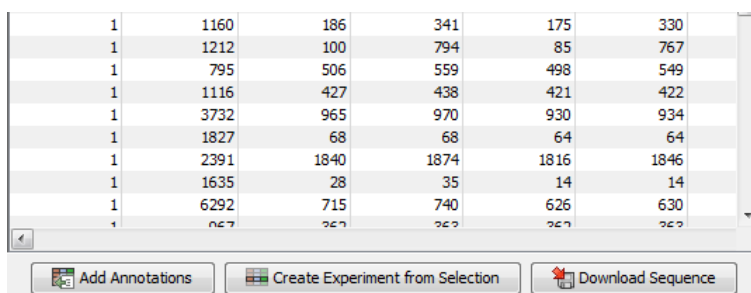
34.1.3 Adding annotations to an experiment

Annotation files provide additional information about each feature. This information could be which GO categories the protein belongs to, which pathways, various transcript and protein identifiers etc. See section K for information about the different annotation file formats that are supported *CLC Genomics Workbench*.

The annotation file can be imported into the Workbench and will get a special icon (📄). See an overview of annotation formats supported by *CLC Genomics Workbench* in section K. In order to associate an annotation file with an experiment, either select the annotation file when you set up the experiment (see section 34.1.1), or click:

Tools | Microarray Analysis (📄) | Annotation Test (📄) | Add Annotations (📄)

Select the experiment (📄) and the annotation file (📄) and click **Finish**. You will now be able to see the annotations in the experiment as described in section 34.1.2. You can also add annotations by pressing the **Add Annotations** (📄) button at the bottom of the table (see figure 34.13).



1	1160	186	341	175	330
1	1212	100	794	85	767
1	795	506	559	498	549
1	1116	427	438	421	422
1	3732	965	970	930	934
1	1827	68	68	64	64
1	2391	1840	1874	1816	1846
1	1635	28	35	14	14
1	6292	715	740	626	630
1	667	267	267	267	267

Figure 34.13: Adding annotations by clicking the button at the bottom of the experiment table.

This will bring up a dialog where you can select the annotation file that you have imported together with the experiment you wish to annotate. Click **Next** to specify settings as shown in figure 34.14).

In this dialog, you can specify how to match the annotations to the features in the sample. The Workbench looks at the columns in the annotation file and lets you choose which column that

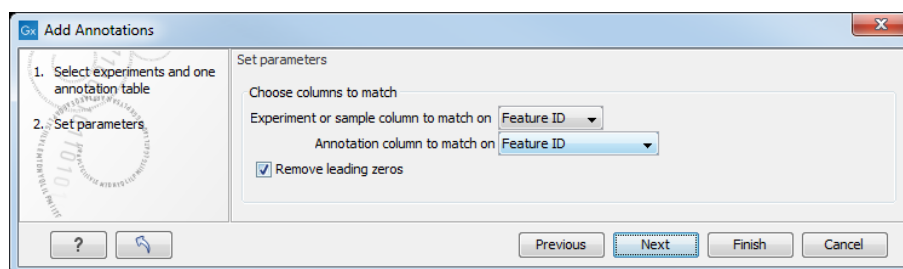


Figure 34.14: Choosing how to match annotations with samples.

should be used for matching to the feature IDs in the experimental data (experiment or sample) as well as for the annotations. Usually the default is right, but for some annotation files, you need to select another column.

Some annotation files have leading zeros in the identifier which you can remove by checking the **Remove leading zeros** box.

Note! Existing annotations on the experiment will be overwritten.

34.1.4 Scatter plot view of an experiment

At the bottom of the experiment table, you can switch between different views of the experiment (see figure 34.15).

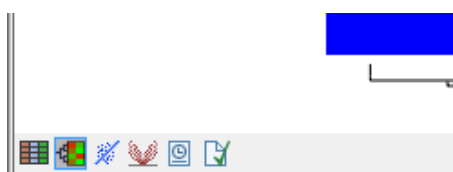


Figure 34.15: An experiment can be viewed in several ways.

One of the views is the **Scatter Plot** (scatter plot icon). The scatter plot can be adjusted to show e.g. the group means for two groups (see more about how to adjust this below).

An example of a scatter plot is shown in figure 34.16.

In the **Side Panel** to the left, there are a number of options to adjust this view. Under **Graph preferences**, you can adjust the general properties of the scatter plot:

- **Lock axes** This will always show the axes even though the plot is zoomed to a detailed level.
- **Frame** Shows a frame around the graph.
- **Show legends** Shows the data legends.
- **Tick type** Determine whether tick lines should be shown outside or inside the frame.
- **Tick lines at** Choosing Major ticks will show a grid behind the graph.
- **Horizontal axis range** Sets the range of the horizontal axis (x axis). Enter a value in **Min** and **Max**, and press Enter. This will update the view. If you wait a few seconds without pressing Enter, the view will also be updated.

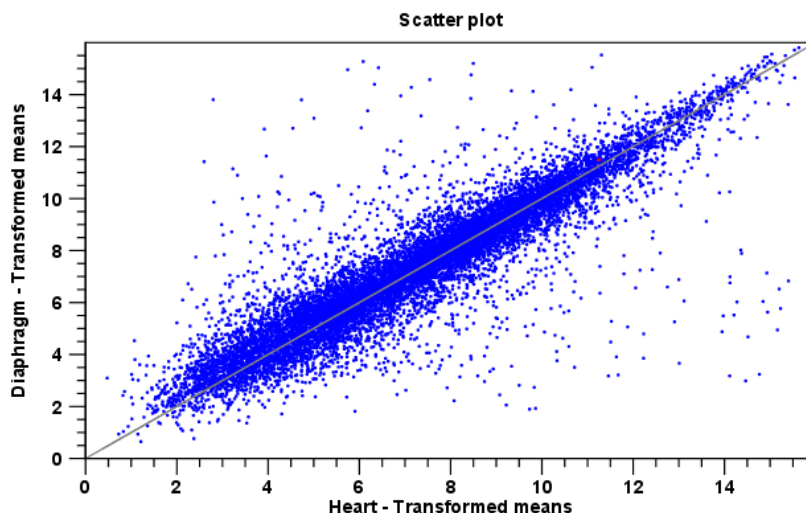


Figure 34.16: A scatter plot of group means for two groups (transformed expression values).

- **Vertical axis range** Sets the range of the vertical axis (y axis). Enter a value in **Min** and **Max**, and press Enter. This will update the view. If you wait a few seconds without pressing Enter, the view will also be updated.
- **Draw $x = y$ axis.** This will draw a diagonal line across the plot. This line is shown per default.
- **Line width** Thin, Medium or Wide
- **Line type** None, Line, Long dash or Short dash
- **Line color** Click the color box to select a color.
- **Show Pearson correlation** When checked, the Pearson correlation coefficient (r) is displayed on the plot.

Below the general preferences, you find the **Dot properties** preferences, where you can adjust coloring and appearance of the dots:

- **Dot type** Can be None, Cross, Plus, Square, Diamond, Circle, Triangle, Reverse triangle, or Dot.
- **Dot color** Click the color box to select a color.

Finally, the group at the bottom - **Values to plot** - is where you choose the values to be displayed in the graph. The default for a two-group experiment is to plot the group means.

Note that if you wish to use the same settings next time you open a scatter plot, you need to save the settings of the **Side Panel** (see section 4.6).

34.1.5 Cross-view selections

There are a number of different ways of looking at an experiment as shown in figure 34.17).



Figure 34.17: An experiment can be viewed in several ways.

Beside the **Experiment table** (table icon) which is the default view, the views are: **Scatter plot** (scatter plot icon), **Volcano plot** (volcano plot icon) and the **Heat map** (heatmap icon). By pressing and holding the Ctrl (⌘ on Mac) button while you click one of the view buttons in figure 34.17, you can make a split view. This will make it possible to see e.g. the experiment table in one view and the volcano plot in another view.

An example of such a split view is shown in figure 34.18.

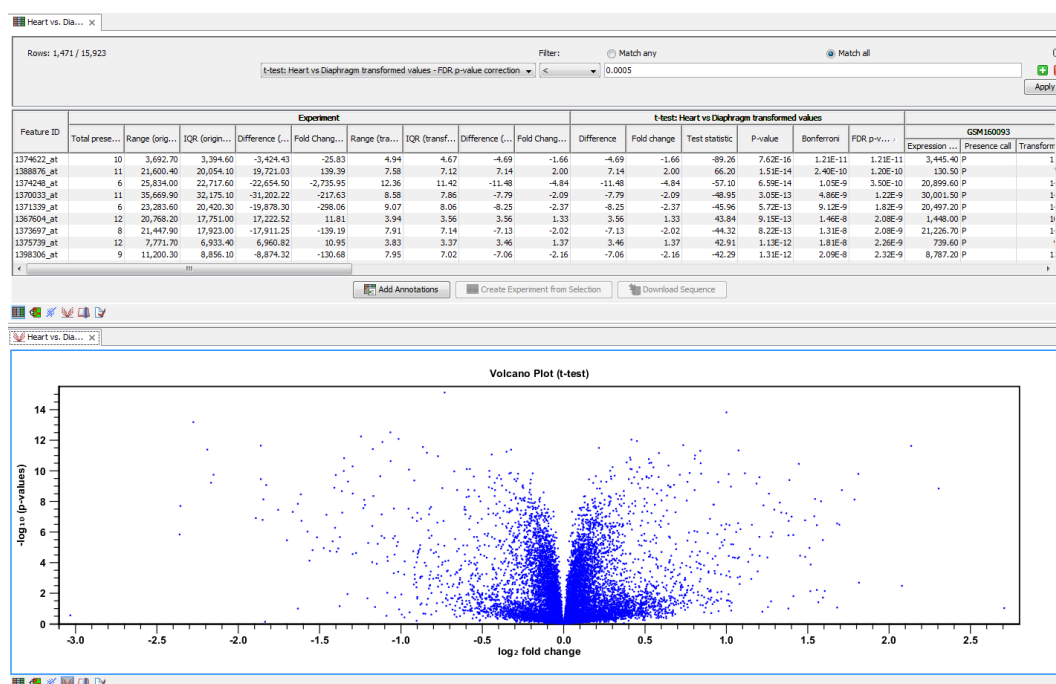


Figure 34.18: A split view showing an experiment table at the top and a volcano plot at the bottom (note that you need to perform statistical analysis to show a volcano plot, see section 34.5).

Selections are shared between all these different views of an experiment. This means that if you select a number of rows in the table, the corresponding dots in the scatter plot, volcano plot or heatmap will also be selected. The selection can be made in any view, also the heatmap, and all other open views will reflect the selection.

A common use of the split views is where you have an experiment and have performed a statistical analysis. You filter the experiment to identify all genes that have an FDR corrected p-value below 0.05 and a fold change for the test above say, 2. You can select all the rows in the experiment table satisfying these filters by holding down the Ctrl button and clicking 'a'. If you have a split view of the experiment and the volcano plot all points in the volcano plot corresponding to the selected features will be red. Note that the volcano plot allows two sets of values in the columns under the test you are considering to be displayed on the x-axis: the 'Fold change's and the 'Difference's. You control which to plot in the side panel. If you have filtered on 'Fold change' you will typically want to choose 'Fold change' in the side panel. If you have filtered on 'Difference'

(e.g. because your original data is on the log scale, see the note on fold change in 34.1.2) you typically want to choose 'Difference'.

34.2 Transformation and normalization

The original expression values often need to be transformed and/or normalized in order to ensure that samples are comparable and assumptions on the data for analysis are met [Allison et al., 2006]. These are essential requirements for carrying out a meaningful analysis. The raw expression values often exhibit a strong dependency of the variance on the mean, and it may be preferable to remove this by log-transforming the data. Furthermore, the sets of expression values in the different samples in an experiment may exhibit systematic differences that are likely due to differences in sample preparation and array processing, rather being the result of the underlying biology. These noise effects should be removed before statistical analysis is carried out.

When you perform transformation and normalization, the original expression values will be kept, and the new values will be added. If you select an experiment (📊), the new values will be added to the experiment (not the original samples). And likewise if you select a sample (🇬🇧) or (🇺🇸) - in this case the new values will be added to the sample (the original values are still kept on the sample).

34.2.1 Selecting transformed and normalized values for analysis

A number of the tools for Expression Analysis use the following expression level values: *Original*, *Transformed* and *Normalized* (figure 34.19).

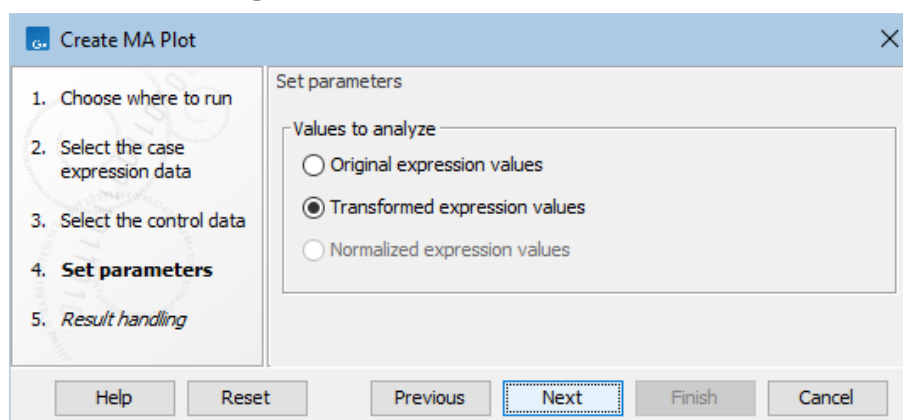


Figure 34.19: Selecting which version of the expression values to analyze. In this case, the values have not been normalized, so it is not possible to select normalized values.

In this case, the values have not been normalized, so it is not possible to select normalized values.

34.2.2 Transformation

The *CLC Genomics Workbench* lets you transform expression values based on logarithm and adding a constant:

Tools | Microarray Analysis (📊) | Transformation and Normalization | Transform (🔧)

Select a number of samples ( or () or an experiment () and click **Next**.

This will display a dialog as shown in figure 34.20.

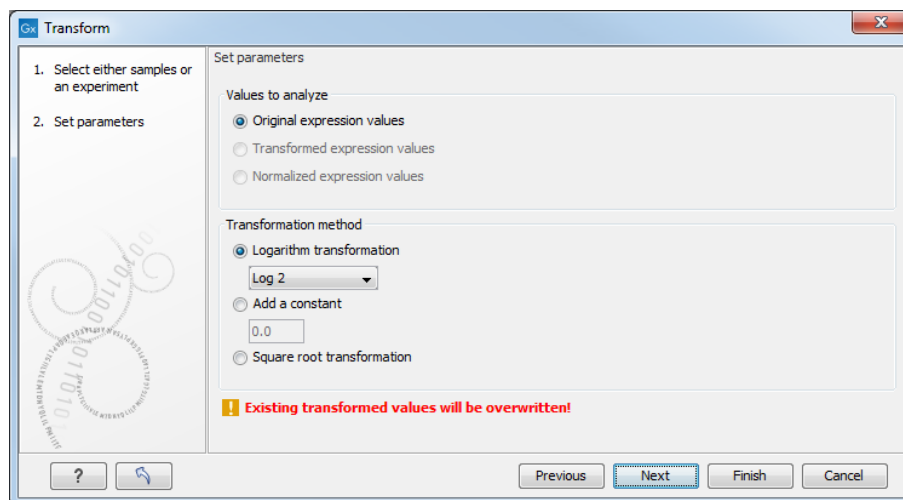


Figure 34.20: Transforming expression values.

At the top, you can select which values to transform (see section 34.2.1).

Next, you can choose three kinds of transformation:

- **Logarithm transformation.** Transformed expression values will be calculated by taking the logarithm (of the specified type) of the values you have chosen to transform.
 - **10.**
 - **2.**
 - **Natural logarithm.**
- **Adding a constant.** Transformed expression values will be calculated by adding the specified constant to the values you have chosen to transform.
- **Square root transformation.**

Click **Finish** to start the tool.

34.2.3 Normalization

The *CLC Genomics Workbench* lets you normalize expression values.

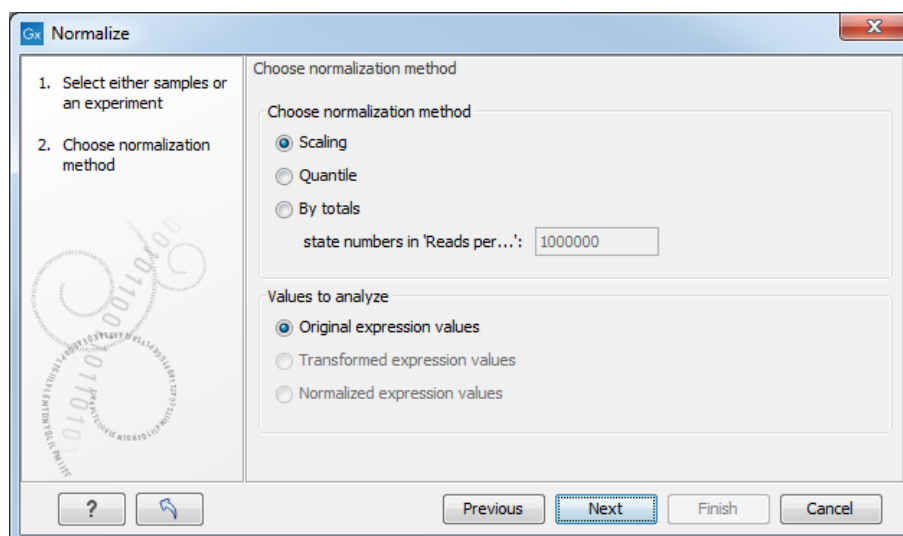
To start the normalization:

Tools | Microarray Analysis () | Transformation and Normalization | Normalize ()

Select a number of samples () or () or an experiment () and click **Next**.

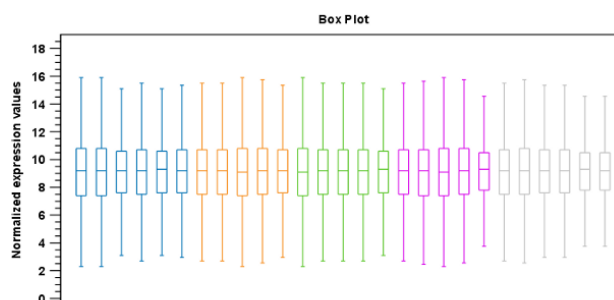
This will display a dialog as shown in figure 34.21.

At the top, you can choose three kinds of normalization (for mathematical descriptions see [Bolstad et al., 2003]):

Figure 34.21: *Choosing normalization method.*

- **Scaling.** The sets of the expression values for the samples will be multiplied by a constant so that the sets of normalized values for the samples have the same 'target' value (see description of the **Normalization value** below).
- **Quantile.** The empirical distributions of the sets of expression values for the samples are used to calculate a common target distribution, which is used to calculate normalized sets of expression values for the samples.
- **By totals.** This option is intended to be used with count-based data, i.e. data from small RNA or expression profiling by tags. A sum is calculated for the expression values in a sample. The transformed value are generated by dividing the input values by the sample sum and multiplying by the factor (e.g. per '1,000,000').

Figures 34.22 and 34.23 show the effect on the distribution of expression values when using scaling or quantile normalization, respectively.

Figure 34.22: *Box plot after scaling normalization.*

At the bottom of the dialog in figure 34.21, you can select which values to normalize (see section 34.2.1).

Clicking **Next** will display a dialog as shown in figure 34.24.

The following parameters can be set:

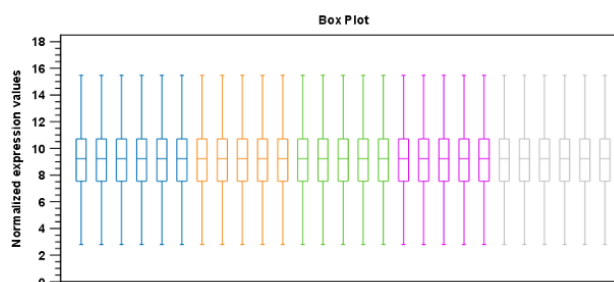


Figure 34.23: Box plot after quantile normalization.

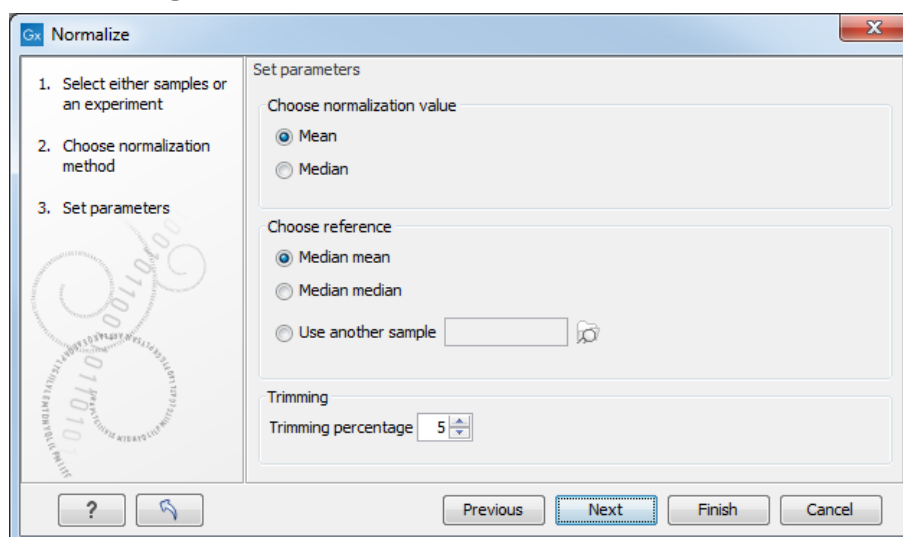


Figure 34.24: Normalization settings.

- **Normalization value.** The type of value of the samples which you want to ensure are equal for the normalized expression values
 - **Mean.**
 - **Median.**
- **Reference.** The specific value that you want the normalized value to be after normalization.
 - **Median mean.**
 - **Median median.**
 - **Use another sample.**
- **Trimming percentage.** Expression values that lie below the value of this percentile, or above 100 minus the value of this percentile, in the empirical distribution of the expression values in a sample will be excluded when calculating the normalization and reference values.

Click **Finish** to start the tool.

34.3 Quality control

The *CLC Genomics Workbench* includes a number of tools for quality control. These allow visual inspection of the overall distributions, variability and similarity of the sets of expression values in

samples, and may be used to spot unwanted systematic differences between samples, outlying samples and samples of poor quality, that you may want to exclude.

34.3.1 Create Box Plot

In most cases you expect the majority of genes to behave similarly under the conditions considered, and only a smaller proportion to behave differently. Thus, at an overall level you would expect the distributions of the sets of expression values in samples in a study to be similar. A boxplot provides a visual presentation of the distributions of expression values in samples. For each sample the distribution of its values is presented by a line representing a center, a box representing the middle part, and whiskers representing the tails of the distribution. Differences in the overall distributions of the samples in a study may indicate that normalization is required before the samples are comparable. An atypical distribution for a single sample (or a few samples), relative to the remaining samples in a study, could be due to imperfections in the preparation and processing of the sample, and may lead you to reconsider using the sample(s).

To create a box plot:

Tools | Microarray Analysis (📊) | Quality Control (📊) | Create Box Plot (📊)

Select a number of samples (📊) or (📊) or an experiment (📊) and click **Next**.

This will display a dialog as shown in figure 34.25.

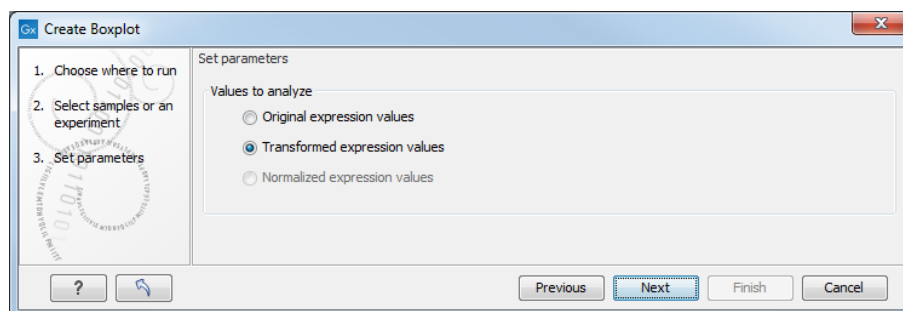


Figure 34.25: Choosing values to analyze for the box plot.

Here you select which values to use in the box plot (see section 34.2.1).

Click **Finish** to start the tool.

Viewing box plots

An example of a box plot of a two-group experiment with 12 samples is shown in figure 34.26.

Note that the boxes are colored according to their group relationship. At the bottom you find the names of the samples, and the y-axis shows the expression values. The box also includes the IQR values (from the lower to the upper quartile) and the median is displayed as a line in the box. The ends of the boxplot whiskers are the lowest data point within 1.5 times the inter quartile range (IQR) of the lower quartile and the highest data point within 1.5 IQR of the upper quartile.

It is possible to change the default value of 1.5 using the side panel option "Whiskers range factor".

In the **Side Panel** to the left, there is a number of options to adjust this view. Under **Graph**

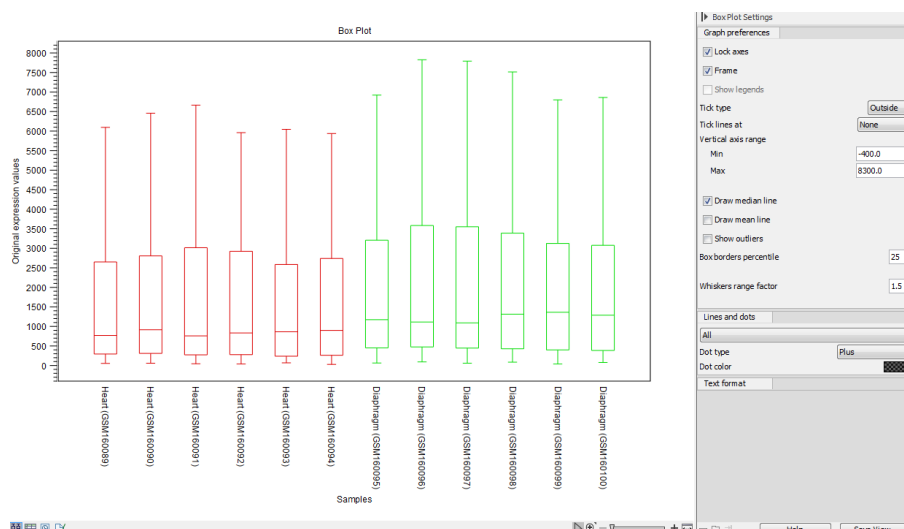


Figure 34.26: A box plot of 12 samples in a two-group experiment, colored by group.

preferences, you can adjust the general properties of the box plot (see figure 34.27).

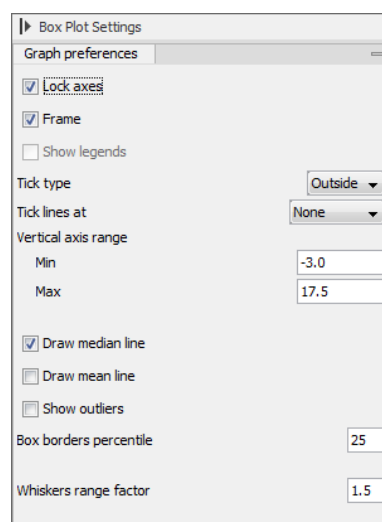


Figure 34.27: Graph preferences for a box plot.

- **Lock axes** This will always show the axes even though the plot is zoomed to a detailed level.
- **Frame** Shows a frame around the graph.
- **Show legends** Shows the data legends.
- **Tick type** Determine whether tick lines should be shown outside or inside the frame.
- **Tick lines at** Choosing Major ticks will show a grid behind the graph.
- **Vertical axis range** Sets the range of the vertical axis (y axis). Enter a value in **Min** and **Max**, and press Enter. This will update the view. If you wait a few seconds without pressing Enter, the view will also be updated.
- **Draw median line.** This is the default - the median is drawn as a line in the box.

- **Draw mean line.** Alternatively, you can also display the mean value as a line.
- **Show outliers.** The values outside the whiskers range are called outliers. Per default they are not shown. Note that the dot type that can be set below only takes effect when outliers are shown. When you select and deselect the **Show outliers**, the vertical axis range is automatically re-calculated to accommodate the new values.

Below the general preferences, you find the **Lines and dots** preferences, where you can adjust coloring and appearance (see figure 34.28).

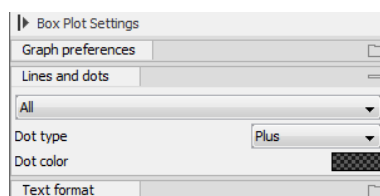


Figure 34.28: *Lines and dot preferences for a box plot.*

- **Select sample or group.** When you wish to adjust the properties below, first select an item in this drop-down menu. That will apply the changes below to this item. If your plot is based on an experiment, the drop-down menu includes both group names and sample names, as well as an entry for selecting "All". If your plot is based on single elements, only sample names will be visible. Note that there are sometimes "mixed states" when you select a group where two of the samples e.g. have different colors. Selecting a new color in this case will erase the differences.
- **Dot type** Can be None, Cross, Plus, Square, Diamond, Circle, Triangle, Reverse triangle, or Dot.
- **Dot color** Click the color box to select a color.

Note that if you wish to use the same settings next time you open a box plot, you need to save the settings of the **Side Panel** (see section 4.6).

Interpreting box plots

This section will show how to interpret a box plot through a few examples.

First, if you look at figure 34.29, you can see a box plot for an experiment with 5 groups and 27 samples.

None of the samples stand out as having distributions that are atypical: the boxes and whiskers ranges are about equally sized. The locations of the distributions however, differ some, and indicate that normalization may be required. Figure 34.30 shows a box plot for the same experiment after quantile normalization: the distributions have been brought into par.

In figure 34.31 a box plot for a two group experiment with 5 samples in each group is shown.

The distribution of values in the second sample from the left is quite different from those of other samples, and could indicate that the sample should not be used.

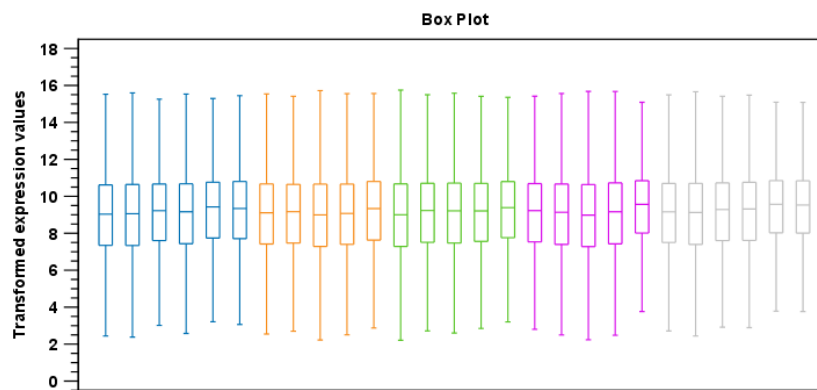


Figure 34.29: Box plot for an experiment with 5 groups and 27 samples.

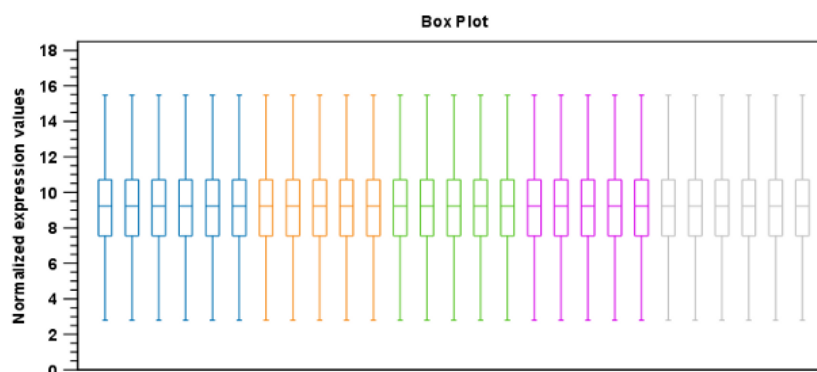


Figure 34.30: Box plot after quantile normalization.

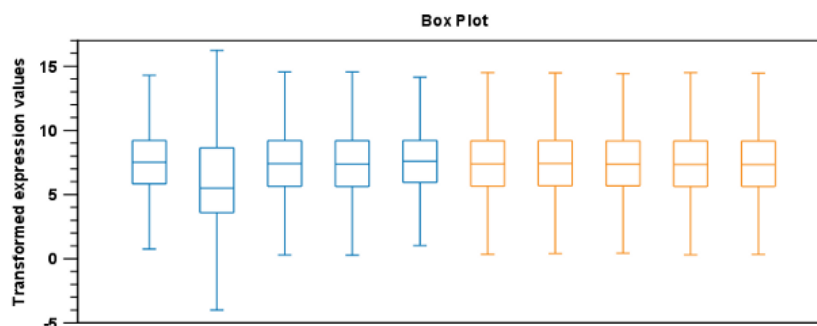


Figure 34.31: Box plot for a two-group experiment with 5 samples.

34.3.2 Hierarchical Clustering of Samples

A hierarchical clustering of samples is a tree representation of their relative similarity.

The tree structure is generated by

1. letting each sample be a cluster
2. calculating pairwise distances between all clusters
3. joining the two closest clusters into one new cluster

4. iterating 2-3 until there is only one cluster left (which will contain all samples).

The tree is drawn so that the distances between clusters are reflected by the lengths of the branches in the tree. Thus, features with expression profiles that closely resemble each other have short distances between them, those that are more different, are placed further apart.

(See [Eisen et al., 1998] for a classical example of application of a hierarchical clustering algorithm in microarray analysis. The example is on features rather than samples).

To start the clustering:

Tools | Microarray Analysis (🔧) | Quality Control (📊) | Hierarchical Clustering of Samples (🔗)

Select a number of samples (🔗) or (🔗) or an experiment (📊) and click **Next**.

This will display a dialog as shown in figure 34.32. The hierarchical clustering algorithm requires that you specify a distance measure and a cluster linkage. The similarity measure is used to specify how distances between two samples should be calculated. The cluster distance metric specifies how you want the distance between two clusters, each consisting of a number of samples, to be calculated.

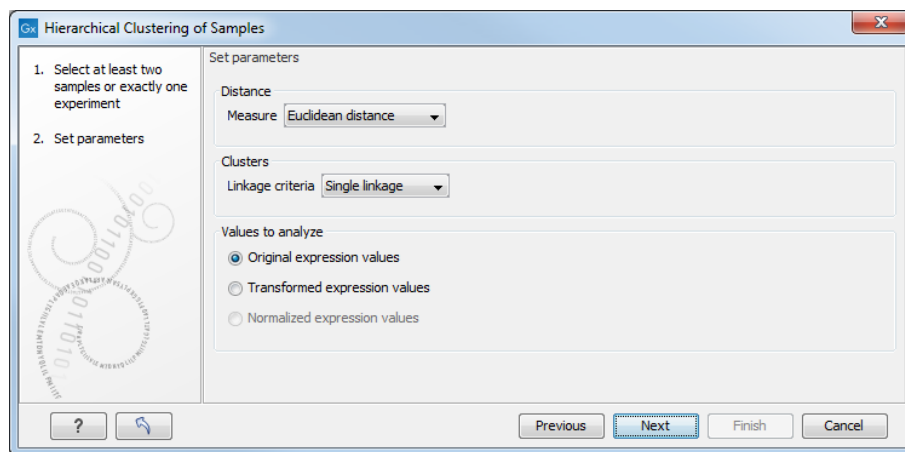


Figure 34.32: Parameters for hierarchical clustering of samples.

There are three kinds of **distance measures**:

- **Euclidean distance.** The length of the segment connecting two points. If $u = (u_1, u_2, \dots, u_n)$ and $v = (v_1, v_2, \dots, v_n)$, then the Euclidean distance between u and v is

$$|u - v| = \sqrt{\sum_{i=1}^n (u_i - v_i)^2}.$$

- **Manhattan distance.** The distance between two points measured along axes at right angles. If $u = (u_1, u_2, \dots, u_n)$ and $v = (v_1, v_2, \dots, v_n)$, then the Manhattan distance between u and v is

$$|u - v| = \sum_{i=1}^n |u_i - v_i|.$$

- **1 - Pearson correlation.** The Pearson correlation coefficient between $x = (x_1, x_2, \dots, x_n)$ and $y = (y_1, y_2, \dots, y_n)$ is defined as

$$r = \frac{1}{n-1} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{s_x} \right) \cdot \left(\frac{y_i - \bar{y}}{s_y} \right)$$

where \bar{x}/\bar{y} and s_x/s_y are the average and sample standard deviation, respectively, of the values in x/y values.

The Pearson correlation coefficient ranges from -1 to 1, with high absolute values indicating strong correlation, and values near 0 suggesting little to no relationship between the elements.

Using $1 - |\text{Pearson correlation}|$ as the distance measure ensures that highly correlated elements have a shorter distance, while elements with low correlation are farther apart.

The distance between two clusters is determined using one of the following linkage types:

- **Single linkage.** The distance between the two closest elements in the two clusters.
- **Average linkage.** The average distance between elements in the first cluster and elements in the second cluster.
- **Complete linkage.** The distance between the two farthest elements in the two clusters.

At the bottom, you can select which values to cluster (see section 34.2.1).

Click **Finish** to start the tool.

Note: To be run on a server, the tool has to be included in a workflow, and the results will be displayed in a stand-alone new heat map rather than added into the input experiment table.

Result of hierarchical clustering of samples

The result of a sample clustering is shown in figure 34.33.

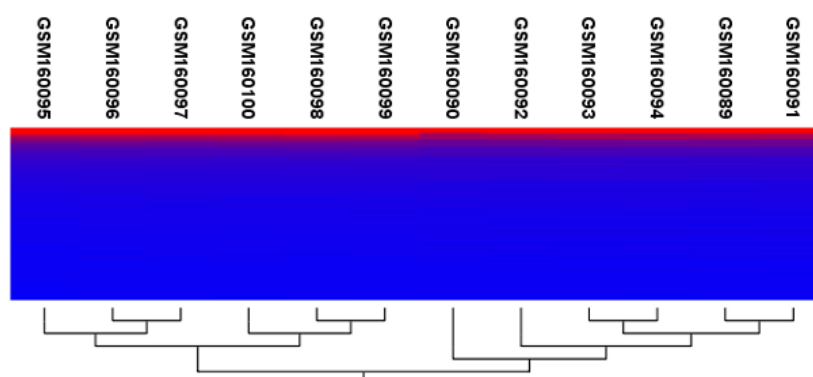




Figure 34.33: Sample clustering.

If you have used an **experiment** (🧪) and ran the non-workflow version of the tool, the clustering is added to the experiment and will be saved when you save the experiment. It can be viewed by clicking the **Show Heat Map** (🔍) button at the bottom of the view (see figure 34.34).



Figure 34.34: Showing the hierarchical clustering of an experiment.

If you have run the workflow version of the tool, or selected a number of **samples** () or () as input, a new element will be created that has to be saved separately.

Regardless of the input, the view of the clustering is the same. As you can see in figure 34.33, there is a tree at the bottom of the view to visualize the clustering. The names of the samples are listed at the top. The features are represented as horizontal lines, colored according to the expression level. If you place the mouse on one of the lines, you will see the names of the feature to the left. The features are sorted by their expression level in the first sample (in order to cluster the features, see section 34.4.1).

Researchers often have a priori knowledge of which samples in a study should be similar (e.g. samples from the same experimental condition) and which should be different (samples from biological distinct conditions). Thus, researchers have expectations about how they should cluster. Samples that are placed unexpectedly in the hierarchical clustering tree may be samples that have been wrongly allocated to a group, samples of unintended or unclear tissue composition or samples for which the processing has gone wrong. Unexpectedly placed samples, of course, could also be highly interesting samples.

There are a number of options to change the appearance of the heat map. At the top of the **Side Panel**, you find the **Heat map** preference group (see figure 34.35).

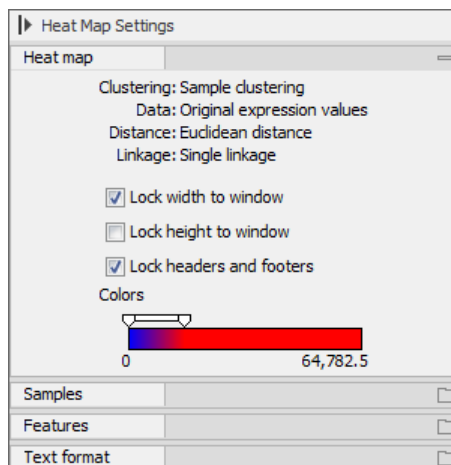


Figure 34.35: Side Panel of heat map.

At the top, there is information about the heat map currently displayed. The information regards type of clustering, expression value used together with distance and linkage information. If you have performed more than one clustering, you can choose between the resulting heat maps in a drop-down box (see figure 34.36).

Note that if you perform an identical clustering, the existing heat map will simply be replaced. Below this box, there is a number of settings for displaying the heat map.

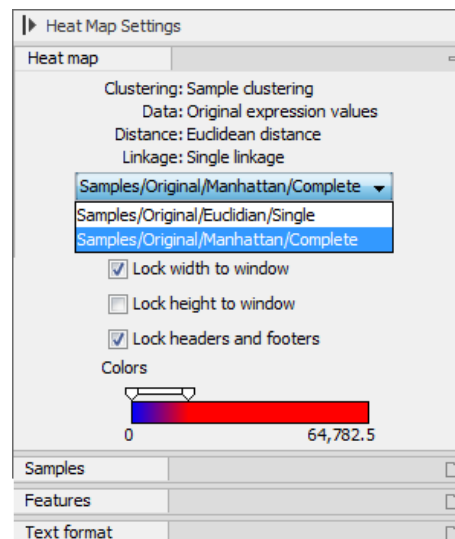


Figure 34.36: When more than one clustering has been performed, there will be a list of heat maps to choose from.

- **Lock width to window.** When you zoom in the heat map, you will per default only zoom in on the vertical level. This is because the width of the heat map is locked to the window. If you uncheck this option, you will zoom both vertically and horizontally. Since you always have more features than samples, it is useful to lock the width since you then have all the samples in view all the time.
- **Lock height to window.** This is the corresponding option for the height. Note that if you check both options, you will not be able to zoom at all, since both the width and the height is fixed.
- **Lock headers and footers.** This will ensure that you are always able to see the sample and feature names and the trees when you zoom in.
- **Colors.** The expression levels are visualized using a gradient color scheme, where the right side color is used for high expression levels and the left side color is used for low expression levels. You can change the coloring by clicking the box, and you can change the relative coloring of the values by dragging the two knobs on the white slider above.

Below you find the **Samples** and **Features** groups. They contain options to show names, legend, and tree above or below the heat map. Note that for clustering of samples, you find the tree options in the **Samples** group, and for clustering of features, you find the tree options in the **Features** group. With the tree options, you can also control the **Tree size**, from tiny to very large, and the option of showing the full tree, no matter how much space it will use.

For clustering of features, the **Features** group has an option to "Optimize tree layout". This attempts to reorder the features, consistently with the tree, such that the most expressed features form a diagonal from the top-left to the bottom-right of the heat map.

The **Samples** group contains an "Order by:" dropdown that allows re-ordering of the columns of the heat map. When clustering by samples it is possible to choose between using the "Tree" to determine the sample ordering, and showing the "Samples" in the order they were input to the tool. When clustering by features, only the "Samples" input order is available.

Note that if you wish to use the same settings next time you open a heat map, you need to save the settings of the **Side Panel** (see section 4.6).

34.3.3 Principal Component Analysis

A principal component analysis is a mathematical analysis that identifies and quantifies the directions of variability in the data. For a set of samples, e.g. an experiment, this can be done either by finding the eigenvectors and eigenvalues of the *covariance matrix* of the samples or the *correlation matrix* of the samples (the correlation matrix is a 'normalized' version of the covariance matrix: the entries in the covariance matrix look like this $Cov(X, Y)$, and those in the correlation matrix like this: $Cov(X, Y)/(sd(X) * sd(Y))$). A covariance maybe any value, but a correlation is always between -1 and 1).

The eigenvectors are orthogonal. The first principal component is the eigenvector with the largest eigenvalue, and specifies the direction with the largest variability in the data. The second principal component is the eigenvector with the second largest eigenvalue, and specifies the direction with the second largest variability. Similarly for the third, etc. The data can be projected onto the space spanned by the eigenvectors. A plot of the data in the space spanned by the first and second principal component will show a simplified version of the data with variability in other directions than the two major directions of variability ignored.

To start the analysis:

Tools | Microarray Analysis (📊) | Quality Control (📊) | Principal Component Analysis (📊)

Select a number of samples (📊) or (📊) or an experiment (📊) and click **Next**.

This will display a dialog as shown in figure 34.37.

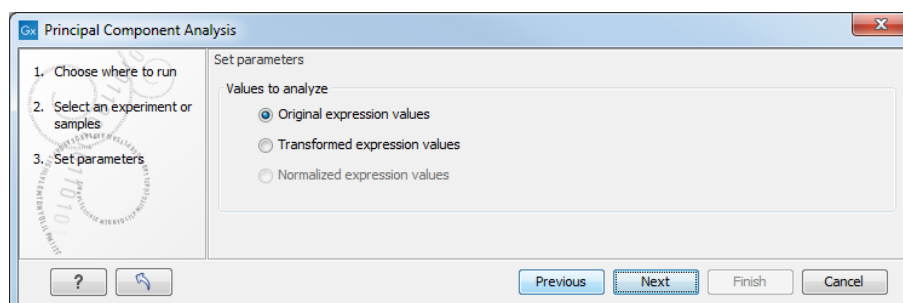


Figure 34.37: Selecting which values the principal component analysis should be based on.

In this dialog, you select the values to be used for the principal component analysis (see section 34.2.1).

Click **Finish** to start the tool.

Principal component analysis plot

This will create a principal component plot as shown in figure 34.38.

The plot shows the projection of the samples onto the two-dimensional space spanned by the first and second principal component of the covariance matrix. In the bottom part of the side-panel, the 'Projection/Correlation' part, you can change to show the projection onto the *correlation*

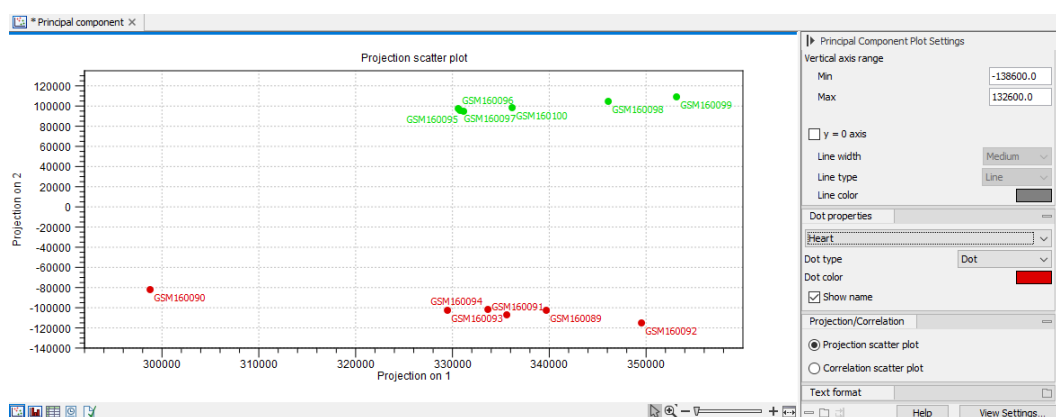


Figure 34.38: A principal component analysis.

matrix rather than the *covariance* matrix by choosing 'Correlation scatter plot'. Both plots will show how the samples separate along the two directions between which the samples exhibit the largest amount of variation. For the 'projection scatter plot' this variation is measured in absolute terms, and depends on the units in which you have measured your samples. The correlation scatter plot is a normalized version of the projection scatter plot, which makes it possible to compare principal component analysis between experiments, even when these have not been done using the same units (e.g an experiment that uses 'original' scale data and another one that uses 'log-scale' data).

The plot in figure 34.38 is based on a two-group experiment. The group relationships are indicated by color. We expect the samples from within a group to exhibit less variability when compared, than samples from different groups. Thus samples should cluster according to groups and this is what we see. The PCA plot is thus helpful in identifying outlying samples and samples that have been wrongly assigned to a group.

In the **Side Panel** to the left, there is a number of options to adjust the view. Under **Graph preferences**, you can adjust the general properties of the plot.

- **Lock axes** This will always show the axes even though the plot is zoomed to a detailed level.
- **Frame** Shows a frame around the graph.
- **Show legends** Shows the data legends.
- **Tick type** Determine whether tick lines should be shown outside or inside the frame.
- **Tick lines at** Choosing Major ticks will show a grid behind the graph.
- **Horizontal axis range** Sets the range of the horizontal axis (x axis). Enter a value in **Min** and **Max**, and press Enter. This will update the view. If you wait a few seconds without pressing Enter, the view will also be updated.
- **Vertical axis range** Sets the range of the vertical axis (y axis). Enter a value in **Min** and **Max**, and press Enter. This will update the view. If you wait a few seconds without pressing Enter, the view will also be updated.
- **y = 0 axis.** Draws a line where y = 0. Below there are some options to control the appearance of the line:


- **Line width** Thin, Medium or Wide
- **Line type** None, Line, Long dash or Short dash
- **Line color** Click the color box to select a color.

Below the general preferences, you find the **Dot properties**:

- **Select sample or group.** When you wish to adjust the properties below, first select an item in this drop-down menu. That will apply the changes below to this item. If your plot is based on an experiment, the drop-down menu includes both group names and sample names, as well as an entry for selecting "All". If your plot is based on single elements, only sample names will be visible. Note that there are sometimes "mixed states" when you select a group where two of the samples e.g. have different colors. Selecting a new color in this case will erase the differences.
- **Dot type** Can be None, Cross, Plus, Square, Diamond, Circle, Triangle, Reverse triangle, or Dot.
- **Dot color** Click the color box to select a color.
- **Show name.** This will show a label with the name of the sample next to the dot. Note that the labels quickly get crowded, so that is why the names are not put on per default.

Note that if you wish to use the same settings next time you open a principal component plot, you need to save the settings of the **Side Panel** (see section 4.6).

Scree plot

Besides the view shown in figure 34.38, the result of the principal component can also be viewed as a scree plot by clicking the **Show Scree Plot**  button at the bottom of the view. The scree plot shows the proportion of variation in the data explained by each of the principal components. The first principal component accounts for the largest part of the variability.


In the **Side Panel** to the left, there is a number of options to adjust the view. Under **Graph preferences**, you can adjust the general properties of the plot.

- **Lock axes** This will always show the axes even though the plot is zoomed to a detailed level.
- **Frame** Shows a frame around the graph.
- **Show legends** Shows the data legends.
- **Tick type** Determine whether tick lines should be shown outside or inside the frame.
- **Tick lines at** Choosing Major ticks will show a grid behind the graph.
- **Horizontal axis range** Sets the range of the horizontal axis (x axis). Enter a value in **Min** and **Max**, and press Enter. This will update the view. If you wait a few seconds without pressing Enter, the view will also be updated.

- **Vertical axis range** Sets the range of the vertical axis (y axis). Enter a value in **Min** and **Max**, and press Enter. This will update the view. If you wait a few seconds without pressing Enter, the view will also be updated.

The **Lines and plots** below contains the following parameters:

- **Dot type** Can be None, Cross, Plus, Square, Diamond, Circle, Triangle, Reverse triangle, or Dot.
- **Dot color** Click the color box to select a color.
- **Line width** Thin, Medium or Wide
- **Line type** None, Line, Long dash or Short dash
- **Line color** Click the color box to select a color.

Note that the graph title and the axes titles can be edited simply by clicking them with the mouse. These changes will be saved when you **Save** () the graph - whereas the changes in the **Side Panel** need to be saved explicitly (see section 4.6).

34.4 Feature clustering

Feature clustering is used to identify and cluster together features with similar expression patterns over samples (or experimental groups). Features that cluster together may be involved in the same biological process or be co-regulated. Also, by examining annotations of genes within a cluster, one may learn about the underlying biological processes involved in the experiment studied.

34.4.1 Hierarchical clustering of features

A hierarchical clustering of features is a tree presentation of the similarity in expression profiles of the features over a set of samples (or groups).




The tree structure is generated by

1. letting each feature be a cluster
2. calculating pairwise distances between all clusters
3. joining the two closest clusters into one new cluster
4. iterating 2-3 until there is only one cluster left (which will contain all samples).

The tree is drawn so that the distances between clusters are reflected by the lengths of the branches in the tree. Thus, features with expression profiles that closely resemble each other have short distances between them, those that are more different, are placed further apart.

To start the clustering of features:

Tools | Microarray Analysis () | **Feature Clustering** () | **Hierarchical Clustering of Features** ()

Select at least two samples () or () or an experiment ().

Note! If your data contains many features, the clustering will take very long time and could make your computer unresponsive. It is recommended to perform this analysis on a subset of the data (which also makes it easier to make sense of the clustering. Typically, you will want to filter away the features that are thought to represent only noise, e.g. those with mostly low values, or with little difference between the samples). See how to create a sub-experiment in section 34.1.2.

Clicking **Next** will display a dialog as shown in figure 34.39. The hierarchical clustering algorithm requires that you specify a distance measure and a cluster linkage. The distance measure is used specify how distances between two features should be calculated. The cluster linkage specifies how you want the distance between two clusters, each consisting of a number of features, to be calculated.

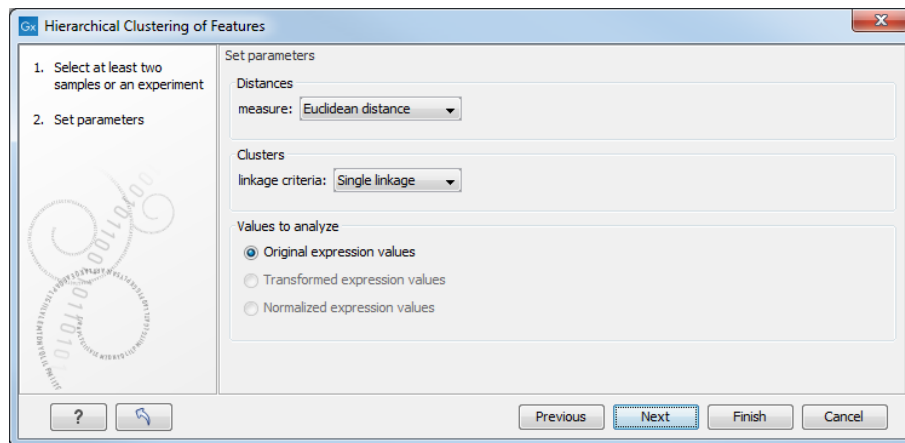


Figure 34.39: Parameters for hierarchical clustering of features.

There are three kinds of **distance measures**:

- **Euclidean distance.** The length of the segment connecting two points. If $u = (u_1, u_2, \dots, u_n)$ and $v = (v_1, v_2, \dots, v_n)$, then the Euclidean distance between u and v is

$$|u - v| = \sqrt{\sum_{i=1}^n (u_i - v_i)^2}.$$

- **Manhattan distance.** The distance between two points measured along axes at right angles. If $u = (u_1, u_2, \dots, u_n)$ and $v = (v_1, v_2, \dots, v_n)$, then the Manhattan distance between u and v is

$$|u - v| = \sum_{i=1}^n |u_i - v_i|.$$

- **1 - Pearson correlation.** The Pearson correlation coefficient between $x = (x_1, x_2, \dots, x_n)$ and $y = (y_1, y_2, \dots, y_n)$ is defined as

$$r = \frac{1}{n-1} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{s_x} \right) \cdot \left(\frac{y_i - \bar{y}}{s_y} \right)$$

where \bar{x}/\bar{y} and s_x/s_y are the average and sample standard deviation, respectively, of the values in x/y values.

The Pearson correlation coefficient ranges from -1 to 1, with high absolute values indicating strong correlation, and values near 0 suggesting little to no relationship between the elements.

Using $1 - |\text{Pearson correlation}|$ as the distance measure ensures that highly correlated elements have a shorter distance, while elements with low correlation are farther apart.

The distance between two clusters is determined using one of the following linkage types:

- **Single linkage.** The distance between the two closest elements in the two clusters.
- **Average linkage.** The average distance between elements in the first cluster and elements in the second cluster.
- **Complete linkage.** The distance between the two farthest elements in the two clusters.

At the bottom, you can select which values to cluster (see section 34.2.1).

Click **Finish** to start the tool.

Result of hierarchical clustering of features

The result of a feature clustering is shown in figure 34.40.

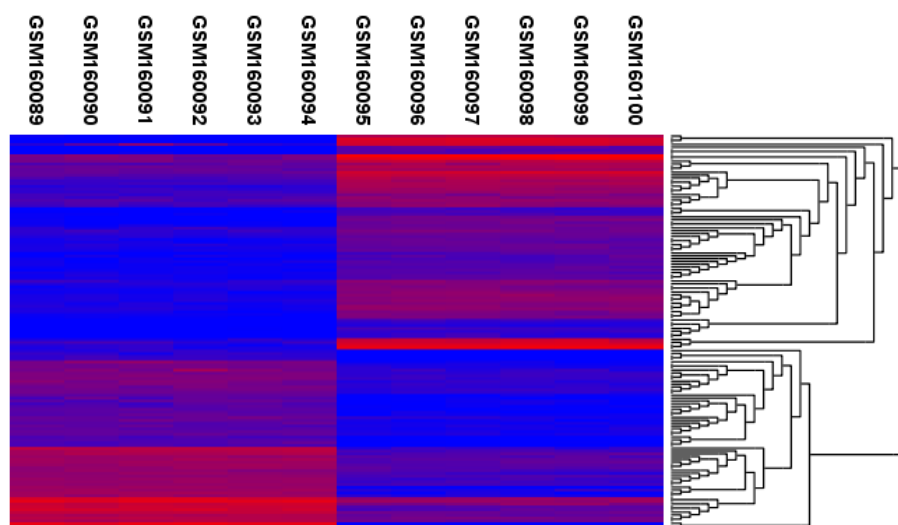


Figure 34.40: Hierarchical clustering of features.

If you have used an **experiment** (🧪) as input, the clustering is added to the experiment and will be saved when you save the experiment. It can be viewed by clicking the **Show Heat Map** (🔍) button at the bottom of the view (see figure 34.41).

If you have selected a number of **samples** (🌿) or (🇺🇸) as input, a new element will be created that has to be saved separately.

Regardless of the input, a hierarchical tree view with associated heatmap is produced (figure 34.40). In the heatmap each row corresponds to a feature and each column to a sample. The



Figure 34.41: Showing the hierarchical clustering of an experiment.

color in the i 'th row and j 'th column reflects the expression level of feature i in sample j (the color scale can be set in the side panel). The order of the rows in the heatmap are determined by the hierarchical clustering. If you place the mouse on one of the rows, you will see the name of the corresponding feature to the left. The order of the columns (that is, samples) is determined by their input order or (if defined) experimental grouping. The names of the samples are listed at the top of the heatmap and the samples are organized into groups.

There are a number of options to change the appearance of the heat map. At the top of the **Side Panel**, you find the **Heat map** preference group (see figure 34.42).

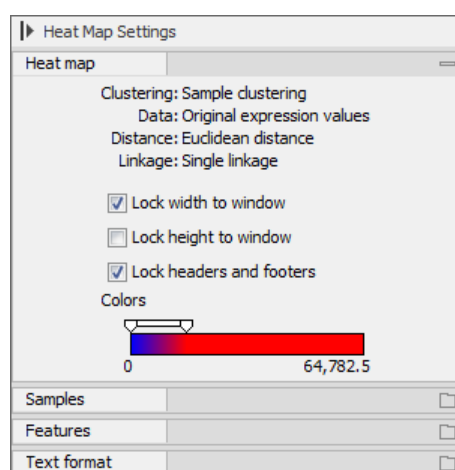


Figure 34.42: Side Panel of heat map.

At the top, there is information about the heat map currently displayed. The information regards type of clustering, expression value used together with distance and linkage information. If you have performed more than one clustering, you can choose between the resulting heat maps in a drop-down box (see figure 34.43).

Note that if you perform an identical clustering, the existing heat map will simply be replaced. Below this box, there is a number of settings for displaying the heat map.

- **Lock width to window.** When you zoom in the heat map, you will per default only zoom in on the vertical level. This is because the width of the heat map is locked to the window. If you uncheck this option, you will zoom both vertically and horizontally. Since you always have more features than samples, it is useful to lock the width since you then have all the samples in view all the time.
- **Lock height to window.** This is the corresponding option for the height. Note that if you check both options, you will not be able to zoom at all, since both the width and the height is fixed.
- **Lock headers and footers.** This will ensure that you are always able to see the sample and

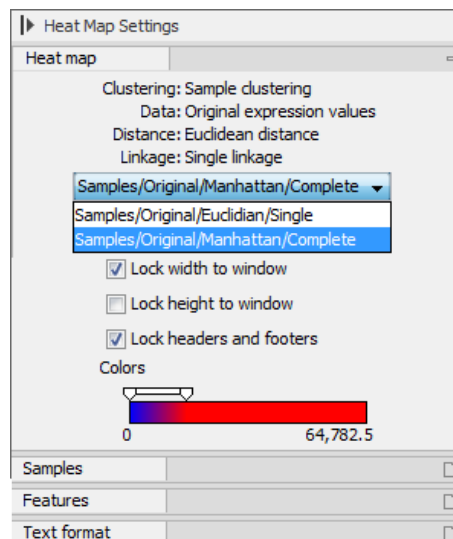


Figure 34.43: When more than one clustering has been performed, there will be a list of heat maps to choose from.

feature names and the trees when you zoom in.

- **Colors.** The expression levels are visualized using a gradient color scheme, where the right side color is used for high expression levels and the left side color is used for low expression levels. You can change the coloring by clicking the box, and you can change the relative coloring of the values by dragging the two knobs on the white slider above.

Below you find the **Samples** and **Features** groups. They contain options to show names, legend, and tree above or below the heat map. Note that for clustering of samples, you find the tree options in the **Samples** group, and for clustering of features, you find the tree options in the **Features** group. With the tree options, you can also control the **Tree size**, from tiny to very large, and the option of showing the full tree, no matter how much space it will use.

For clustering of features, the **Features** group has an option to "Optimize tree layout". This attempts to reorder the features, consistently with the tree, such that the most expressed features form a diagonal from the top-left to the bottom-right of the heat map.




The **Samples** group contains an "Order by:" dropdown that allows re-ordering of the columns of the heat map. When clustering by samples it is possible to choose between using the "Tree" to determine the sample ordering, and showing the "Samples" in the order they were input to the tool. When clustering by features, only the "Samples" input order is available.

Note that if you wish to use the same settings next time you open a heat map, you need to save the settings of the **Side Panel** (see section 4.6).

34.4.2 K-means/medoids clustering

In a k-means or medoids clustering, features are clustered into k separate clusters. The procedures seek to find an assignment of features to clusters, for which the distances between features within the cluster is small, while distances between clusters are large.

Tools | Microarray Analysis (🇺🇸) | Feature Clustering (🇩🇪) | K-means/medoids Clustering (🇩🇪)

Select at least two samples () or () or an experiment ().

Note! If your data contains many features, the clustering will take very long time and could make your computer unresponsive. It is recommended to perform this analysis on a subset of the data (which also makes it easier to make sense of the clustering). See how to create a sub-experiment in section 34.1.2.

Clicking **Next** will display a dialog as shown in figure 34.44.

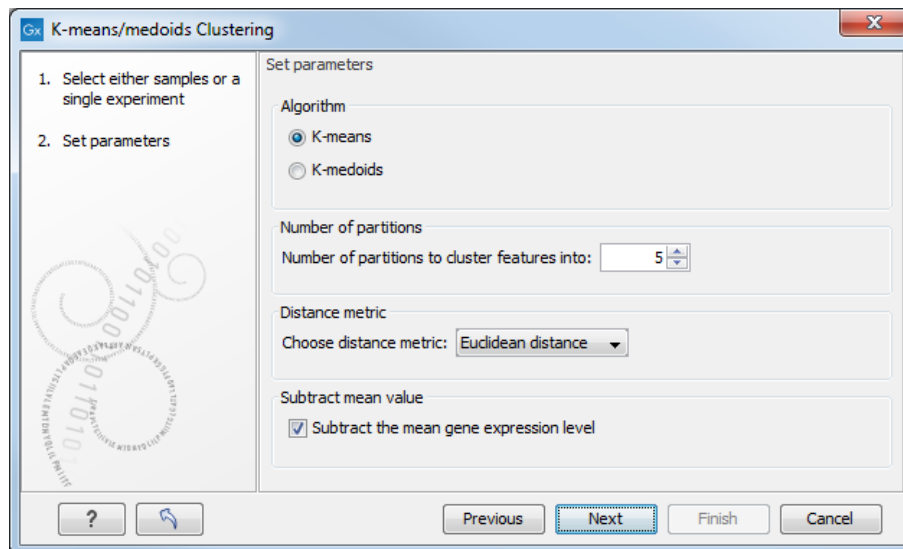


Figure 34.44: Parameters for k-means/medoids clustering.

The parameters are:

- **Algorithm.** You can choose between two clustering methods:

- **K-means.** K-means clustering assigns each point to the cluster whose center is nearest. The center/centroid of a cluster is defined as the average of all points in the cluster. If a data set has three dimensions and the cluster has two points $X = (x_1, x_2, x_3)$ and $Y = (y_1, y_2, y_3)$, then the centroid Z becomes $Z = (z_1, z_2, z_3)$, where $z_i = (x_i + y_i)/2$ for $i = 1, 2, 3$. The algorithm attempts to minimize the intra-cluster variance defined by:

$$V = \sum_{i=1}^k \sum_{x_j \in S_i} (x_j - \mu_i)^2$$

where there are k clusters $S_i, i = 1, 2, \dots, k$ and μ_i is the centroid of all points $x_j \in S_i$. The detailed algorithm can be found in [Lloyd, 1982].

- **K-medoids.** K-medoids clustering is computed using the PAM-algorithm (PAM is short for Partitioning Around Medoids). It chooses datapoints as centers in contrast to the K-means algorithm. The PAM-algorithm is based on the search for k representatives (called medoids) among all elements of the dataset. When having found k representatives k clusters are now generated by assigning each element to its nearest medoid. The algorithm first looks for a good initial set of medoids (the BUILD phase). Then it

finds a local minimum for the objective function:

$$V = \sum_{i=1}^k \sum_{x_j \in S_i} (x_j - c_i)^2$$

where there are k clusters $S_i, i = 1, 2, \dots, k$ and c_i is the medoid of S_i . This solution implies that there is no single switch of an object with a medoid that will decrease the objective (this is called the SWAP phase). The PAM-algorithm is described in [Kaufman and Rousseeuw, 1990].

- **Number of partitions.** The maximum number of partitions to cluster features into: the final number of clusters can be smaller than that.
- **Distance metric.** The metric to compute distance between data points.
 - **Euclidean distance.** The ordinary distance between two elements - the length of the segment connecting them. If $u = (u_1, u_2, \dots, u_n)$ and $v = (v_1, v_2, \dots, v_n)$, then the Euclidean distance between u and v is

$$|u - v| = \sqrt{\sum_{i=1}^n (u_i - v_i)^2}.$$

- **Manhattan distance.** The Manhattan distance between two elements is the distance measured along axes at right angles. If $u = (u_1, u_2, \dots, u_n)$ and $v = (v_1, v_2, \dots, v_n)$, then the Manhattan distance between u and v is

$$|u - v| = \sum_{i=1}^n |u_i - v_i|.$$

- **Subtract mean value.** For each gene, subtract the mean gene expression value over all input samples.

Clicking **Next** will display a dialog as shown in figure 34.45.

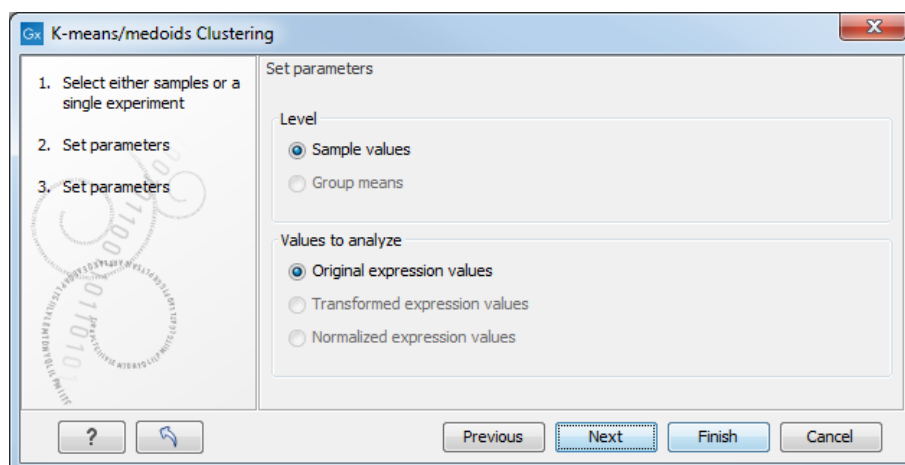


Figure 34.45: Parameters for *k*-means/medoids clustering.

At the top, you can choose the **Level** to use. Choosing 'sample values' means that distances will be calculated using all the individual values of the samples. When 'group means' are chosen, distances are calculated using the group means.

At the bottom, you can select which values to cluster (see section 34.2.1).

Click **Finish** to start the tool.

The k-means implementation first assigns each feature to a cluster at random. Then, at each iteration, it reassigns features to the centroid of the nearest cluster. During this reassignment, it can happen that one or more of the clusters becomes empty, explaining why the final number of clusters might be smaller than the one specified in "number of partitions". Note that the initial assignment of features to clusters is random, so results can differ when the algorithm is run again.

Viewing the result of k-means/medoids clustering

The result of the clustering is a number of graphs. The number depends on the number of partitions chosen (figure 34.44) - there is one graph per cluster. Using drag and drop as explained in section 2.1.4, you can arrange the views to see more than one graph at the time.

Figure 34.46 shows an example where four clusters have been arranged side-by-side.

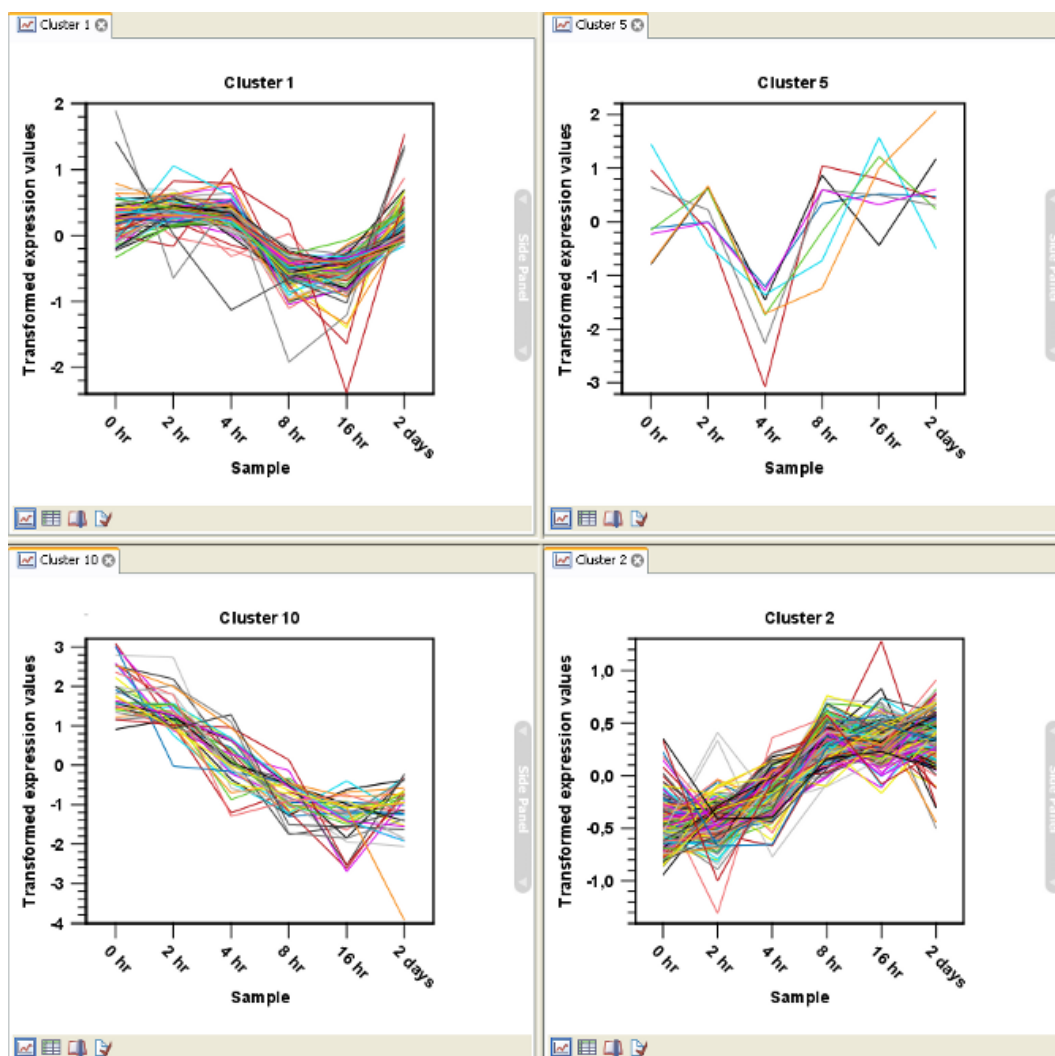


Figure 34.46: Four clusters created by k-means/medoids clustering.

The samples used are from a time-series experiment, and you can see that the expression levels for each cluster have a distinct pattern. The two clusters at the bottom have falling and rising expression levels, respectively, and the two clusters at the top both fall at the beginning but then rise again (the one to the right starts to rise earlier than the other one).

Having inspected the graphs, you may wish to take a closer look at the features represented in each cluster. In the experiment table, the clustering has added an extra column with the name of the cluster that the feature belongs to. In this way you can filter the table to see only features from a specific cluster. This also means that you can select the feature of this cluster in a volcano or scatter plot as described in section 34.1.5.

34.5 Statistical analysis - identifying differential expression


The *CLC Genomics Workbench* is designed to help you identify differential expression.

You have a choice of a number of standard statistical tests, that are suitable for different data types and different types of experimental settings. There are two main types of tests: tests that assume that data consists of counts and compare these or their proportions (described in section 34.5.1) and tests that assume that the data is real-valued, has Gaussian distributions and compare means (described in section 34.5.2).

To run the statistical analysis:

Microarray Analysis  | **Statistical Analysis**  | **Proportion-based Statistical Analysis** 

or **Microarray Analysis**  | **Statistical Analysis**  | **Gaussian Statistical Analysis** 

For all kinds of statistical analyses, you first select the experiment  that you wish to use and click **Next** (learn more about setting up experiments in section 34.1.1).

The first part of the explanation of how to proceed and perform the statistical analysis is divided into three, depending on whether you are doing tests on proportions or Gaussian-based tests. The last part has an explanation of the options regarding corrected p-values which applies to all tests.

34.5.1 Tests on proportions

The proportions-based tests are applicable in situations where your data samples consists of counts of a number of 'types' of data. This could e.g. be in a study where gene expression levels are measured by tag profiling for example. Here the different 'types' could correspond to the different 'genes' in a reference genome, and the counts could be the numbers of reads matching each of these genes. The tests compare counts by considering the proportions that they make up the total sum of counts in each sample. By comparing the expression levels at the level of proportions rather than raw counts, the data is corrected for sample size.

There are two tests available for comparing proportions: the test of [Kal et al., 1999] and the test of [Baggerly et al., 2003]. Both tests compare pairs of groups. If you have a multi-group experiment (see section 34.1.1), you may choose either to have tests produced for all pairs of groups (by clicking the 'All pairs' button) or to have a test produced for each group compared to a specified reference group (by clicking the 'Against reference' button). In the last case you must

specify which of the groups you want to use as reference (the default is to use the group you specified as Group 1 when you set up the experiment).

Note that the proportion-based tests use the total sample counts (that is, the sum over all expression values). If one (or more) of the counts are NaN, the sum will be NaN and all the test statistics will be NaN. As a consequence all p-values will also be NaN. You can avoid this by filtering your experiment and creating a new experiment so that no NaN values are present, before you apply the tests.

Kal et al.'s test (Z-test)

Kal et al.'s test [Kal et al., 1999] compares a single sample against another single sample, and thus requires that each group in your experiment has only one sample. The test relies on an approximation of the binomial distribution by the normal distribution [Kal et al., 1999]. Considering proportions rather than raw counts the test is also suitable in situations where the sum of counts is different between the samples.

When Kal's test is run on an experiment four columns will be added to the experiment table for each pair of groups that are analyzed. The 'Proportions difference' column contains the difference between the proportion in group 2 and the proportion in group 1. The 'Fold Change' column tells you how many times bigger the proportion in group 2 is relative to that of group 1. If the proportion in group 2 is bigger than that in group 1 this value is the proportion in group 2 divided by that in group 1. If the proportion in group 2 is smaller than that in group 1 the fold change is the proportion in group 1 divided by that in group 2 with a negative sign. The 'Test statistic' column holds that value of the test statistic, and the 'P-value' holds the two-sided p-value for the test. Up to two more columns may be added if the options to calculate Bonferroni and FDR corrected p-values were chosen.

Baggerley et al.'s test (Beta-binomial)

Baggerley et al.'s test [Baggerly et al., 2003] compares the proportions of counts in a group of samples against those of another group of samples, and is suited to cases where replicates are available in the groups. The samples are given different weights depending on their sizes (total counts). The weights are obtained by assuming a Beta distribution on the proportions in a group, and estimating these, along with the proportion of a binomial distribution, by the method of moments. The result is a weighted t-type test statistic.

When Baggerley's test is run on an experiment four columns will be added to the experiment table for each pair of groups that are analyzed. The 'Weighted proportions difference' column contains the difference between the mean of the weighted proportions across the samples assigned to group 2 and the mean of the weighted proportions across the samples assigned to group 1. The 'Weighted proportions fold change' column tells you how many times bigger the mean of the weighted proportions in group 2 is relative to that of group 1. If the mean of the weighted proportions in group 2 is bigger than that in group 1 this value is the mean of the weighted proportions in group 2 divided by that in group 1. If the mean of the weighted proportions in group 2 is smaller than that in group 1 the fold change is the mean of the weighted proportions in group 1 divided by that in group 2 with a negative sign. The 'Test statistic' column holds that value of the test statistic, and the 'P-value' holds the two-sided p-value for the test. Up to two more columns may be added if the options to calculate Bonferroni and FDR corrected p-values were chosen.

34.5.2 Gaussian-based tests

The tests based on the Gaussian distribution essentially compare the mean expression level in the experimental groups in the study, and evaluates the significance of the difference relative to the variance (or 'spread') of the data within the groups. The details of the formula used for calculating the test statistics vary according to the experimental setup and the assumptions you make about the data (read more about this in the sections on t-test and ANOVA below). The explanation of how to proceed is divided into two, depending on how many groups there are in your experiment. First comes the explanation for t-tests which is the only analysis available for two-group experimental setups (t-tests can also be used for pairwise comparison of groups in multi-group experiments). Next comes an explanation of the ANOVA test which can be used for multi-group experiments.

Note that the test statistics for the t-test and ANOVA analysis use the estimated group variances in their denominators. If all expression values in a group are identical the estimated variance for that group will be zero. If the estimated variances for both (or all) groups are zero the denominator of the test statistic will be zero. The numerator's value depends on the difference of the group means. If this is zero, the numerator is zero and the test statistic will be 0/0 which is NaN. If the numerator is different from zero the test statistic will be + or - infinity, depending on which group mean is bigger. If all values in all groups are identical the test statistic is set to zero.

T-tests

For experiments with two groups you can, among the Gaussian tests, only choose a **T-test** as shown in figure 34.47.

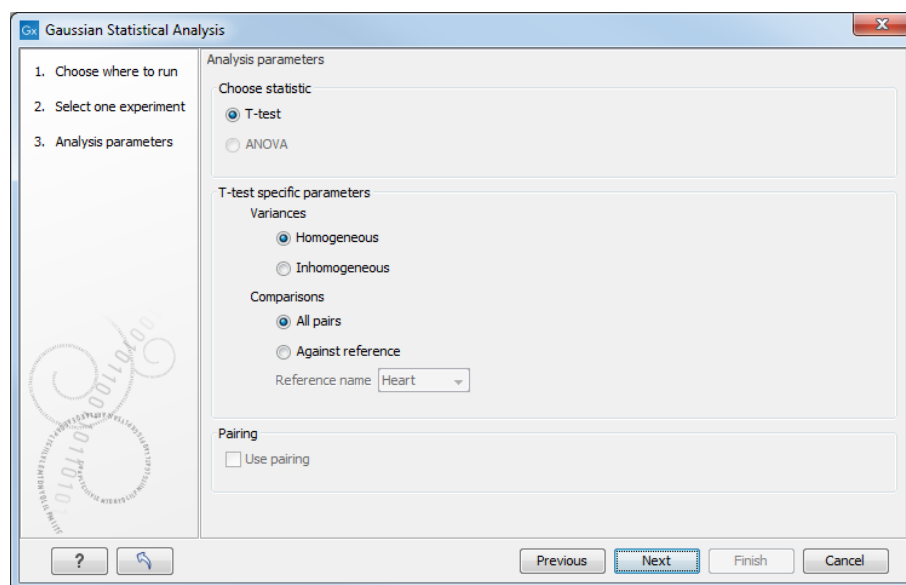


Figure 34.47: Selecting a t-test.

There are different types of t-tests, depending on the assumption you make about the variances in the groups. By selecting 'Homogeneous' (the default) calculations are done assuming that the groups have equal variances. When 'In-homogeneous' is selected, this assumption is not made.

The t-test can also be chosen if you have a multi-group experiment. In this case you may choose

either to have t-tests produced for all pairs of groups (by clicking the 'All pairs' button) or to have a t-test produced for each group compared to a specified reference group (by clicking the 'Against reference' button). In the last case you must specify which of the groups you want to use as reference (the default is to use the group you specified as Group 1 when you set up the experiment).

If a experiment with pairing was set up (see section 34.1.1) the **Use pairing** tick box is active. If ticked, paired t-tests will be calculated, if not, the formula for the standard t-test will be used.

When a t-test is run on an experiment four columns will be added to the experiment table for each pair of groups that are analyzed. The 'Difference' column contains the difference between the mean of the expression values across the samples assigned to group 2 and the mean of the expression values across the samples assigned to group 1. The 'Fold Change' column tells you how many times bigger the mean expression value in group 2 is relative to that of group 1. If the mean expression value in group 2 is bigger than that in group 1 this value is the mean expression value in group 2 divided by that in group 1. If the mean expression value in group 2 is smaller than that in group 1 the fold change is the mean expression value in group 1 divided by that in group 2 with a negative sign. The 'Test statistic' column holds that value of the test statistic, and the 'P-value' holds the two-sided p-value for the test. Up to two more columns may be added if the options to calculate Bonferroni and FDR corrected p-values were chosen.

ANOVA

For experiments with more than two groups you can choose **T-test**, see section 34.5.2, or **ANOVA**.

The ANOVA method allows analysis of an experiment with one factor and a number of groups, e.g. different types of tissues, or time points. In the analysis, the variance within groups is compared to the variance between groups. You get a significant result (that is, a small ANOVA p-value) if the difference you see between groups relative to that within groups, is larger than what you would expect, if the data were really drawn from groups with equal means.

If an experiment with pairing was set up (see section 34.1.1) the **Use pairing** tick box is active. If ticked, a repeated measures one-way ANOVA test will be calculated, if not, the formula for the standard one-way ANOVA will be used.

When an ANOVA analysis is run on an experiment four columns will be added to the experiment table for each pair of groups that are analyzed. The 'Max difference' column contains the difference between the maximum and minimum of the mean expression values of the groups, multiplied by -1 if the group with the maximum mean expression value occurs before the group with the minimum mean expression value (with the ordering: group 1, group 2, ...). The 'Max fold change' column contains the ratio of the maximum of the mean expression values of the groups to the minimum of the mean expression values of the groups, multiplied by -1 if the group with the maximum mean expression value occurs before the group with the minimum mean expression value (with the ordering: group 1, group 2, ...). The 'Test statistic' column holds the value of the test statistic, and the 'P-value' holds the two-sided p-value for the test. Up to two more columns may be added if the options to calculate Bonferroni and FDR corrected p-values were chosen.

34.5.3 Corrected p-values

Clicking **Next** will display a dialog as shown in figure 34.48.

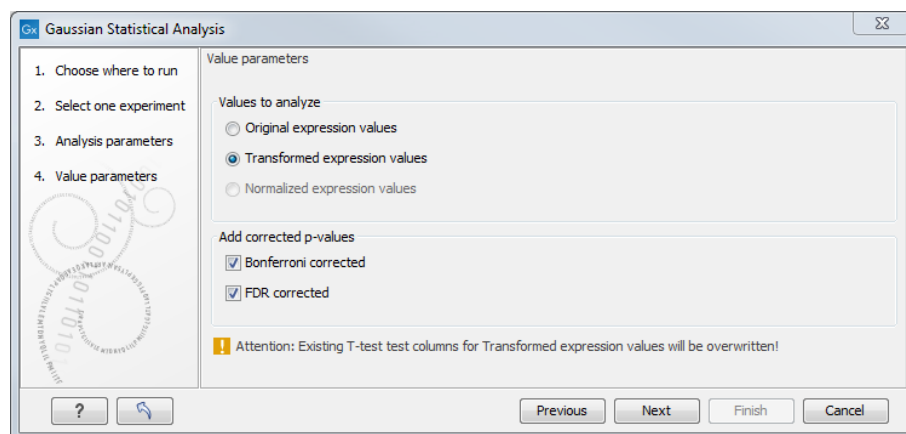


Figure 34.48: Additional settings for the statistical analysis.

At the top, you can select which values to analyze (see section 34.2.1).

Below you can select to add two kinds of corrected p-values to the analysis (in addition to the standard p-value produced for the test statistic):

- **Bonferroni corrected.**
- **FDR corrected.**

Both are calculated from the original p-values, and aim in different ways to take into account the issue of multiple testing [Dudoit et al., 2003]. The problem of multiple testing arises because the original p-values are related to a single test: the p-value is the probability of observing a more extreme value than that observed in the test carried out. If the p-value is 0.04, we would expect an as extreme value as that observed in 4 out of 100 tests carried out among groups with no difference in means. Popularly speaking, if we carry out 10000 tests and select the features with original p-values below 0.05, we will expect about 0.05 times 10000 = 500 to be false positives.

The Bonferroni corrected p-values handle the multiple testing problem by controlling the 'family-wise error rate': the probability of making at least one false positive call. They are calculated by multiplying the original p-values by the number of tests performed. The probability of having at least one false positive among the set of features with Bonferroni corrected p-values below 0.05, is less than 5%. The Bonferroni correction is conservative: there may be many genes that are differentially expressed among the genes with Bonferroni corrected p-values above 0.05, that will be missed if this correction is applied.

Instead of controlling the family-wise error rate we can control the false discovery rate: FDR. The false discovery rate is the proportion of false positives among all those declared positive. We expect 5 % of the features with FDR corrected p-values below 0.05 to be false positive. There are many methods for controlling the FDR - the method used in *CLC Genomics Workbench* is that of [Benjamini and Hochberg, 1995].

Click **Finish** to start the tool.

Note that if you have already performed statistical analysis on the same values, the existing one will be overwritten.

34.5.4 Volcano plots - inspecting the result of the statistical analysis

The results of the statistical analysis are added to the experiment and can be shown in the experiment table, see section 34.1.2. Typically columns containing the differences (or weighted differences) of the mean group values and the fold changes (or weighted fold changes) of the mean group values will be added along with a column of p-values. Also, columns with FDR or Bonferroni corrected p-values will be added if these were calculated. This added information allows features to be sorted and filtered to exclude the ones without sufficient proof of differential expression (learn more in section 9).

If you want a more visual approach to the results of the statistical analysis, you can click the **Show Volcano Plot** (🔍) button at the bottom of the experiment table view. In the same way as the scatter plot (see section 34.1.4 for details), the volcano plot is yet another view on the experiment. Because it uses the p-values and mean differences produced by the statistical analysis, the plot is only available once a statistical analysis has been performed on the experiment.

An example of a volcano plot is shown in figure 34.49.

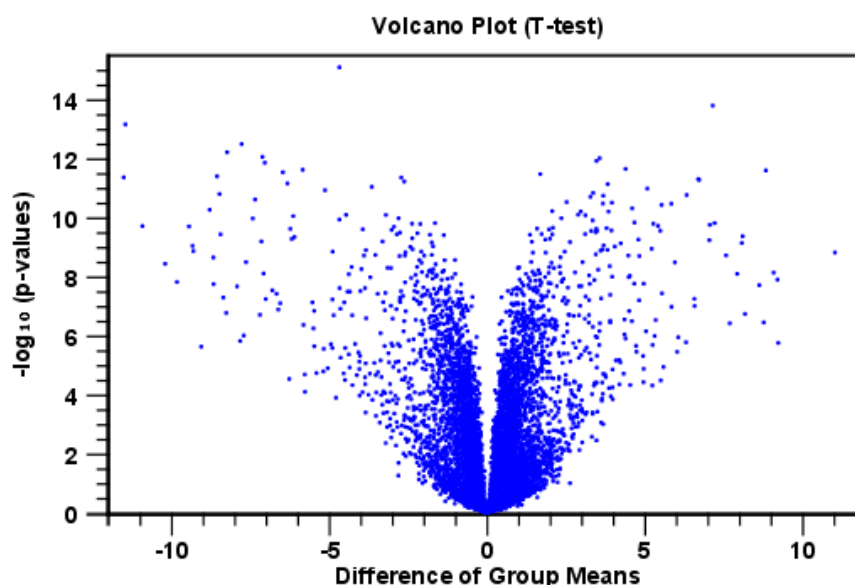


Figure 34.49: *Volcano plot.*

The volcano plot shows the relationship between the p-values of a statistical test and the magnitude of the difference in expression values of the samples in the groups. On the y-axis the $-\log_{10}$ p-values are plotted. For the x-axis you may choose between two sets of values by choosing either 'Fold change' or 'Difference' in the volcano plot side panel's 'Values' part. If you choose 'Fold change' the log of the values in the 'fold change' (or 'Weighted fold change') column for the test will be displayed. If you choose 'Difference' the values in the 'Difference' (or 'Weighted difference') column will be used. Which values you wish to display will depend upon the scale of your data (Read the note on fold change in section 34.1.2).

The larger the difference in expression of a feature, the more extreme its point will lie on

the X-axis. The more significant the difference, the smaller the p-value and thus the higher the $-\log_{10}(p)$ value. Thus, points for features with highly significant differences will lie high in the plot. Features of interest are typically those which change significantly and by a certain magnitude. These are the points in the upper left and upper right hand parts of the volcano plot.

If you have performed different tests or you have an experiment with multiple groups you need to specify for which test and which group comparison you want the volcano plot to be shown. You do this in the 'Test' and 'Values' parts of the volcano plot side panel.

Options for the volcano plot are described in further detail when describing the **Side Panel** below.

If you place your mouse on one of the dots, a small text box will tell the name of the feature. Note that you can zoom in and out on the plot (see section 2.2).

In the **Side Panel** to the right, there is a number of options to adjust the view of the volcano plot. Under **Graph preferences**, you can adjust the general properties of the volcano plot

- **Lock axes** This will always show the axes even though the plot is zoomed to a detailed level.
- **Frame** Shows a frame around the graph.
- **Show legends** Shows the data legends.
- **Tick type** Determine whether tick lines should be shown outside or inside the frame.
- **Tick lines at** Choosing Major ticks will show a grid behind the graph.
- **Horizontal axis range** Sets the range of the horizontal axis (x axis). Enter a value in **Min** and **Max**, and press Enter. This will update the view. If you wait a few seconds without pressing Enter, the view will also be updated.
- **Vertical axis range** Sets the range of the vertical axis (y axis). Enter a value in **Min** and **Max**, and press Enter. This will update the view. If you wait a few seconds without pressing Enter, the view will also be updated.

Below the general preferences, you find the **Dot properties**, where you can adjust coloring and appearance of the dots.

- **Dot type** Can be None, Cross, Plus, Square, Diamond, Circle, Triangle, Reverse triangle, or Dot.
- **Dot color** Click the color box to select a color.

At the very bottom, you find two groups for choosing which values to display:

- **Test.** In this group, you can select which kind of test you want the volcano plot to be shown for.
- **Values.** Under **Values**, you can select which values to plot. If you have multi-group experiments, you can select which groups to compare. You can also select whether to plot **Difference** or **Fold change** on the x-axis. Read the note on fold change in section 34.1.2.

Note that if you wish to use the same settings next time you open a box plot, you need to save the settings of the **Side Panel** (see section 4.6).

34.6 Annotation tests

The annotation tests are tools for detecting significant patterns among features (e.g. genes) of experiments, based on their annotations. This may help in interpreting the analysis of the large numbers of features in an experiment in a biological context. Which biological context, depends on which annotation you choose to examine, and could e.g. be biological process, molecular function or pathway as specified by the Gene Ontology or KEGG. The annotation testing tools of course require that the features in the experiment you want to analyze are annotated. Learn how to annotate an experiment in section 34.1.3.

34.6.1 Hypergeometric Tests on Annotations

The first approach to using annotations to extract biological information is the hypergeometric annotation test. This test measures the extent to which the annotation categories of features in a smaller gene list, 'A', are over or under-represented relative to those of the features in larger gene list 'B', of which 'A' is a sub-list. Gene list B is often the features of the full experiment, possibly with features which are thought to represent only noise, filtered away. Gene list A is a sub-experiment of the full experiment where most features have been filtered away and only those that seem of interest are kept. Typically gene list A will consist of a list of candidate differentially expressed genes. This could be the gene list obtained after carrying out a statistical analysis on the experiment, and choosing to keep only those features with FDR corrected p-values < 0.05 and a fold change larger than 2 in absolute value. The hyper geometric test procedure implemented is similar to the unconditional GStats test of [Falcon and Gentleman, 2007].

Tools | Microarray Analysis (📁) | Annotation Test (📁) | Hypergeometric Tests on Annotations (🌐)

This will show a dialog where you can select the two experiments - the larger experiment, e.g. the original experiment including the full list of features - and a sub-experiment (see how to create a sub-experiment in section 34.1.2).

Click **Next**. This will display the dialog shown in figure 34.50.

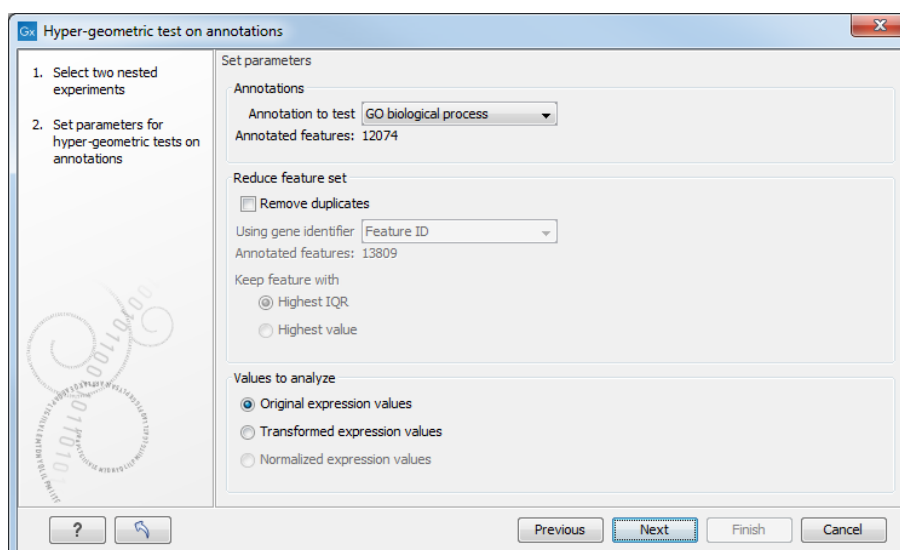


Figure 34.50: Parameters for performing a hypergeometric test on annotations.

At the top, you select which annotation to use for testing. You can select from all the annotations available on the experiment, but it is of course only a few that are biologically relevant. Once you have selected an annotation, you will see the number of features carrying this annotation below.

Annotations are typically given at the gene level. Often a gene is represented by more than one feature in an experiment. If this is not taken into account it may lead to a biased result. The standard way to deal with this is to reduce the set of features considered, so that each gene is represented only once. In the next step, **Remove duplicates**, you can choose the basis on which the feature set will be reduced:

- **Using gene identifier.**
- **Keep feature with:**
 - **Highest IQR.** The feature with the highest interquartile range (IQR) is kept.
 - **Highest value.** The feature with the highest expression value is kept.

First you specify which annotation you want to use as gene identifier. Once you have selected this, you will see the number of features carrying this annotation below. Next you specify which feature you want to keep for each gene. This may be either the feature with the highest inter-quartile range or the highest value.

At the bottom, you can select which values to analyze (see section 34.2.1). Only features that have a numerical value assigned to them will be used for the analysis. That is, any feature which has a value of plus infinity, minus infinity or NaN will not be included in the feature list taken into the test. Thus, the choice of value at this step can affect the features that are taken forward into the test in two ways:

- If there are features with values of plus infinity, minus infinity or NaN, those features will not be taken forward into the test. This can be a consideration when choosing transformed values, where the mathematical manipulations involved may lead to such values.
- If you chose to remove duplicates, then the value type you choose here is the value used for checking the highest IQR or value to determine which feature is taken forward into the test.

Click **Finish** to start the tool.

The final number of features used for the test is reported in this history view of the test results.

Result of hypergeometric tests on annotations The result of performing hypergeometric tests on annotations using GO biological process is shown in figure 34.51.

The table shows the following information:

- **Category.** This is the identifier for the category.
- **Description.** This is the description belonging to the category. Both of these are simply extracted from the annotations.

Category	Description	Full set	In subset	Expected in subset	Observed - expected	p-value
0055114	oxidation-reduction process (MGI:MGI:483...	561	11	3	8	4.59E-4
0051496	positive regulation of stress fiber assembly...	27	3	0	3	5.25E-4
0001913	T cell mediated cytotoxicity (MGI:MGI:303...	7	2	0	2	7.10E-4
0006629	lipid metabolic process (MGI:MGI:1354194 ...	351	8	2	6	1.10E-3
0006956	complement activation (MGI:MGI:3833206)...	9	2	0	2	1.21E-3
0009058	biosynthetic process (MGI:MGI:2152098 [I...	38	3	0	3	1.45E-3
0006855	drug transmembrane transport (MGI:MGI:...	11	2	0	2	1.83E-3
0032869	cellular response to insulin stimulus (MGI:M...	45	3	0	3	2.36E-3
0008152	metabolic process (MGI:MGI:2152098 [IEA...	456	8	3	5	5.51E-3
0000105	histidine biosynthetic process (MGI:MGI:13...	1	1	0	1	5.90E-3
2001213	negative regulation of vasculogenesis (MG...	1	1	0	1	5.90E-3
0048241	epinephrine transport (MGI:MGI:4417868 [...	1	1	0	1	5.90E-3
0009115	xanthine catabolic process (MGI:MGI:4417...	1	1	0	1	5.90E-3
0050427	3'-phosphoadenosine 5'-phosphosulfate m...	1	1	0	1	5.90E-3
0033301	cell cycle comprising mitosis without cytokin...	1	1	0	1	5.90E-3
0006507	GPI anchor release (MGI:MGI:5447609)PM...	1	1	0	1	5.90E-3

Figure 34.51: The result of testing on GO biological process.

- **Full set.** The number of features in the original experiment (not the subset) with this category. (Note that this is after removal of duplicates).
- **In subset.** The number of features in the subset with this category. (Note that this is after removal of duplicates).
- **Expected in subset.** The number of features we would have expected to find with this annotation category in the subset, if the subset was a random draw from the full set.
- **Observed - expected.** 'In subset' - 'Expected in subset'
- **p-value.** The tail probability of the hyper geometric distribution This is the value used for sorting the table.

Categories with small p-values are over-represented on the features in the subset relative to the full set.

GO terms are organized in a hierarchical structure. For example, the term "GO:0033151 V(D)J recombination" from the Gene Ontology [Ashburner et al., 2000, *The Gene Ontology Consortium*, 2019] (<https://geneontology.org/>) is a descendant of "GO:0006259 DNA metabolic process".

When testing for the significance of a particular GO term, all features linked to descendant GO terms are included in the test. This can lead to a higher number of detected genes in the output table, compared to the number of genes linked to the tested GO term.

Due to the hierarchical structure, GO terms are not independent of one another, and the p-values provided in the table should be interpreted with caution.

34.6.2 Gene Set Enrichment Analysis

When carrying out a hypergeometric test on annotations you typically compare the annotations of the genes in a subset containing 'the significantly differentially expressed genes' to those of the total set of genes in the experiment. Which, and how many, genes are included in the subset is

somewhat arbitrary - using a larger or smaller p-value cut-off will result in including more or less. Also, the magnitudes of differential expression of the genes is not considered.

The Gene Set Enrichment Analysis (GSEA) does NOT take a sublist of differentially expressed genes and compare it to the full list - it takes a single gene list (a single experiment). The idea behind GSEA is to consider a measure of association between the genes and phenotype of interest (e.g. test statistic for differential expression) and rank the genes according to this measure of association. A test is then carried out for each annotation category, for whether the ranks of the genes in the category are evenly spread throughout the ranked list, or tend to occur at the top or bottom of the list.

The GSEA test implemented here is that of [Tian et al., 2005]. The test implicitly calculates and uses a standard t-test statistic for two-group experiments, and ANOVA statistic for multiple group experiments for each feature, as measures of association. For each category, the test statistics for the features in that category are summed and a category based test statistic is calculated as this sum divided by the square root of the number of features in the category. Note that if a feature has the value NaN in one of the samples, the t-test statistic for the feature will be NaN. Consequently, the combined statistic for each of the categories in which the feature is included will be NaN. Thus, it is advisable to filter out any feature that has a NaN value before applying GSEA.

The p-values for the GSEA test statistics are calculated by permutation: The original test statistics for the features are permuted and new test statistics are calculated for each category, based on the permuted feature test statistics. This is done the number of times specified by the user in the wizard. For each category, the lower and upper tail probabilities are calculated by comparing the original category test statistics to the distribution of the permutation-based test statistics for that category. The lower and higher tail probabilities are the number of these that are lower and higher, respectively, than the observed value, divided by the number of permutations.

As the p-values are based on permutations you may sometimes see results where category x's test statistic is lower than that of category y and the categories are of equal size, but where the lower tail probability of category x is higher than that of category y. This is due to imprecision in the estimations of the tail probabilities from the permutations. The higher the number of permutations, the more stable the estimation.

You may run a GSEA on a full experiment, or on a sub-experiment where you have filtered away features that you think are un-informative and represent only noise. Typically you will remove features that are constant across samples (those for which the value in the 'Range' column is zero' – these will have a t-test statistic of zero) and/or those for which the inter-quantile range is small. As the GSEA algorithm calculates and ranks genes on p-values from a test of differential expression, it will generally not make sense to filter the experiment on p-values produced in an analysis of differential expression, prior to running GSEA on it.

Tools | Microarray Analysis ( | Annotation Test ( | Gene Set Enrichment Analysis (GSEA) (

Select an experiment and click **Next**.

Click **Next**. This will display the dialog shown in figure 34.52.

At the top, you select which annotation to use for testing. You can select from all the annotations available on the experiment, but it is of course only a few that are biologically relevant. Once you have selected an annotation, you will see the number of features carrying this annotation below.

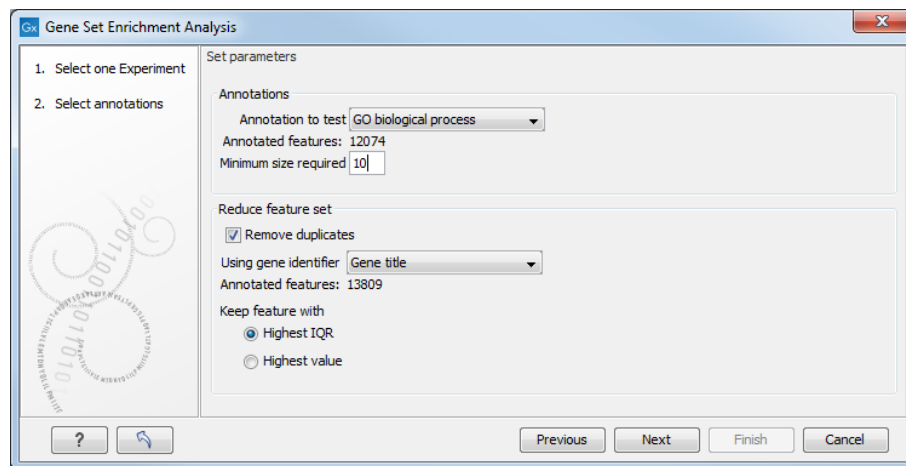


Figure 34.52: Gene set enrichment analysis on GO biological process.

In addition, you can set a filter: **Minimum size required**. Only categories with more genes (i.e. features) than the specified number will be considered. Excluding categories with small numbers of genes may lead to more robust results.

Annotations are typically given at the gene level. Often a gene is represented by more than one feature in an experiment. If this is not taken into account it may lead to a biased result. The standard way to deal with this is to reduce the set of features considered, so that each gene is represented only once. Check the **Remove duplicates** check box to reduce the feature set, and you can choose how you want this to be done:

- **Using gene identifier.**
- **Keep feature with:**
 - **Highest IQR.** The feature with the highest interquartile range (IQR) is kept.
 - **Highest value.** The feature with the highest expression value is kept.

First you specify which annotation you want to use as gene identifier. Once you have selected this, you will see the number of features carrying this annotation below. Next you specify which feature you want to keep for each gene. This may be either the feature with the highest inter-quartile range or the highest value.

Clicking **Next** will display the dialog shown in figure 34.53.

At the top, you can select which values to analyze (see section 34.2.1).

Below, you can set the **Permutations for p-value calculation**. For the GSEA test a p-value is calculated by permutation: p permuted data sets are generated, each consisting of the original features, but with the test statistics permuted. The GSEA test is run on each of the permuted data sets. The test statistic is calculated on the original data, and the resulting value is compared to the distribution of the values obtained for the permuted data sets. The permutation based p-value is the number of permutation based test statistics above (or below) the value of the test statistic for the original data, divided by the number of permuted data sets. For reliable permutation-based p-value calculation a large number of permutations is required (100 is the default).

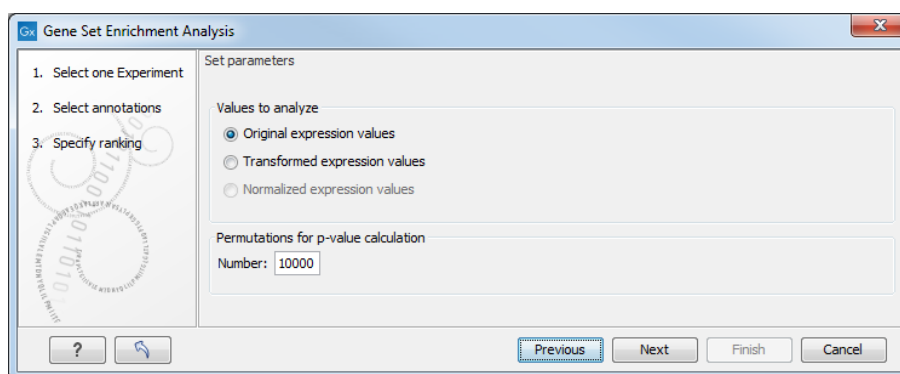


Figure 34.53: Gene set enrichment analysis parameters.

Click **Finish** to start the tool.

Result of gene set enrichment analysis The result of performing gene set enrichment analysis using GO biological process is shown in figure 34.54.

Category	Description	Size	Test statistic	Lower tail	Upper tail
0006508	proteolysis	658	-11.83	0.00	1.00
0044260	cellular macromolecule metabolic process	3323	-9.70	0.00	1.00
0044264	cellular polysaccharide metabolic process	50	-21.35	0.00	1.00
0048634	regulation of muscle organ development	102	-17.84	0.00	1.00
0048641	regulation of skeletal muscle tissue development	44	-32.11	0.00	1.00
0048642	negative regulation of skeletal muscle tissue development	10	-26.77	0.00	1.00
0014904	myotube cell development	23	-28.90	0.00	1.00
0045661	regulation of myoblast differentiation	39	-20.11	0.00	1.00
0005977	glycogen metabolic process	44	-24.15	0.00	1.00
0014888	striated muscle adaptation	22	-25.33	0.00	1.00
0048741	skeletal muscle fiber development	21	-30.18	0.00	1.00
0014819	regulation of skeletal muscle contraction	10	-34.57	0.00	1.00
0043170	macromolecule metabolic process	3739	-10.33	0.00	1.00
0019538	protein metabolic process	2337	-8.90	0.00	1.00
0007519	skeletal muscle tissue development	52	-25.32	0.00	1.00
0009057	macromolecule catabolic process	450	-13.59	0.00	1.00

Figure 34.54: The result of gene set enrichment analysis on GO biological process.

The table shows the following information:

- **Category.** This is the identifier for the category.
- **Description.** This is the description belonging to the category. Both of these are simply extracted from the annotations.
- **Size.** The number of features with this category. (Note that this is after removal of duplicates).
- **Test statistic.** This is the GSEA test statistic.
- **Lower tail.** This is the mass in the permutation based p-value distribution below the value of the test statistic.
- **Upper tail.** This is the mass in the permutation based p-value distribution above the value of the test statistic.

A small lower (or upper) tail p-value for an annotation category is an indication that features in this category viewed as a whole are perturbed among the groups in the experiment considered.

GO terms are organized in a hierarchical structure. For example, the term "GO:0033151 V(D)J recombination" from the Gene Ontology [Ashburner et al., 2000, The Gene Ontology Consortium, 2019] (<https://geneontology.org/>) is a descendant of "GO:0006259 DNA metabolic process".

When testing for the significance of a particular GO term, all features linked to descendant GO terms are included in the test. This can lead to a higher number of detected genes in the output table, compared to the number of genes linked to the tested GO term.

34.7 General plots

In the **General Plots** folder, you find three general plots that may be useful at various point of your analysis work flow. The plots are explained in detail below.

34.7.1 Histogram

A histogram shows a distribution of a set of values. Histograms are often used for examining and comparing distributions, e.g. of expression values of different samples, in the quality control step of an analysis. You can create a histogram showing the distribution of expression value for a sample:

Tools | Microarray Analysis (📁) | General Plots (📁) | Create Histogram (📊)

Select a number of samples (📁), (📁), (📁) or a graph track. When you have selected more than one sample, a histogram will be created for each one. Clicking **Next** will display a dialog as shown in figure 34.55.

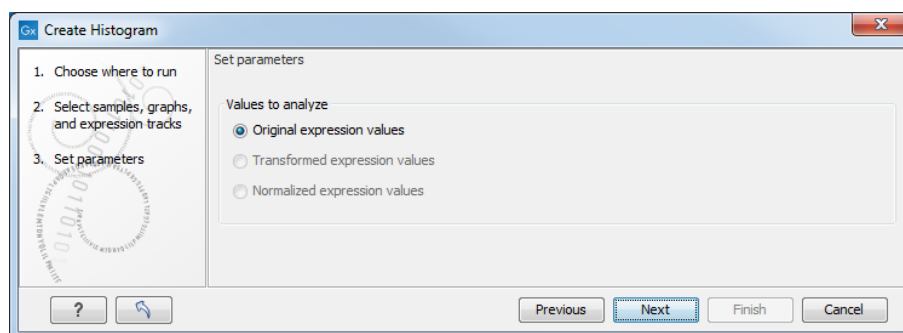


Figure 34.55: Selecting which values the histogram should be based on.

In this dialog, you select the values to be used for creating the histogram (see section 34.2.1).

Click **Finish** to start the tool.

Viewing histograms

The resulting histogram is shown in a figure 34.56

The histogram shows the expression value on the x axis (in the case of figure 34.56 the transformed expression values) and the counts of these values on the y axis.

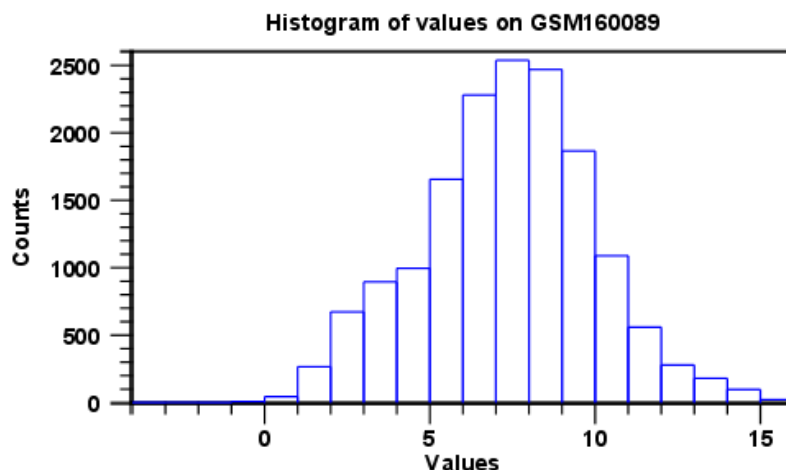


Figure 34.56: Histogram showing the distribution of transformed expression values.

In the **Side Panel** to the left, there is a number of options to adjust the view. Under **Graph preferences**, you can adjust the general properties of the plot.

- **Lock axes** This will always show the axes even though the plot is zoomed to a detailed level.
- **Frame** Shows a frame around the graph.
- **Show legends** Shows the data legends.
- **Tick type** Determine whether tick lines should be shown outside or inside the frame.
- **Tick lines at** Choosing Major ticks will show a grid behind the graph.
- **Horizontal axis range** Sets the range of the horizontal axis (x axis). Enter a value in **Min** and **Max**, and press Enter. This will update the view. If you wait a few seconds without pressing Enter, the view will also be updated.
- **Vertical axis range** Sets the range of the vertical axis (y axis). Enter a value in **Min** and **Max**, and press Enter. This will update the view. If you wait a few seconds without pressing Enter, the view will also be updated.
- **Break points.** Determines where the bars in the histogram should be:
 - **Sturges method.** This is the default. The number of bars is calculated from the range of values by Sturges formula [Sturges, 1926].
 - **Equi-distanced bars.** This will show bars from **Start** to **End** and with a width of **Sep**.
 - **Number of bars.** This will simply create a number of bars starting at the lowest value and ending at the highest value.

Below the graph preferences, you find **Line color**. Allows you to choose between many different colors. Click the color box to select a color.

Note that if you wish to use the same settings next time you open a principal component plot, you need to save the settings of the **Side Panel** (see section 4.6).

Besides the histogram view itself, the histogram can also be shown in a table, summarizing key properties of the expression values. An example is shown in figure 34.57.

Data	Count
Total number of values	15923
Number values used	15923
Number -Inf values	0
Number +Inf values	0
Number NaN values	0

Figure 34.57: Table view of a histogram.

The table lists the following properties:

- **Number +Inf values**
- **Number -Inf values**
- **Number NaN values**
- **Number values used**
- **Total number of values**

34.7.2 MA plot

The MA plot is a scatter rotated by 45° . For two samples of expression values it plots for each gene the difference in expression against the mean expression level. MA plots are often used for quality control, in particular, to assess whether normalization and/or transformation is required.

You can create an MA plot comparing two samples:

Tools | Microarray Analysis (📊) | General Plots (📊) | Create MA Plot (📊)

In the first two dialogs, select two samples ((📊), (📊) or (📊)): the first must be the case expression data, and the second the control data. Clicking **Next** will display a dialog as shown in figure 34.58.

In this dialog, you select the values to be used for creating the MA plot (see section 34.2.1).

Click **Finish** to start the tool.

Viewing MA plots

The resulting plot is shown in a figure 34.59.

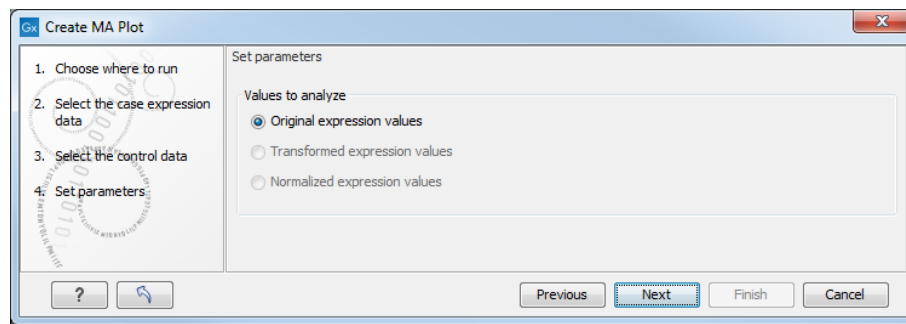


Figure 34.58: Selecting which values the MA plot should be based on.

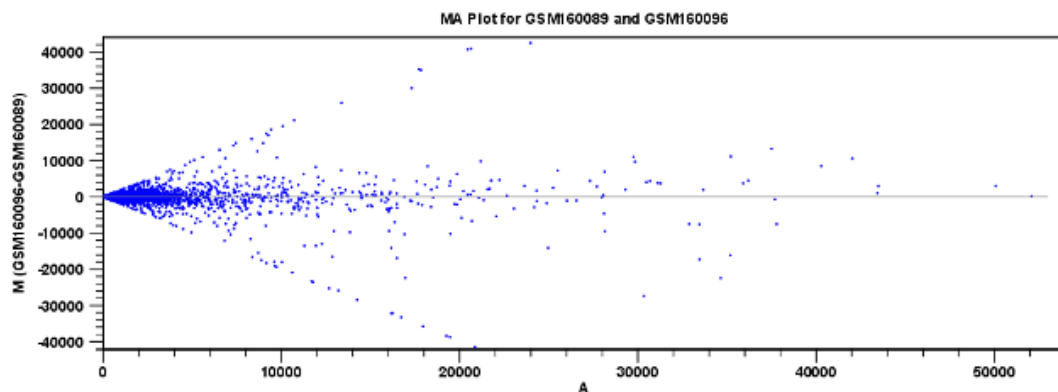


Figure 34.59: MA plot based on original expression values.

The X axis shows the mean expression level of a feature on the two samples and the Y axis shows the difference in expression levels for a feature on the two samples. From the plot shown in figure 34.59 it is clear that the variance increases with the mean. With an MA plot like this, you will often choose to transform the expression values (see section 34.2.2).

Figure 34.60 shows the same two samples where the MA plot has been created using log2 transformed values.

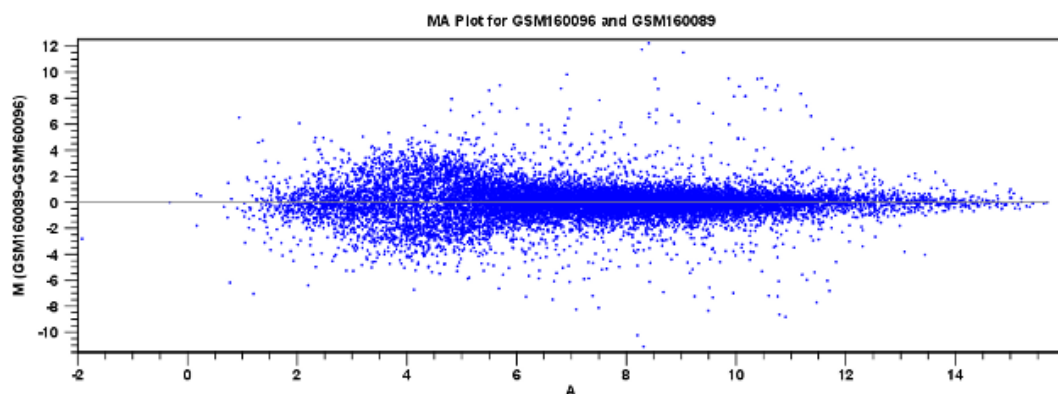


Figure 34.60: MA plot based on transformed expression values.

The much more symmetric and even spread indicates that the dependence of the variance on the mean is not as strong as it was before transformation.

In the **Side Panel** to the left, there is a number of options to adjust the view. Under **Graph preferences**, you can adjust the general properties of the plot.

- **Lock axes** This will always show the axes even though the plot is zoomed to a detailed level.
- **Frame** Shows a frame around the graph.
- **Show legends** Shows the data legends.
- **Tick type** Determine whether tick lines should be shown outside or inside the frame.
- **Tick lines at** Choosing Major ticks will show a grid behind the graph.
- **Horizontal axis range** Sets the range of the horizontal axis (x axis). Enter a value in **Min** and **Max**, and press Enter. This will update the view. If you wait a few seconds without pressing Enter, the view will also be updated.
- **Vertical axis range** Sets the range of the vertical axis (y axis). Enter a value in **Min** and **Max**, and press Enter. This will update the view. If you wait a few seconds without pressing Enter, the view will also be updated.
- **y = 0 axis.** Draws a line where $y = 0$. Below there are some options to control the appearance of the line:
 - **Line width** Thin, Medium or Wide
 - **Line type** None, Line, Long dash or Short dash
 - **Line color** Click the color box to select a color.
- **Line width** Thin, Medium or Wide
- **Line type** None, Line, Long dash or Short dash
- **Line color** Click the color box to select a color.

Below the general preferences, you find the **Dot properties** preferences, where you can adjust coloring and appearance of the dots:

- **Dot type** Can be None, Cross, Plus, Square, Diamond, Circle, Triangle, Reverse triangle, or Dot.
- **Dot color** Click the color box to select a color.

Note that if you wish to use the same settings next time you open a scatter plot, you need to save the settings of the **Side Panel** (see section 4.6).

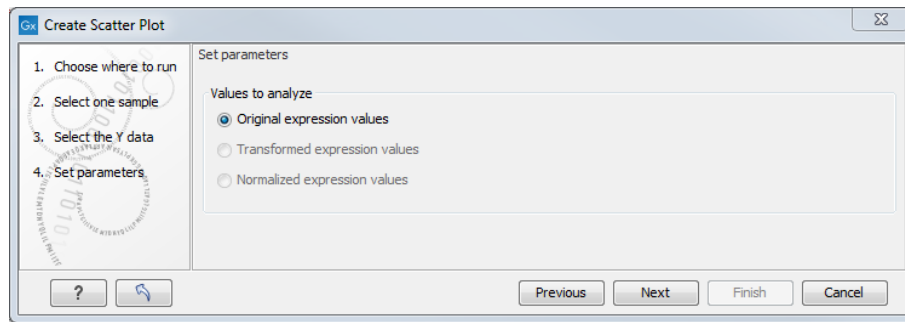


Figure 34.61: Selecting which values the scatter plot should be based on.

34.7.3 Scatter plot

As described in section 34.1.4, an experiment can be viewed as a scatter plot. However, you can also create a "stand-alone" scatter plot of two samples:

Tools | Microarray Analysis (📁) | General Plots (📁) | Create Scatter Plot (🔍)

In the first two dialogs, select two samples (📁), (📁) or (📁): the first is the sample that will be plotted on the X axis of the plot, the second the one that will define the Y axis. Clicking **Next** will display a dialog as shown in figure 34.61.

In this dialog, you select the values to be used for creating the scatter plot (see section 34.2.1).

Click **Finish** to start the tool.

For more information about the scatter plot view and how to interpret it, please see section 34.1.4.

Chapter 35

De Novo sequencing

Contents

35.1 De Novo Assembly	1117
35.1.1 The De Novo Assembly algorithm	1118
35.1.2 Best practices	1128
35.1.3 Randomness in the results	1131
35.1.4 De Novo Assembly parameters	1132
35.1.5 De Novo Assembly output	1134
35.2 Map Reads to Contigs	1138
35.3 De Novo Assemble Long Reads	1141
35.3.1 De Novo Assemble Long Reads parameters	1141
35.3.2 De Novo Assemble Long Reads output	1142
35.4 Polish Contigs with Reads	1144
35.4.1 Polish Contigs with Reads output	1145

35.1 De Novo Assembly

This section covers the tool for de novo assembly of short reads, **De Novo Assembly**. This tool supports mixing of paired reads (both insert size and orientation).

The tool runs in two stages:

1. First, simple contig sequences are created by using all the information that are in the read sequences. This is the actual *de novo* part of the process. These simple contig sequences do not contain any information about which reads the contigs are built from. This part is elaborated in section [35.1.1](#).
2. Second, all the reads are mapped using the simple contig sequence as reference. This is done in order to show coverage levels along the contigs and to enable more downstream analysis like SNP detection and creating mapping reports. Note that although a read aligns to a certain position on the contig, it does not mean that the information from this read was used for building the contig, because the mapping of the reads is a completely separate part of the algorithm.

If you wish to only perform stage 1 above and get the simple contig sequences as output, this can be chosen when starting the de novo assembly (see section 35.1.4).

Note: The De Novo Assembly tool was optimized for genomes up to the size and complexity of the human genome.

35.1.1 The De Novo Assembly algorithm

The CLC de novo assembly algorithm works by using de Bruijn graphs. This is a common approach used with de novo assembly algorithms described in [Zerbino and Birney, 2008](#), [Zerbino et al., 2009](#), [Li et al., 2010](#), [Gnerre et al., 2011](#). The basic idea is to make a table of all sub-sequences of a certain length (called words) found in the reads. The words are relatively short, e.g. about 20 for small data sets and 27 for a large data set. In the following section, we use the term word size to denote the length of a word. The word size is by default determined automatically (see explanation below).

Given a word in the table, we can look up all the potential neighboring words (in all the examples here, word of length 16 are used) as shown in figure 35.1.

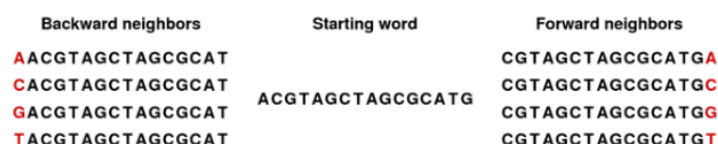


Figure 35.1: The word in the middle is 16 bases long, and it shares the 15 first bases with the backward neighboring word and the last 15 bases with the forward neighboring word.

Typically, only one of the backward neighbors and one of the forward neighbors will be present in the table. A graph can then be made where each node is a word that is present in the table and edges connect nodes that are neighbors. This is called a de Bruijn graph.

For genomic regions without repeats or sequencing errors, we get long linear stretches of connected nodes. We may choose to reduce such stretches of nodes with only one backward and one forward neighbor into nodes representing sub-sequences longer than the initial words.

Figure 35.2 shows an example where one node has two forward neighbors:



Figure 35.2: Three nodes connected, each sharing 15 bases with its neighboring node and ending with two forward neighbors.

After reduction, the three first nodes are merged, and the two sets of forward neighboring nodes are also merged as shown in figure 35.3.

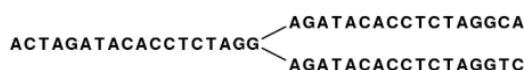


Figure 35.3: The five nodes are compacted into three. Note that the first node is now 18 bases and the second nodes are each 17 bases.

So bifurcations in the graph leads to separate nodes. In this case we get a total of three nodes after the reduction. Note that neighboring nodes still have an overlap (in this case 15 nucleotides since the word size is 16).

Given this way of representing the de Bruijn graph for the reads, we can consider some different situations:

When we have a *SNP* or a *sequencing error*, we get a so-called bubble (this is explained in detail in section 35.1.1) as shown in figure 35.4.



Figure 35.4: A bubble caused by a heterozygous SNP or a sequencing error.

Here, the central position may be either a C or a G. If this was a sequencing error occurring only once, it would be represented in the bubble as a path that is associated with a word that only occurs a single time'. On the other hand if this was a heterozygous SNP we would see both paths represented more or less equally in terms of the number of words that support each path. Thus, having information about how many times this particular word is seen in all the reads is very useful and this information is stored in the initial word table together with the words.

The most difficult problem for de novo assembly is repeats. Repeat regions in large genomes often get very complex: a repeat may be found thousands of times and part of one repeat may also be part of another repeat. Sometimes a repeat is longer than the read length (or the paired distance when pairs are available) and then it becomes impossible to resolve the length of the repeat. This is simply because there is no information available about how to connect the nodes before the repeat to the nodes after the repeat, and we just do not know how long the repeat is.

In the simple example, if we have a *repeat sequence* that is present twice in the genome, we would get a graph as shown in figure 35.5.



Figure 35.5: The central node represents the repeat region that is represented twice in the genome. The neighboring nodes represent the flanking regions of this repeat in the genome.

Note that this repeat is 57 nucleotides long (the length of the sub-sequence in the central node above plus regions into the neighboring nodes where the sequences are identical). If the repeat had been shorter than 15 nucleotides, it would not have shown up as a repeat at all since the word size is 16. This is an argument for using long words in the word table. On the other hand, the longer the word, the more words from a read are affected by a sequencing error. Also, for each increment in the word size, we get one less word from each read. This is in particular an issue for very short reads. For example, if the read length is 35, we get 16 words out of each read if the word size is 20. If the word size is 25, we get only 11 words from each read.

To strike a balance, our de novo assembler chooses a word size based on the amount of input data: the more data, the longer the word length. It is based on the following:

```
word size 12: 0 bp - 30000 bp
word size 13: 30001 bp - 90002 bp
word size 14: 90003 bp - 270008 bp
word size 15: 270009 bp - 810026 bp
word size 16: 810027 bp - 2430080 bp
word size 17: 2430081 bp - 7290242 bp
word size 18: 7290243 bp - 21870728 bp
```

```

word size 19: 21870729 bp - 65612186 bp
word size 20: 65612187 bp - 196836560 bp
word size 21: 196836561 bp - 590509682 bp
word size 22: 590509683 bp - 1771529048 bp
word size 23: 1771529049 bp - 5314587146 bp
word size 24: 5314587147 bp - 15943761440 bp
word size 25: 15943761441 bp - 47831284322 bp
word size 26: 47831284323 bp - 143493852968 bp
word size 27: 143493852969 bp - 430481558906 bp
word size 28: 430481558907 bp - 1291444676720 bp
word size 29: 1291444676721 bp - 3874334030162 bp
word size 30: 3874334030163 bp - 11623002090488 bp
etc.

```

This pattern (multiplying by 3) continues until word size of 64 which is the max. See how to adjust the word size in section [35.1.4](#)

Resolve repeats using reads

Having build the de Bruijn graph using words, our de novo assembler removes repeats and errors using reads. This is done in the following order:

- Remove weak edges
- Remove dead ends
- Resolve repeats using reads without conflicts
- Resolve repeats with conflicts
- Remove weak edges
- Remove dead ends

Each phase will be explained in the following.

Remove weak edges The de Bruijn graph is expected to contain artifacts from errors in the data. The number of reads agreeing upon an error is likely to be low especially compared to the number of reads without errors for the same region. When this relative difference is large enough, it's possible to conclude something is an error.

In the remove weak edges phase we consider each node and calculate the number c_1 of edges connected to the node and the number of times k_1 a read is passing through these edges. An average of reads going through an edge is calculated $avg_1 = k_1/c_1$ and then the process is repeated using only those edges which have more than or equal avg_1 reads going through it. Let c_2 be the number of edges which meet this requirement and k_2 the number of reads passing through these edges. A second average $avg_2 = k_2/c_2$ is used to calculate a limit,

$$limit = \frac{\log(avg_2)}{2} + \frac{avg_2}{40}$$

and each edge connected to the node which has less than or equal *limit* number of reads passing through it will be removed in this phase.

Remove dead ends Some read errors might occur more often than expected, either by chance or because they are systematic sequencing errors. These are not removed by the "Remove weak edges" phase and will cause "dead ends" to occur in the graph, which are short paths in the graph that terminate after a few nodes. Furthermore, the "Remove weak edges" sometimes only removes a part of the graph, which will also leave dead ends behind. Dead ends are identified by searching for paths in the graph where there exists an alternative path containing four times more nucleotides. All nodes in such paths are then removed in this step.

Resolve repeats without conflicts Repeats and other shared regions between the reads lead to ambiguities in the graph. These must be resolved otherwise the region will be output as multiple contigs, one for each node in the region.

The algorithm for resolving repeats without conflicts considers a number of nodes called the *window*. To start with, a window only contains one node, say *R*. We also define the *border nodes* as the nodes outside the window connected to a node in the window. The idea is to divide the border nodes into sets such that border nodes *A* and *C* are in the same set if there is a read going through *A*, through nodes in the window and then through *C*. If there are strictly more than one of these sets we can resolve the repeat area, otherwise we expand the window.

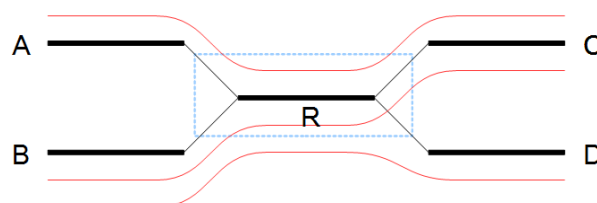


Figure 35.6: A set of nodes.

In the example in figure 35.6 all border nodes *A*, *B*, *C* and *D* are in the same set since one can reach every border nodes using reads (shown as red lines). Therefore we expand the window and in this case add node *C* to the window as shown in figure 35.7.

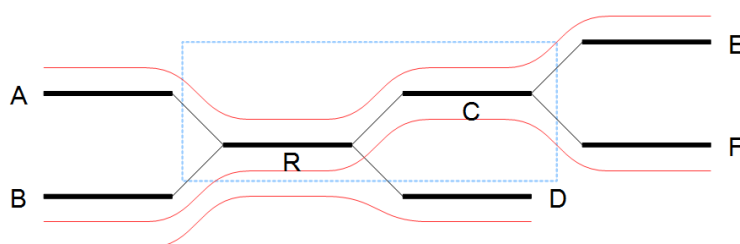


Figure 35.7: Expanding the window to include more nodes.

After the expansion of the window, the border nodes will be grouped into two groups being set *A*, *E* and set *B*, *D*, *F*. Since we have strictly more than one set, the repeat is resolved by copying the nodes and edges used by the reads which created the set. In the example the resolved repeat is shown in figure 35.8.

The algorithm for resolving repeats without conflict can be described the following way:

1. A node is selected as the window
2. The border is divided into sets using reads going through the window. If we have multiple sets, the repeat is resolved.

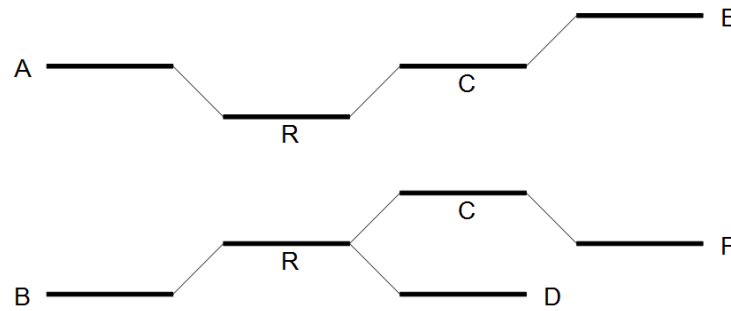


Figure 35.8: Resolving the repeat.

3. If the repeat cannot be resolved, we expand the window with nodes if possible and go to step 2.

The above steps are performed for every node.

Resolve repeats with conflicts In the previous section repeats were resolved without excluding any reads that goes through the window. While this lead to a simpler graph, the graph will still contain artifacts, which have to be removed. The next phase removes most of these errors and is similar to the previous phase:

1. A node is selected as the initial window
2. The border is divided into sets using reads going through the window. If we have multiple sets, the repeat is resolved.
3. If the repeat cannot be resolved, the border nodes are divided into sets using reads going through the window where reads containing errors are excluded. If we have multiple sets, the repeat is resolved.
4. The window is expanded with nodes if possible and step 2 is repeated.

The algorithm described above is similar to the algorithm used in the previous section, except step 3 where the reads with errors are excluded. This is done by calculating an average $avg_1 = m_1/c_1$ where m_1 is the number of reads going through the window and c_1 is the number of distinct pairs of border nodes having one (or more) of these reads connecting them. A second average $avg_2 = m_2/c_2$ is calculated where m_2 is the number of reads going through the window having at least avg_1 or more reads connecting their border nodes and c_2 the number of distinct pairs of border nodes having avg_1 or more reads connecting them. Then, a read between two border nodes B and C is excluded if the number of reads going through B and C is less than or equal to $limit$ given by

$$limit = \frac{\log(avg_2)}{2} + \frac{avg_2}{16}$$

An example where we resolve a repeat with conflicts is given in 35.9 where we have a total of 21 reads going through the window with $avg_1 = 21/3 = 7$, $avg_2 = 20/2 = 10$ and $limit = 1/2 + 10/16 = 1.125$. Therefore all reads between border nodes B and C are excluded resulting in two sets of border nodes A, C and B, D . The resolved repeat is shown in figure 35.10.

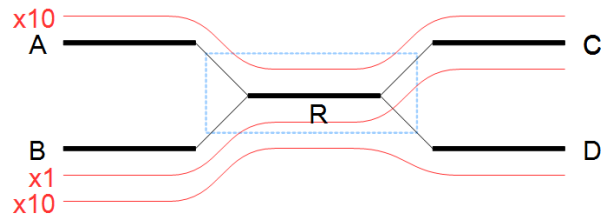


Figure 35.9: A repeat with conflicts.

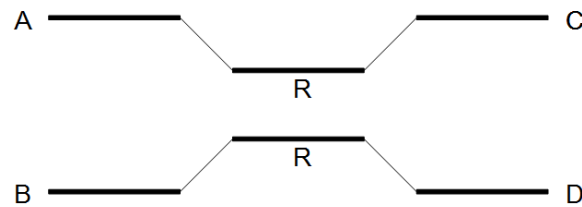


Figure 35.10: Resolving a repeat with conflicts.

Automatic paired distance estimation

The default behavior of the de novo assembler is to use the paired distances provided by the user. If the automatic paired distance estimation is enabled, the assembler will attempt to estimate the distance between paired reads. This is done by analysing the mapping of paired reads to the long unambiguous paths in the graph which are created in the read optimization step described above. The distance estimation algorithm creates a histogram (H) of the paired distances between reads in each set of paired reads (see figure 35.11). Each of these histograms are then used to estimate paired distances as described in the following.

We denote the average number of observations in the histogram $H_{avg} = \frac{1}{|H|} \sum_d H(d)$ where $H(d)$ is the number of observations (reads) with distance d and $|H|$ is the number of bins in H . The gradient of H at distance d is denoted $H'(d)$. The following algorithm is then used to compute a distance interval for each histogram.

- Identify peaks in H as $\max_{i \leq d \leq j} H(d)$ where $[i, j]$ is any interval in H where $\{H(d) \geq \frac{H_{avg}}{2} | i \leq d \leq j\}$.
- For the two largest peaks found, expand the respective intervals $[i, j]$ to $[k, l]$ where $H'(k) < 0.001 \wedge k \leq i \wedge H'(l) > -0.001 \wedge j \leq l$. I.e. we search for a point in both directions where the number of observations becomes stable. A window of size 5 is used to calculate H' in this step.
- Compute the total number of observations in each of the two expanded intervals.
- If only one peak was found, the corresponding interval $[k, l]$ is used as the distance estimate unless the peak was at a negative distance in which case no distance estimate is calculated.
- If two peaks were found and the interval $[k, l]$ for the largest peak contains less than 1% of all observations, the distance is not estimated.
- If two peaks were found and the interval $[k, l]$ for the largest peak contains $< 2X$ observations compared to the smaller peak, the distance estimate is only computed if the range of

distances is positive for the largest peak and negative for the smallest peak. If this is the case the interval $[k, l]$ for the positive peak is used as a distance estimate.

- If two peaks were found and the largest peak has $\geq 2X$ observations compared to the smaller peak, the interval $[k, l]$ corresponding to the largest peak is used as the distance estimate.

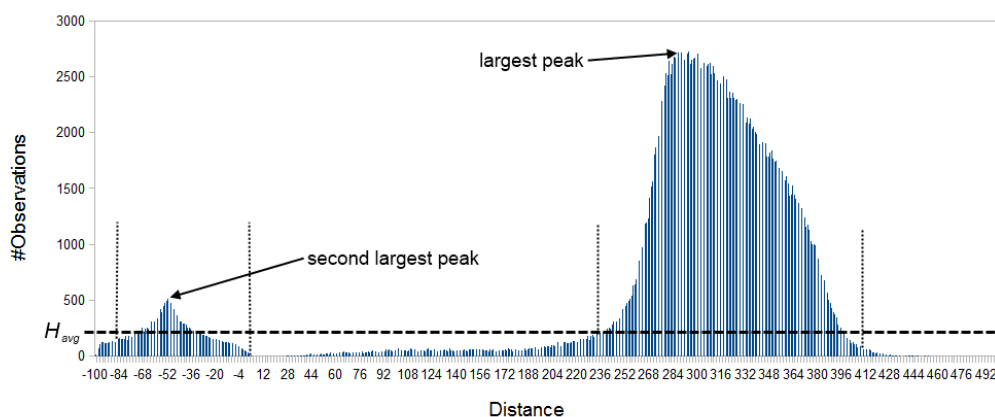


Figure 35.11: Histogram of paired distances where H_{avg} is indicated by the horizontal dashed line. There is two peaks, one is at a negative distance while the other larger peak is at a positive distance. The extended interval $[k, l]$ for each peak is indicated by the vertical dotted lines.

Optimization of the graph using paired reads

When paired reads are available, we can use the paired information to resolve large repeat regions that are not spanned by individual reads, but are spanned by read pairs. Given a set of paired reads that align to two nodes connected by a repeat region, the repeat region may be resolved for those nodes if we can find a path connecting the nodes with a length corresponding to the paired read distance. However, such a path must be supported by a minimum of four sets of paired reads before the repeat is resolved.

If it's not possible to resolve the repeat, scaffolding is performed where paired read information is used to determine the distances between contigs and the orientation of these. Scaffolding is only considered between two contigs if both are at least 120 bp long, to ensure that enough paired read information is available. An iterative greedy approach is used when performing scaffolding where short gaps are closed first, thus increasing the paired read information available for closing gaps (see figure 35.12).

Contigs in the same scaffold are output as one large contig with Ns inserted in between. The number of Ns inserted correspond to the estimated distance between contigs, which is calculated based on the paired read information. More precisely, for each set of paired reads spanning two contigs a distance estimate is calculated based on the supplied distance between the reads. The average of these distances is then used as the final distance estimate. The distance estimate will often be negative which happens when the paired information indicate that two contigs overlap. The assembler will attempt to align the ends of such contigs and if a high quality overlap is found the contigs are joined into a single contig. If no overlap is found, the distance estimate is set to two so that all remaining scaffolds have positive distance estimates.

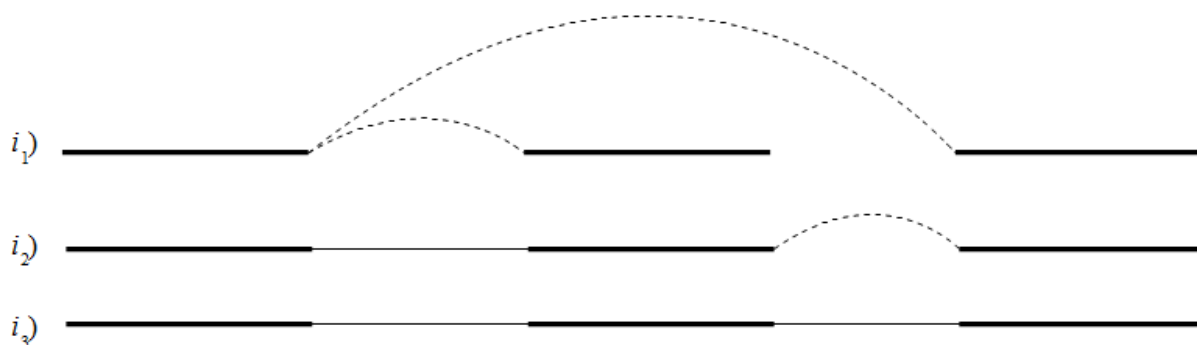



Figure 35.12: Performing iterative scaffolding of the shortest gaps allows long pairs to be optimally used. i_1 shows three contigs with dashed arches indicating potential scaffolding. i_2 is after first iteration when the shortest gap has been closed and long potential scaffolding has been updated. i_3 is the final results with three contigs in one scaffold.

Furthermore, Ns can also be present in output contigs in cases where input sequencing reads themselves contain Ns.

Additional information about repeats being resolved using paired reads and scaffolded contigs is available as annotations on the contig sequences and as summary in the report (see section 35.1.5). This information can also be exported to the AGP format (see section 35.1.1).

The annotations in table format can be viewed by clicking the "Show Annotation Table" icon  at the bottom of the Viewing Area. "Show annotation types" in the side panel allows you to select the annotation "Scaffold" among a list of other annotations. The annotations tell you about the scaffolding that was performed by the de novo assembler. That is, it tells you where particular contigs and those areas containing complete sequence information were joined together across regions without complete sequence information.

For the GFF format there are three types of annotations:

- **Scaffold** refers to the estimated gap region between two contigs where Ns are inserted.
- **Contigs joined** refers to the joining of two contigs connected by a repeat or another ambiguous structure in the graph, that was resolved using paired reads. Can also refer to overlapping contigs in a scaffold that were joined using an overlap.
- **Alternatives excluded** refers to the exclusion of a region in the graph using paired reads that resulted in a join of two contigs.

AGP export

The AGP annotations describe the components that an assembly consists of. This format can be validated by the NCBI AGP validator.

If the exporter is executed on an assembly where the contigs have been updated using a read mapping, the N's in some scaffolds might be resolved if you select the option "Update contigs" (figure 35.13).

If the exporter encounters such a region, it will give a warning but not stop.

Currently we output two types of annotations in AGP format:

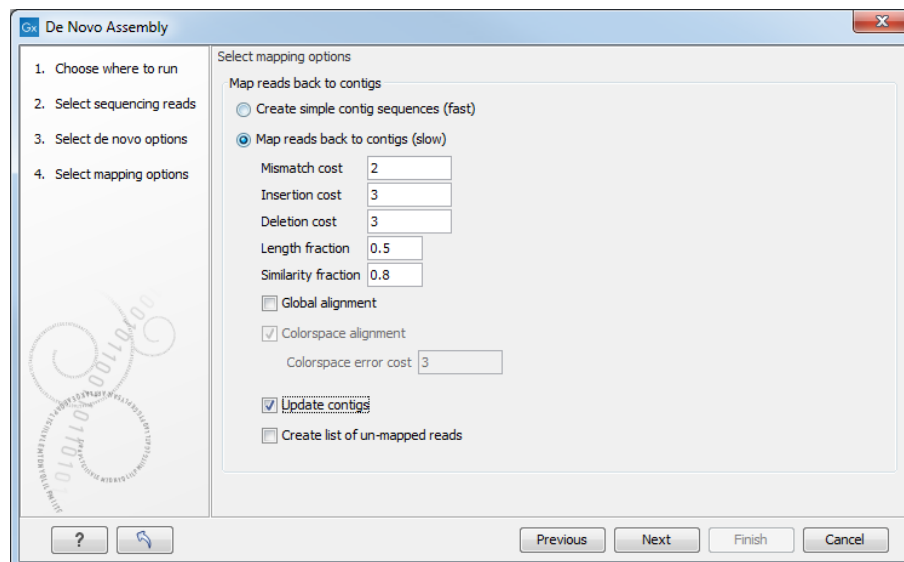


Figure 35.13: Select "update contigs" by ticking the box if you want to resolve scaffolds based on a read mapping.

- **Contig** a non-redundant sequence not containing any scaffolded regions.
- **Scaffold** the estimated gap region between two contigs.

Bubble resolution

Before the graph structure is converted to contig sequences, bubbles are resolved. As mentioned previously, a bubble is defined as a bifurcation in the graph where a path furcates into two nodes and then merge back into one. An example is shown in figure 35.14.



Figure 35.14: A bubble caused by a heterozygous SNP or a sequencing error.

In this simple case the assembler will collapse the bubble and use the route through the graph that has the highest coverage of reads. For a diploid genome with a heterozygous variant, there will be a fifty-fifty distribution of reads on the two variants, and this means that the choice of one allele over the other will be arbitrary. If heterozygous variants are important, they can be identified after the assembly by mapping the reads back to the contig sequences and performing standard variant calling. For random sequencing errors, it is more straightforward; given a reasonable level of coverage, the erroneous variant will be suppressed.

Figure 35.15 shows an example of a data set where the reads have systematic errors. Some reads include five As and others have six. This is a typical example of the homopolymer errors seen with the 454 and Ion Torrent platforms.

When these reads are assembled, this site will give rise to a bubble in the graph. This is not a problem in itself, but if there are several of these sites close together, the two paths in the graph will not be able to merge between each site. This happens when the distance between the sites is smaller than the word size used (see figure 35.16).

In this case, the bubble will be very large because there are no complete words in the regions

```

:AGATGACCAGGGTGTGCGATAAAAAATGCCAATCATCTGGAC,
:AGATGACCAGGGTGTGCGAT - AAAAAATGCCAATCATCTGGAC,
:AGATGACCAGGGTGTGCGAT - AAAAAATGCCAATCATCTGGAC,
:AGATGACCAGGGTGTGCGAT - AAAAAATGCCAATCATCTGGAC,
:AGATGACCAGGGTGTGCGATAAAAAATGCCAATCATCTGGAC,
:AGATGACCAGGGTGTGCGATAAAAAATGCCAATCATCTGGAC,
:AGATGACCAGGGTGTGCGAT - AAAAAATGCCAATCATCTGGAC,
:AGATGACCAGGGTGTGCGAT - AAAAAATGCCAATCATCTGGAC,
:AGATGACCAGGGTGTGCGAT - AAAAAATGCCAATCATCTGGAC,
:AGATGACCAGGGTGTGCGAT - AAAAAATGCCAATCATCTGGAC,

```

Figure 35.15: Reads with systematic errors.

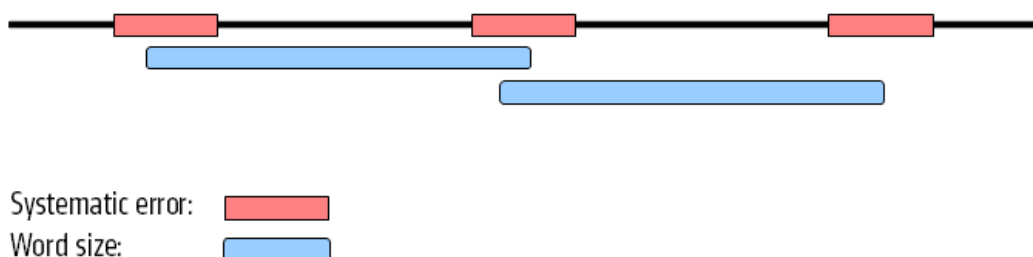


Figure 35.16: Several sites of errors that are close together compared to the word size.

between the homopolymer sites, and the graph will look like figure 35.17.

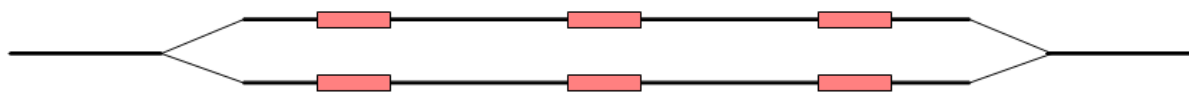


Figure 35.17: The bubble in the graph gets very large.

If the bubble is too large, the assembler will have to break it into several separate contigs instead of producing one single contig.

The maximum size of bubbles that the assembler should try to resolve can be set by the user. In the case from figure 35.17, a bubble size spanning the three error sites will mean that the bubble will be resolved (see figure 35.18).

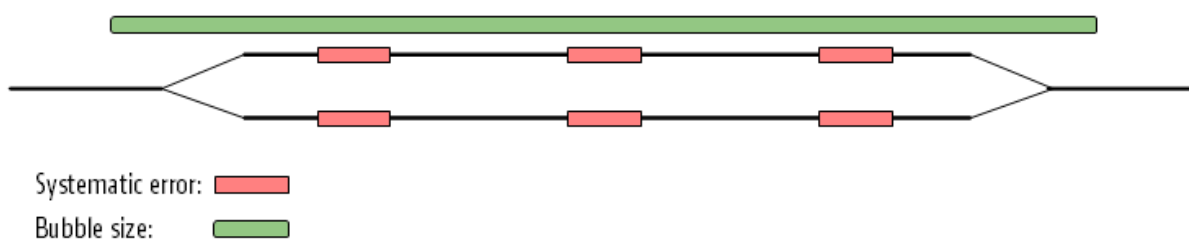


Figure 35.18: The bubble size needs to be set high enough to encompass the three sites.

While the default bubble size is often fine when working with short, high quality reads, considering the bubble size can be especially important for reads generated by sequencing platforms yielding long reads with either systematic errors or a high error rate. In such cases, a higher bubble size is recommended. For example, as a starting point, one could try half the length of the average read in the data set and then experiment with increasing and decreasing the bubble size in small

steps. For data sets with a high error rate it is often necessary to increase the bubble size to the maximum read length or more. Please keep in mind that increasing the bubble size also increases the chance of misassemblies.

Converting the graph to contig sequences

The output of the assembly is not a graph but a list of contig sequences. When all the previous optimization and scaffolding steps have been performed, a contig sequence will be produced for every non-ambiguous path in the graph. If the path cannot be fully resolved, Ns are inserted as an estimation of the distance between two nodes as explained in section 35.1.1.

Summary

So in summary, the de novo assembly algorithm goes through these stages:

- Make a table of the words seen in the reads.
- Build a *de Bruijn* graph from the word table.
- Use the reads to resolve the repeats in the graph.
- Use the information from paired reads to resolve larger repeats and perform scaffolding if necessary.
- Output resulting contigs based on the paths, optionally including annotations from the scaffolding step.

These stages are all performed by the assembler program.

35.1.2 Best practices

The de novo assembler is not designed to effectively use long mate-pair read information (insert size greater than 10 kp). Such data can be incorporated but may not lead to improvements in the final results. If paired end data are being assembled, the inclusion of mate-pair information in the same assembly can sometimes lead to worse results. In such cases, we advise that the long mate-pair data is marked as single (non-paired) data before including it in the assembly (see section 7.3.10).

Before you begin the Assembly

Input Data Quality Good quality data is key to a successful assembly. We strongly recommend using the Trim Reads tool:

- Trimming based on quality can reduce the number of sequencing errors that make their way to the assembler. This reduces the number of spurious words generated during an initial assembly phase. This then reduces the number of words that will need to be discarded in the graph building stage.
- Trimming Adapters from sequences is crucial for generating correct results. Adapter sequences remaining on sequences can lead to the assembler spending considerable time trying to join regions that are not biologically relevant. In other words this can lead to the assembly taking a long time and yielding misleading results.

Input Data Quantity In the case of de novo assembly, more data does not always lead to a better result as we are more likely to observe sequencing errors in high coverage regions. This is disadvantageous because overlapping sequencing errors can result in poor assembly quality. We therefore recommend using data sets with an average read coverage less than 100x. If you expect the average coverage of your genome to be greater than 100x, you can use the Sample Reads tool to reduce coverage. To determine how many reads you need to sample to obtain a maximum average coverage of 100x, you can do the following calculation:

- Obtain an estimated size of the genome you intend to assemble.
- Multiply this genome size by 100. This value will be the total number of bases you should use as input for assembly.
- Divide the total number of bases with the average length of your sequencing reads.
- Use this number as input for the number of reads to obtain as output from the Sample Reads tool.

Running the Assembly The two parameters that can be adjusted to improve assembly quality are Word Size and Bubble Size.

The default values for these parameters can work reasonably well on a range of data sets, but we recommend that you choose and evaluate these values based on what you know about your data.

Word Size If you expect your data to contain long regions of high quality, a larger Word Size, such as a value above 30 is recommended. If your data has a higher error rate, as in cases when homopolymer errors are common, a Word Size below 30 is recommended. Whenever possible, the Word Size should be less than the expected number of bases between sequencing errors.

Bubble Size When adjusting Bubble Size, the repeat structure of your genome should be considered in conjunction with the sequence quality. If you do not expect a repetitive genome you may wish to choose a higher bubble size to improve contiguity. If you anticipate more repeats, you may wish to use a smaller Bubble Size to reduce the possibility of collapsing repeat regions. In cases where the sequence quality is not high a larger bubble size may make more sense for your data.

If you are not sure of what parameters would be best suited for your data, we recommend identifying optimal settings for your de novo assembly empirically. To do so, you may run multiple assembly jobs with different parameters and compare the results.

However, comparing the results of multiple assemblies is often a challenge. For example, you may have one assembly with a large N50 (see section 35.1.5) and another with a larger total contig length. How do you decide which is better? Is the one with the large contig sizes better or the one with more total sequence? Ultimately, the answer to these questions will depend on what the goal of your downstream analysis is. To help with this comparison, we provide some basic guidelines in the sections below.

Evaluating and Refining the Assembly

Three key points to look for in assessing assembly quality are contiguity, completeness, and correctness.

Contiguity: How many contigs are there?

A high N50 and low number of contigs relative to your expected number of chromosomes are ideal. If you aren't sure what type of N50 and contig number might be reasonable to expect, you could try to get an idea by looking at existing assemblies of a similar genome, should these exist. For an even better sense of what would be reasonable for your data, you could make comparisons to an assembly of a similar genome, assembled using a similar amount and type of data. If your assembly results include a large number of very small contigs, it may be that you set the minimum contig length filter too low. Very small contigs, particularly those of low coverage, can generally be ignored.

Completeness: How much of the genome is captured in the assembly?

If a total genome length of 5MB is expected based on existing literature or similar genomes that have already been assembled, but the sum of all contig lengths is only 3.5MB, you may wish to reconsider your assembly parameters.

Two common reasons for an assembly output that is shorter than expected are:

- A Word Size that is higher than optimal for your data: A high Word Size will increase the probability of discarding words because they overlap with sequencing errors. If a word is seen only once, the unique word will be discarded even if there exist many other words that are identical except for one base (e.g., a sequencing error). A discarded word will not be considered in constructing the assembly graph and will therefore be excluded from the assembly contig sequences.
- A Bubble Size that is higher than optimal for your data: A high Bubble Size will increase the probability that two similar sequences are classified as a repeat and thus collapsed into a single contig. It is sometimes possible to identify collapsed repeats by looking at the mapping of your reads to the assembled contigs. A collapsed repeat will show as a high peak of coverage in one location.

Depending on the resources available for the organism you are working on, you might also assess assembly completeness by mapping the assembled contig sequences to a known reference. You can then check for regions of the reference genome that have not been covered by the assembled contigs. Whether this is sensible depends on the sample and reference organisms and what is known about their expected differences.

Correctness Do the contigs that have been assembled accurately represent the genome?

One key question in assessing correctness is whether the assembly is contaminated with any foreign organism sequence data. To check this, you could run a BLAST search using your assembled contigs as query sequences against a database containing possible contaminant species data. In addition to BLAST, checking the coverage can help to identify contaminant sequence data. The coverage of a contaminant contig is often different from the desired organism so you can compare the potential contaminant contigs to the rest of the assembled contigs. You may check for these types of coverage differences between contigs by:

- Map your reads used as input for the de novo assembly to your contigs (if you do not already have a mapping output);
- Create a Detailed Mapping Report;
- In the Result handling step of the wizard, check the option to Create separate table with statistics for each mapping;

- Review the average coverage for each contig in this resulting table.

If there are contigs that have good matches to a very different organism and there are discernable coverage differences, you could either consider removing those contigs from the assembly, or run a new assembly after removing the contaminant reads. One way to remove the contaminant reads would be to run a read mapping against the foreign organism's genome and to check the option to Collect unmapped reads. The unmapped reads Sequence List should now be clean of the contamination. You can then use this set of reads in a new de novo assembly.

Assessing the correctness of an assembly also involves making sure the assembler did not join segments of sequences that should not have been joined - or checking for mis-assemblies. This is more difficult. One option for identifying mis-assemblies is to try running the InDels and Structural Variants tool. If this tool identifies structural variation within the assembly, that could indicate an issue that should be investigated.

Post assembly improvements

If you are working with a smaller genome, the **CLC Genome Finishing Module** may be of interest to you. It has been developed to help finishing small genomes, such as microbes, eukaryotic parasites, or fungi, in order to reduce the extensive workload associated with genome finishing and to facilitate as many steps in the procedure as possible. The module can be downloaded from the Workbench Plugin Manager, or from our website at <https://digitalinsights.qiagen.com/plugins/clc-genome-finishing-module/>. A free trial license is available, as described at https://resources.qiagenbioinformatics.com/manuals/clcgenomefinishing/current/index.php?manual=Plugins_licenses.html.

35.1.3 Randomness in the results

Different runs of the de novo assembler can result in slightly different results. This is caused by multi-threading of the program combined with the use of probabilistic data structures. If you were to run the assembler using a single thread, the effect would not be observed. That is, the same results would be produced in every run. However, an assembly run on a single thread would be very slow. The assembler should run quickly. Thus, we use multiple threads to accelerate the program.

The main reason for the assembler producing different results in each run is that threads construct contigs in an order that is correlated with the thread execution order, which we do not control. The size and "position" of a contig can change dramatically if you start building a contig from two different starting points (i.e. different words, or k-mers), which means that different assembly runs can lead to different results, depending on the order in which threads are executed. Whether a contig is scaffolded with another contig can also be affected by the order that contigs are constructed. In this case, you could see quite large differences in the lengths of some contigs reported. This will be particularly noticeable if you have an assembly with reasonably few contigs of great length.

For the moment, the output of runs may vary slightly, but the overall information content of the assembly should not be markedly different.

35.1.4 De Novo Assembly parameters

To start the assembly:

Tools | De Novo Sequencing (📁) | De Novo Assembly (🔧)

In this dialog, you can select one or more sequence lists or single sequences.

Click **Next** to set the parameters for the assembly. This will show a dialog similar to the one in figure 35.19.

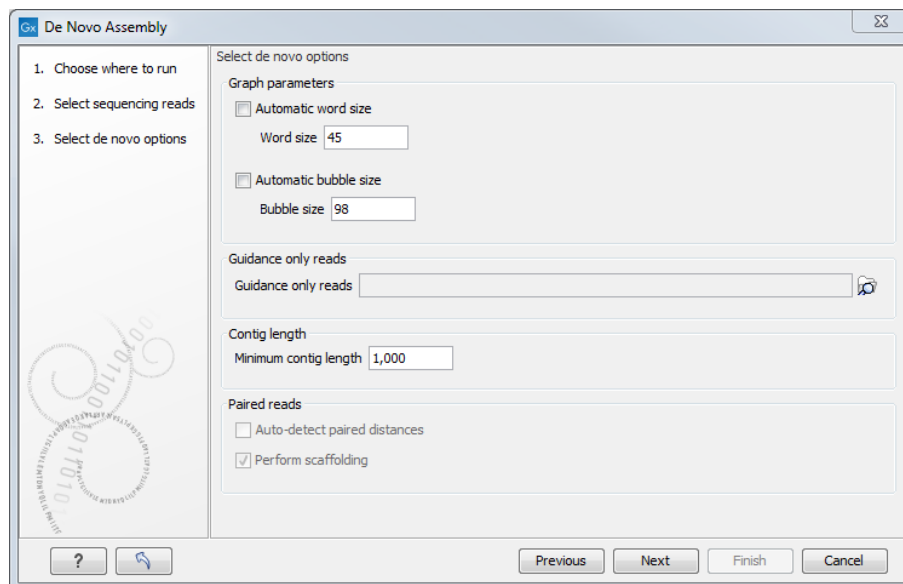


Figure 35.19: Setting parameters for the assembly.

At the top, you select the **Word size** and the **Bubble size** to be used. The principles of setting the word size are described in section 35.1.1. When using automatic calculation, you can see the word size in the **History** (📄) of the result files. Please note that the range of word sizes is 12-64 on 64-bit computers.

The meaning of the bubble size parameter is explained in section 35.1.1. The automatic bubble size is set to 50, unless one of the following conditions apply:

- some of the reads are from either Ion torrent or PacBio;
- the reads are not all Sanger reads and average read length of all input reads is >160bp.

In these cases the bubble size is set to the average read length of all input reads. The value used is also recorded in the **History** (📄) of the result files.

The next option is to specify **Guidance only reads**. The reads supplied here will not be used to create the *de Bruijn* graph and subsequent contig sequence but only used to resolved ambiguities in the graph (see section 35.1.1 and section 35.1.1). With mixed data sets from different sequencing platforms, we recommend using sequencing data with low error rates as the main input for the assembly, whereas data with more errors should be specified only as **Guidance only reads**. This would typically be long reads or paired data sets.

You can also specify the **Minimum contig length** when doing de novo assembly. Contigs below this length will not be reported. The default value is 200 bp. For very large assemblies, the

number of contigs can be huge (over a million), in which case the data structures when mapping reads back to contigs will be very large and take a very long time to handle. In this case, it is a great advantage to raise the minimum contig length to reduce the number of contigs that have to be incorporated into this data structure.

At the bottom, there is an option to **Perform scaffolding**. The scaffolding step is explained in greater detail in section 35.1.1. This will also cause scaffolding annotations to be added to the contig sequences (except when you also choose to Update contigs, see below).

Finally, there is an option to **Auto-detect paired distances**. This will determine the paired distance (insert size) of paired data sets. If several paired sequence lists are used as input, a separate calculation is done for each one to allow for different libraries in the same run. The **History** (📄) view of the result will list the distance used for each data set.

If the automatic detection of pairs is not checked, the assembler will use the information about minimum and maximum distance recorded on the input sequence lists (see section 7.3.10).

For mate-pair data sets with large insert sizes, it may not be possible to infer the correct paired distance. In this case, the automatic distance calculation should not be used.

The best way of checking this is to run a read mapping using the contigs from the de novo assembly as reference and the mate-pair library as reads, and then check the mapping report (see section 29.3). There is a paired distance distribution graph that can be used to check whether the distance estimated by the assembler fits in the distribution found in the read mapping.

When you click **Next**, you will see the dialog shown in figure 35.20

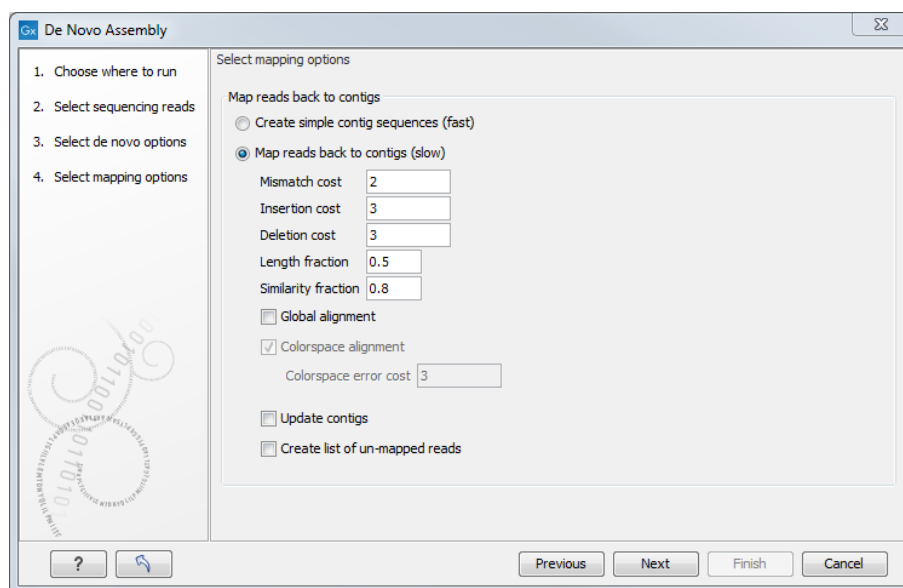


Figure 35.20: Parameters for mapping reads back to the contigs.

There are two general types of output you can generate from the de novo assembly tool:

- **Simple contigs:** the output is a sequence list of the contigs generated.
- **Stand-alone mappings:** a read mapping is carried out after the de novo assembly, where the sequence reads used for the assembly are mapped to the contigs that were assembled.

If you choose to perform a read mapping, you can specify some parameters that are explained in section 30.1.3. The placement of reads that map in more than one position equally well are placed randomly (see section 30.1.5) and the type of gap costs used here are linear.

At the bottom, you can choose to Update contigs based on the subsequent mapping of the input reads back to the contigs generated by the de novo assembly. In general terms, this has the effect of updating the contig sequences based on the evidence provided by the subsequent mapping back of the read data to the de novo assembled contigs. The following are the impacts of choosing this option:

- Contig regions must be supported by at least one read mapping back to them in order to be included in the output. If more than half of the reads in a column of the mapping contain a gap, then a gap will be inserted into the contig sequence. Contig regions where no reads map will be removed. Note that if such a region occurs within a contig, it is removed and the surrounding regions are joined together.
- The most common nucleotide among the mapped reads at a given position is the one assigned to the contig sequence. In NGS data, it would be very unlikely that at a given position there would be an equal number of reads with different nucleotides. Should this occur however, then the nucleotide that comes first in the alphabet would be included in the consensus.

Note that if the "Update contigs" option is selected, the contig lengths may get below the threshold specified in figure 35.19 because this threshold is applied to the original contig sequences. If the "Update contigs" based on mapped reads option is not selected, the original contig sequences from the assembler will be preserved completely also in situations where the reads that are mapped back do not support the contig sequences.

Finally, in the last dialog of the de novo assembly, you can choose to create a report of the results.

35.1.5 De Novo Assembly output

Simple contigs

The output is a sequence list of the contigs generated (figure 35.21), that can also be seen as a table and an annotation table as described for the stand-alone read mapping described above.

Stand-alone mapping

The de novo assembly is followed by a step where the reads used to generate the contigs are mapped against the contigs, using them as reference.

This output is called assembly and opens as a table listing all contigs as seen in figure 35.22.

The information included in the table is:

- **Name.** When mapping reads to a reference, this will be the name of the reference sequence.
- **Consensus length.** The length of the consensus sequence. Subtracting this from the length of the reference will indicate how much of the reference that has not been covered by reads.



Figure 35.21: A de novo assembly read mapping seen as annotation table.

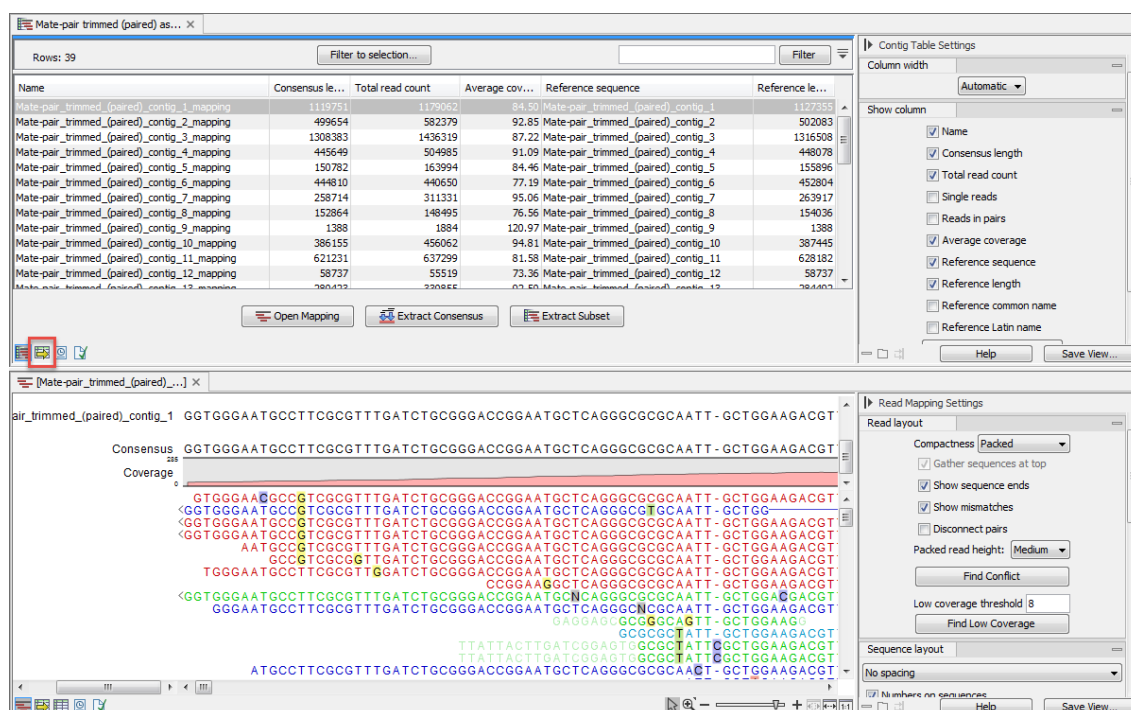


Figure 35.22: A de novo assembly read mapping.

- **Total read count.** The number of reads. Reads with multiple hits on different reference sequences are placed according to your input for **Non-specific matches**
- **Single reads** and **Reads in pair.** Total number of reads, single and/or in pair.
- **Average coverage.** This is simply summing up the bases of the aligned part of all the reads divided by the length of the reference sequence.
- **Reference sequence.** The name of the reference sequence.
- **Reference length.** The length of the reference sequence.

- **Reference common name** and **Reference latin name**. Name, common name and Latin name of each reference sequence.

At the bottom of the table there are three buttons that can be used to open or extract sequences. Select the relevant rows before clicking on the buttons:

- **Open Mapping**. Opens the read mapping for visual inspection. You can also open one mapping simply by double-clicking in the table.
- **Extract Consensus/Contigs**. For de novo assembly results, the contig sequences will be extracted. For results when mapping against a reference, the **Extract Consensus** tool will be used (see section 30.7).
- **Extract Subset**. Creates a new mapping table with the mappings that you have selected.

Double clicking on a contig name will open the read mapping in split view.

It is possible to open the assembly as an annotation table (using the icon highlighted in figure 35.22). The annotations available in the table are the following (see figure 35.23):

- **Alternatives Excluded**. More than one path through the graph was possible in this region but evidence from paired data suggested the exclusion of one or more alternative routes in favour of the route chosen.
- **Contigs Joined**. More than one route was possible through the graph such that an unambiguous choice of how to traverse the graph cannot be made. However evidence from paired data supports one of these routes and on this basis, this route is selected (and other routes excluded).
- **Scaffold**. The route through the graph is not clear but evidence from paired data supports the connection of two contigs. A single contig is then reported with N characters between the two connected regions. This entity is also known as a scaffold. The number of N characters represents the expected distance between the regions, based on the evidence the paired data.

Using the menu in the right end side panel, it is possible to select only one type of annotations to be displayed in the table.

De Novo Assembly report

A de novo assembly report looks like the one shown in figure 35.24.

The report contains the following information when both scaffolding and read mapping is performed:

Nucleotide distribution This includes Ns when scaffolding has been performed.

Contig measurements This section includes statistics about the number and lengths of contigs. When scaffolding is performed and the update contigs option is not selected, there will be two separate sections with these numbers: one including the scaffold regions with Ns and one without these regions.

Sequence	Name	Type	Region	Qualifiers
Mate-pair_trimmed_(paired)_contig_3	Scaffold	Scaffold	1266431..1266441	/score=11 /source=CLC De Novo Assembler /note=Estimated region of a gap between two contigs based on paired reads
Mate-pair_trimmed_(paired)_contig_4	Alternatives excluded	Alternatives excluded	38945..38990	/score=46 /source=CLC De Novo Assembler /note=An unknown graph structure could be excluded using paired reads and eventually lead to joining two contigs into one
Mate-pair_trimmed_(paired)_contig_4	Alternatives excluded	Alternatives excluded	89337..89382	/score=46 /source=CLC De Novo Assembler /note=An unknown graph structure could be excluded using paired reads and eventually lead to joining two contigs into one
Mate-pair_trimmed_(paired)_contig_4	Contigs joined	Contigs joined	44244..44665	/score=422 /source=CLC De Novo Assembler /note=A repeat or another ambiguous structure was solved using paired reads and two contigs could be joined into one

Figure 35.23: A de novo assembly read mapping seen as annotation table.

1 Summary de novo report

1.1 Nucleotide distribution

Nucleotide	Count	Frequency
Adenine (A)	1.113.919	24,5%
Cytosine (C)	1.142.129	25,2%
Guanine (G)	1.157.663	25,5%
Thymine (T)	1.118.847	24,6%
Any nucleotide (N)	6.409	0,1%

1.2 Contig measurements

	Length
N75	41.694
N50	80.414
N25	132.325
Minimum	202
Maximum	191.299
Average	32.421
Count	140
Total	4.538.967

Figure 35.24: Creating a de novo assembly report.

- **N25, N50 and N75** The N25 contig set is calculated by summarizing the lengths of the biggest contigs until you reach 25 % of the total contig length. The minimum contig length in this set is the number that is usually used to report the N25 value of a de novo assembly. The same goes with N50 and N75 which are the 50 % and 75 % of the total contig length, respectively.
- **Minimum, maximum and average** This refers to the contig lengths.
- **Count** The total number of contigs.
- **Total** The number of bases in the result. This can be used for comparison with the estimated genome size to evaluate how much of the genome sequence is included in the assembly.

Accumulated contig lengths This shows the summarized contig length on the y axis and the number of contigs on the x axis, with the biggest contigs ranked first. This answers the question: how many contigs are needed to cover e.g. half of the genome.

If the de novo assembly was followed by a read mapping, it is possible to have the following additional sections.

Summary statistics Gives the count, average length and total bases amount for all reads, matched and non-matched reads, contigs, reads in pairs, and broken paired reads.

Distribution of read length For each sequence length, you can see the number of reads and the distribution in percent. This is mainly useful if you don't have too much variance in the lengths as in Sanger sequencing data for example.

Distribution of matched read length Equivalent to the above, except that this includes only the reads that have been matched to a contig.

Distribution of non-matched read length Shows the distribution of lengths of the unmapped sequences.

Paired reads distance distribution Shows the distribution of paired reads distances.

For a more detailed report, use the **QC for Read Mapping** tool, and see the description of the report in section 29.3.

35.2 Map Reads to Contigs

The Map Reads to Contigs tool allows mapping of reads to contigs. This can be relevant in situations where the reference of a mapping is contigs, such as when:

- Contigs have been imported from an external source
- The output from a de novo assembly is contigs with no read mapping
- You wish to map a new set of reads or a subset of reads to the contigs

The Map Reads to Contigs tool is similar to the Map Reads to Reference tool in that both tools accept the same input reads, and make use of the same read mapper in accordance to the reads input (see the introduction of section 30.1).

The main difference between the two tools is the output. The output from the Map reads to contigs tool is a de novo object that can be edited, in contrast to the reference sequence used when mapping reads to a reference.

To run the Map Reads to Contigs tool:

Tools | De Novo Sequencing (📁) | Map Reads to Contigs (🔍)

This opens up the dialog in figure 35.25 where you select the reads you want to map to the contigs. Click **Next**.

Select the contigs to map the reads against (figure 35.26).

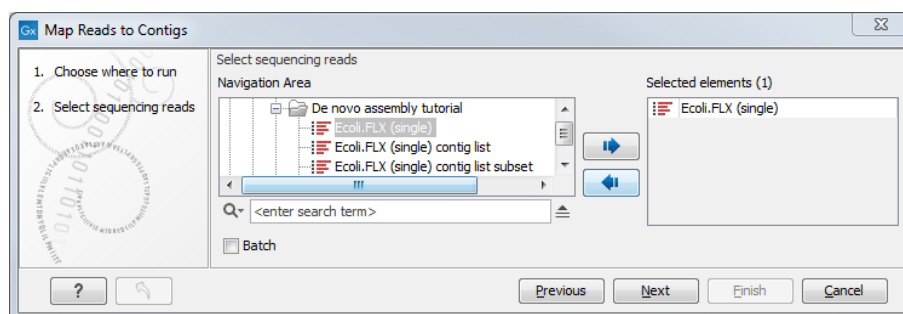


Figure 35.25: Select reads. The contigs will be selected in the next step.

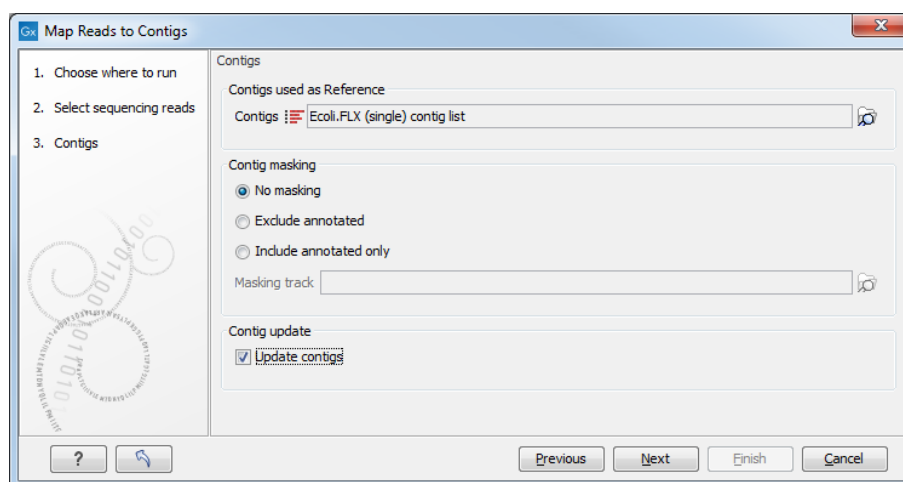


Figure 35.26: Select contigs and specify whether to use masking and the "Update contigs" function.

Under "Contig masking", specify whether to include or exclude specific regions (for a description of this see section 30.1.2).

The contigs can be updated by selecting "Update contigs". The advantage of using this option during read mapping is that the read mapper is better than the de novo assembler at handling errors in reads. Specifically, the actions taken when contigs are updated are:

- Regions of a contig reference, where no reads map, are removed. This leads to a joining of the surrounding regions of the contig as shown in the example in figure 35.27).
- In the case of locations where reads map to a contig reference, but there are some mismatches to that contig, the contig sequence is updated to reflect the majority base at that location among the reads mapped there. If more than half of the reads contain a gap at that location, the contig sequence will be updated to include the gap.

In the Mapping options dialog, the parameters of the Map Reads to Contigs tool are identical to the ones described for the Map Reads to Reference tool (see section 30.1.3).

The output from the Map Reads to Contigs tool can be a track or stand-alone read mappings. If the option to update contigs has been enabled, the track-based output cannot be selected.

When stand-alone read mappings have been selected as output, it is possible to edit and delete contig sequences.

Before the contig is updated:

Reference: AACCTT

Read 1 AA

Read 2 TT

After the contig is updated:

Reference: AATT

Read 1: AA

Read 2: TT

Figure 35.27: When selecting "Update Contig" in the wizard, contigs will be updated according to the reads. This means that regions of a contig where no reads map will be removed.

Figure 35.28 shows two stand-alone read mappings generated by using Map Reads to Reference (top) and Map Reads to Contigs (bottom) on the exact same reads and contigs as input. Contig 1 from both analyses have been opened from their respective Contig Tables. The differences are highlighted with red arrows. The output from the Map Reads to Reference has a consensus sequence; in the output from Map Reads to Contigs, the Contig itself is the consensus sequence if "Update contigs" was selected.

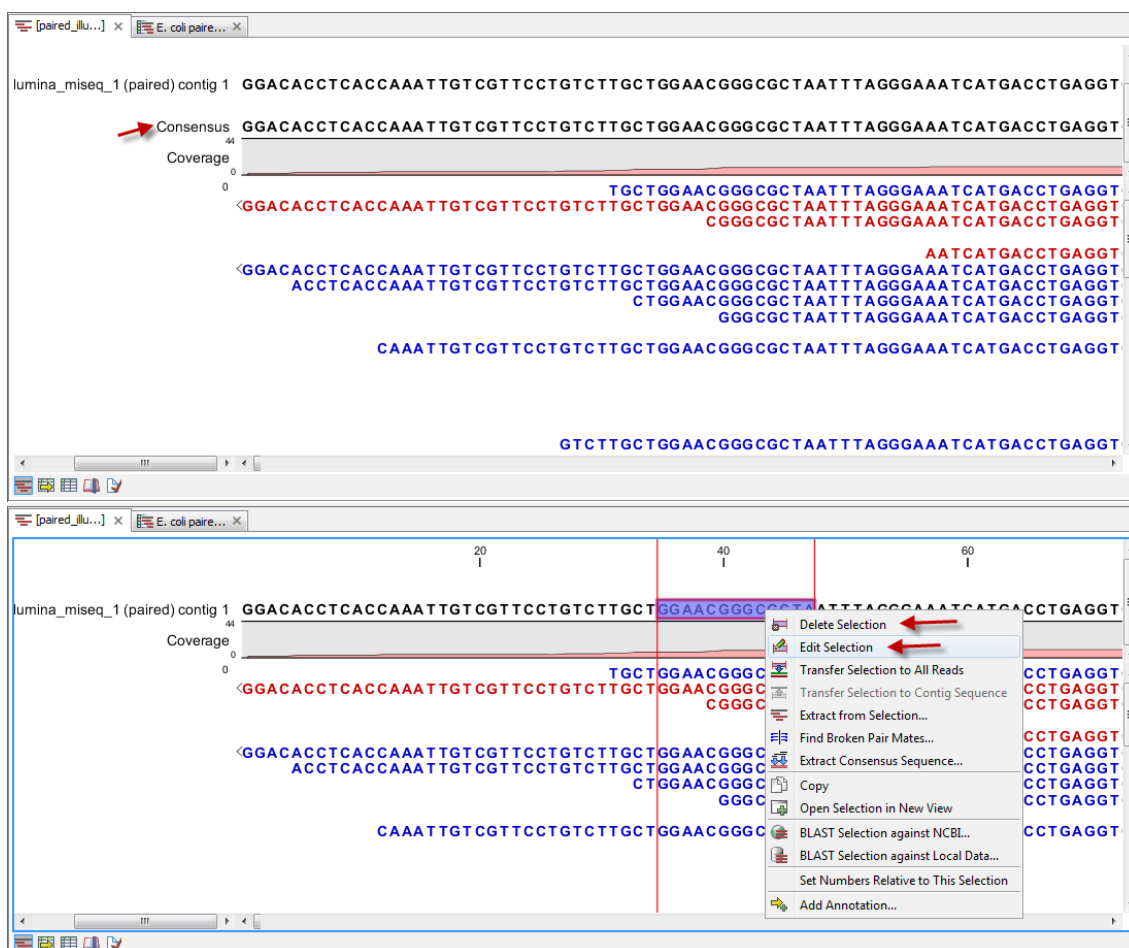


Figure 35.28: Two different read mappings performed with Map Reads to Reference (top) and Map Reads to Contigs (bottom). The differences are highlighted with red arrows.

35.3 De Novo Assemble Long Reads

The De Novo Assemble Long Reads tool facilitates the generation of de novo assemblies from long reads. It is based on open source tools: Raven [Vaser and Šiki, 2021] for analysis of Oxford Nanopore reads and Pacific Biosciences CLR reads, and hifiiasm [Cheng et al., 2021] for assembly of Pacific Biosciences HiFi reads.

The De Novo Assemble Long Reads tool works best with uncorrected long reads why it is not recommended to run the Correct Long Reads (legacy) tool beforehand.

Note that in order to create a de novo assembly, the tool requires a sufficient overlap between reads in the order of hundreds of bases.

35.3.1 De Novo Assemble Long Reads parameters

To run the De Novo Assemble Long Reads tool, go to:

Tools | De Novo Sequencing (📁) | De Novo Assemble Long Reads (🧬)

Select one or more sequence lists containing long reads.

Click **Next** to set the parameters for the assembly (figure 35.29):

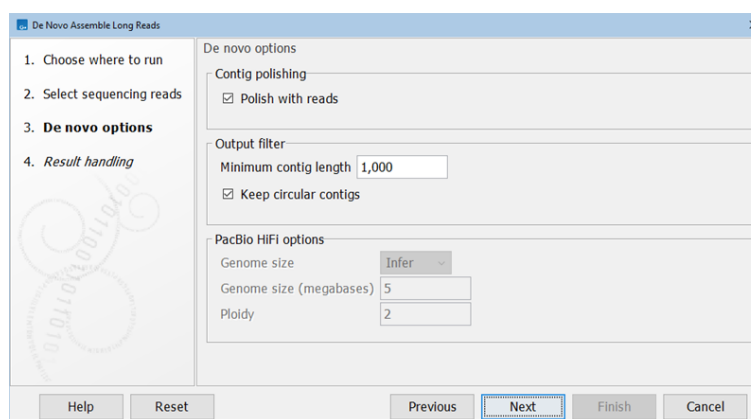


Figure 35.29: The De Novo Assemble Long Reads parameters.

- **Polish with reads.** When this parameter is set, two iterations of read polishing are run on the initial contigs. Disabling this parameter results in raw contigs with error rates similar to the error rate of the reads. This option is disabled for PacBio HiFi reads.
- **Minimum contig length.** The minimum length of contigs included in the output. Shorter contigs will be filtered.
- **Keep circular contigs.** When enabled, the minimum contig length filtering is not applied to circular contigs. This means that all circular contigs will be output regardless of length.
- **PacBio HiFi options** (enabled for PacBio HiFi reads only)
 - **Genome size.** *Infer* automatically determines the genome size as part of the analysis. *Manual* instructs the algorithm to use the genome size specified in the text field below for inferring read coverage.

- **Genome size (megabases)**. Enter the expected genome size.
- **Ploidy**. The number of expected alleles. If it is set to >2, the quality of the assembly for polyploid genomes might be improved.

When running **De Novo Assemble Long Reads** in a workflow, the workflow element dialog will contain an additional option:

- **PacBio HiFi**. Check this option to indicate that input reads are PacBio HiFi reads. When selected, the tool will run with an algorithm optimized for HiFi reads.

When running the De Novo Assemble Long Reads tool separately, the read type is inferred from the input reads.

35.3.2 De Novo Assemble Long Reads output

In addition to the sequence list of assembled contigs, the following outputs are available:

- **Create report**. Creates a summary report.
- **Create assembly graph**. Generates a visual representation of the assembly showing the contigs and connections between them.

De Novo Assemble Long Reads report

The assembly report contains information on the base and length distributions of the contigs. An example of the first sections of the report is shown in figure 35.30.

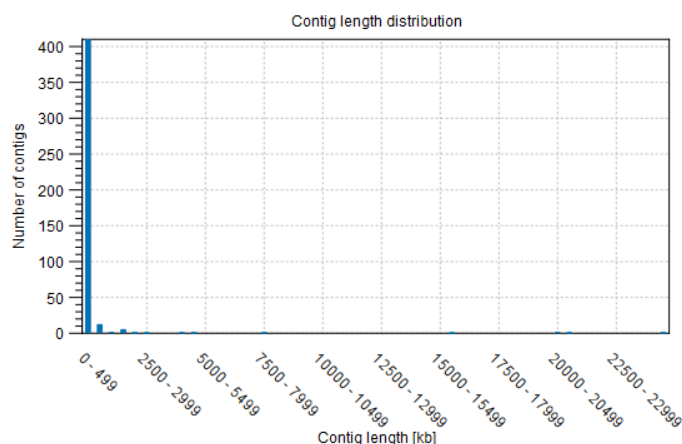
- **Nucleotide distribution**.
- **Contig measurements**. Statistics about the number and lengths of contigs.
 - **Contigs**. The number of contigs.
 - **Minimum, Maximum, Average**. Minimum, maximum and average contig length.
 - **N50**. The length of the shortest contig in sets of contigs of equal length or longer, where the summed length of contigs is at least 50% of the total contig length. As such, N50 is the shortest contig length that must be included to cover 50% of the assembly.
 - **N90**. The length of the shortest contig in a set of contigs of equal length or longer, where the summed length of contigs is at least 90% of the total contig length. As such, N90 is the shortest contig length that must be included to cover 90% of the assembly. N90 will be equal to or smaller than N50.
 - **Total**. The number of bases in the contigs. This can be used for comparison with the estimated genome size to evaluate how much of the genome sequence is included in the assembly.
- **Contig length distribution**. The number of contigs found at a specific length.

1 Nucleotide distribution

Nucleotide	Count	Frequency (%)
Adenine (A)	50,022,125	28.98
Cytosine (C)	36,377,066	21.07
Guanine (G)	36,229,844	20.99
Thymine (T)	50,008,864	28.97

2 Contig measurements

Contigs	441
Minimum	10,824
Maximum	24,726,942
Average	391,469
N50	15,797,497
N90	159,247
Total	172,637,899

Figure 35.30: *De Novo Assemble Long Reads report*

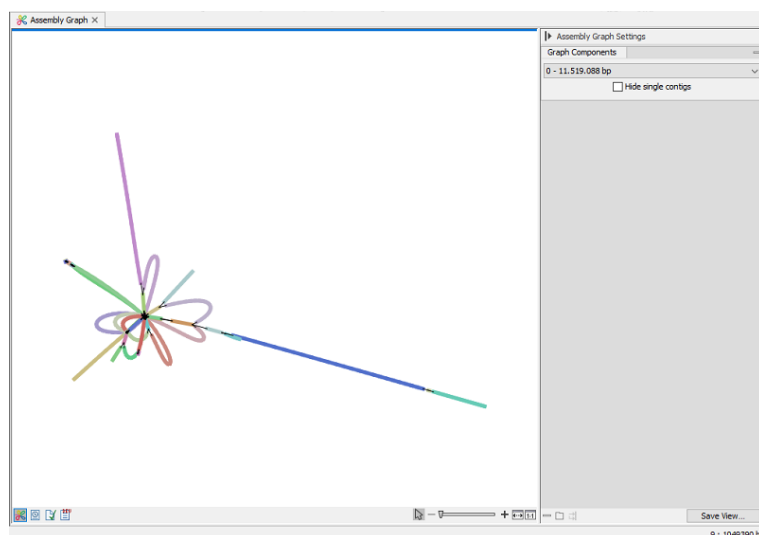
- **Accumulated contig length.** The y-axis shows the summed contig length, while the x-axis represents the number of contigs, arranged with the largest contigs first. This provides insight into the number of contigs required to cover, for instance, half of the genome.

For HiFi data, the report contains an additional section:

- **K-mer coverage.**
 - **K-mer plot.** Illustrates the frequency distribution of k-mers based on their occurrences. For homozygous samples, a single peak is expected around the read coverage. For heterozygous samples, two peaks should be visible, with the smaller peak corresponding to the heterozygous read coverage and the larger peak corresponding to the homozygous read coverage.
 - **Homozygous coverage peak.** Estimated homozygous coverage.
 - **Heterozygous coverage peak.** Estimated heterozygous coverage.

Assembly graph

The assembly graph provides visual representations of the sequences in the contig list, and sequences that were excluded.

Figure 35.31: *Assembly graph view*

The viewer shows a single graph component (group of connected sequences) at a time. You switch between them using the Side Panel *Graph components* dropdown. To exclude graphs composed of a single contig, select **Hide single contigs**.

When hovering over a contig, its name and length are accessible as a tooltip and in the lower right corner of the view.

Dragging a contig initiates a layout animation. The animation aims to adhere to a graph layout governed by a set of force-directed rules. Consequently, when a contig is moved in a specific direction, the algorithm will strive to return to a low-energy state.

35.4 Polish Contigs with Reads

The **Polish Contigs with Reads** tool facilitates the process of refining a set of contigs with high quality reads. This enables the creation of hybrid assemblies by first creating an assembly from long, error prone reads, and subsequently using high-quality reads to polish the contigs.

Before polishing, the high-quality reads should be stripped of adapters and lower quality bases. This can be done using **Trim Reads** (section 28.2).

The tool uses Racon [Vaser et al., 2017] with additional improvements inspired by Minipolish [Wick and Holt, 2019]. Racon uses a divide-and-conquer approach for rapid consensus calling. The partial order alignment (POA) of the reads against the target sequences occurs in non-overlapping windows on the target sequences. This approach has the consequence that Racon may not always use a globally optimal alignment of reads for consensus calling.

Polishing is conducted in two steps. Following each step, a set of corrections are carried out to improve contig quality:

- Contig ends that were truncated by Racon are reintroduced.
- Circular contigs are rotated by half the sequence length in each of the iterations.
- For circular contigs, the mapping is corrected for reads that span the junction.

To run the **Polish Contigs with Reads** tool, go to:

Tools | De Novo Sequencing (🔧) | Polish Contigs with Reads (🔧)

Select a contig sequence list.

In the next dialog, set the polishing parameters (figure 35.32):

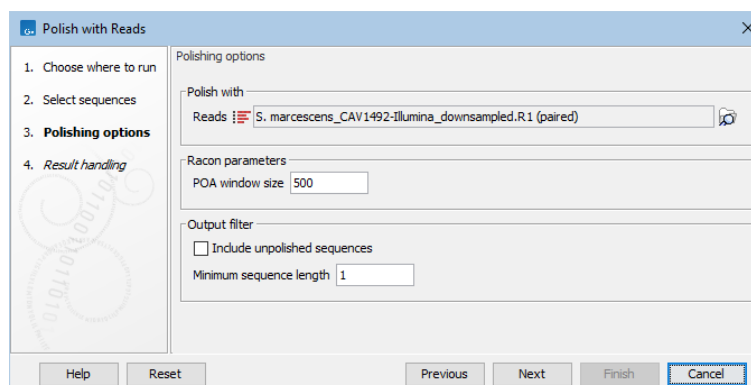


Figure 35.32: *Polish Contigs with Reads parameters*

- **Reads.** Select a sequence list of trimmed, high-quality reads to be used for polishing.
- **POA window size.** The window size for which Racon computes partial order alignments (POA). A larger window size enhances the ability to capture more global structure during the polishing process, but it also increases the memory requirement.
- **Include unpolished sequences.** Check the checkbox to keep sequences for which polishing was not possible.
- **Minimum sequence length.** The minimum length of sequences to be included in the output.

35.4.1 Polish Contigs with Reads output

In addition to the sequence list of polished sequences, a summary report is available via the output option **Create report**.

Polish Contigs with Reads report

An example of the first sections of the Polish Contigs with Reads summary report is shown in figure 35.33.

The report contains the following information:

- **Polishing summary**
 - **Input sequences.** Number of target input sequences.
 - **Input reads.** Number of high-quality input reads.
 - **Alignments (includes multiple alignments).** The number of alignments between target sequences and high-quality reads.

1 Polishing summary

Input sequences	6
Input reads	300,000
Alignments (includes multiple alignments)	284,345
Polished sequences	6
Polished windows (%)	99.51

2 Sequences

Name	Nucleotides	Mapped reads	Polished windows (%)
Utg2138	5,204,845	151,491	99.82
Utg2140	268,426	6,321	99.63
Utg2144	94,163	17,726	81.48
Utg2132	69,147	2,250	100.00
Utg2136	3,229	84,858	100.00
Utg2142	199,410	21,699	99.75

3 Sequence statistics

Nucleotide	Count	Frequency (%)
Adenine (A)	1,205,076	20.64
Cytosine (C)	1,708,305	29.26
Guanine (G)	1,712,751	29.33
Thymine (T)	1,213,088	20.77

Sequences	6
Minimum	3,229
Maximum	5,204,845
Average	973,203
N50	5,204,845
N90	268,426
Total	5,839,220

Figure 35.33: Polish Contigs with Reads report

- **Polished sequences.** Number of sequences that were polished.
- **Polished windows (%).** The percentage of windows that were polished.
- **Sequences.** For each target input sequence, the following information is given:
 - **Name.** Name of the sequence.
 - **Nucleotides.** Number of nucleotides in the sequence.
 - **Mapped reads.** Number of high-quality reads that mapped to the sequence.
 - **Polished windows (%).** The percentage of windows that were polished.
- **Sequence statistics**
 - **Sequences.** Number of sequences.
 - **Minimum, Maximum, Average.** Minimum, maximum and average sequence length.
 - **N50.** The length of the shortest sequence in sets of sequences of equal length or longer, where the summed length of sequences is at least 50% of the total sequence length.
 - **N90.** The length of the shortest sequence in a set of sequences of equal length or longer, where the summed length of sequence is at least 90% of the total sequence length. N90 will be equal to or smaller than N50.

- **Total.** The number of bases in the sequences.
- **Sequence length distribution.** A graph depicting the number of sequences found at a specific length.

Chapter 36

Epigenomics analysis

Contents

36.1 Histone Chip-Seq	1148
36.2 ChIP-Seq Analysis	1151
36.2.1 Quality Control of ChIP-Seq data	1151
36.2.2 Learning peak shapes	1152
36.2.3 Applying peak shape filters to call peaks	1153
36.2.4 Running the Transcription Factor ChIP-Seq tool	1154
36.2.5 Peak track	1156
36.3 Bisulfite Sequencing	1157
36.3.1 Detecting DNA methylation	1157
36.3.2 Map Bisulfite Reads to Reference	1160
36.3.3 Call Methylation Levels	1169
36.3.4 Create RRBS-fragment Track	1175
36.4 Advanced Peak Shape Tools	1175
36.4.1 Learn Peak Shape Filter	1176
36.4.2 Apply Peak Shape Filter	1177
36.4.3 Score Regions	1179

36.1 Histone Chip-Seq

ChIP-Seq experiments are increasingly used to investigate histone modifications. In contrast to transcription factors, histone marks are of variable length and can span across entire gene bodies. Although the experimental procedures are similar, the resulting data needs to be treated accordingly to take this variability into account. While narrow peaks resulting from Transcription Factor ChIP-Seq can be detected using a fixed window size, broad peak detection has to cope with the additional boundary-problem in the sense that the distance between start and end depends on the regions of the underlying genes.

Some existing approaches [Heinz et al., 2010] first detect narrow peaks using a fixed window size, and then merge close peaks in order to avoid the computational cost of finding regions of variable

length. Nevertheless, different histone marks can also exhibit distinct shapes across gene bodies [Li et al., 2007], which renders them amenable to a shape-based detection algorithms.

By using existing annotations, the Histone ChIP-Seq tool is able to classify gene regions according to the peak shape and thereby provides a good practical trade-off between computational complexity and biological sensitivity. The primary application areas are the analysis of ChIP-Seq data for diverse histone-modifications such as (mono-, di-, and tri-) methylation, acetylation, ubiquitination, etc., in combination with a set of annotated gene regions. The tool is well suited to analyze data from organisms with available gene annotations, while finding peaks in intergenic regions can be accomplished with the Transcription Factor ChIP-Seq tool.

To run the Histone ChIP-Seq tool:

Tools | Epigenomics Analysis (🔍) | Histone ChIP-Seq (📊)

In the first wizard window, select the mapped ChIP-Seq reads as input data (figure 36.1). Multiple inputs (such as replicate experiments) are accepted, provided that they refer to the same genome. It is also possible to work in batch (see section 12.3).

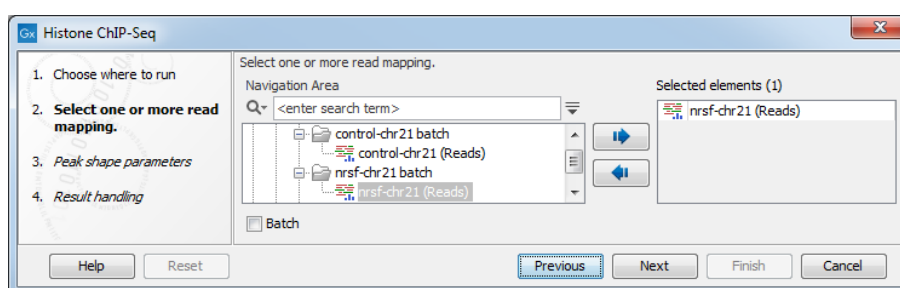


Figure 36.1: Selecting input tracks for the Histone ChIP-Seq tool.

In the second step (figure 36.2), the gene track and control data are defined, along with the p-value. This value defines which regions have a significant fit with the peak-shape, and only these are copied to the output track.

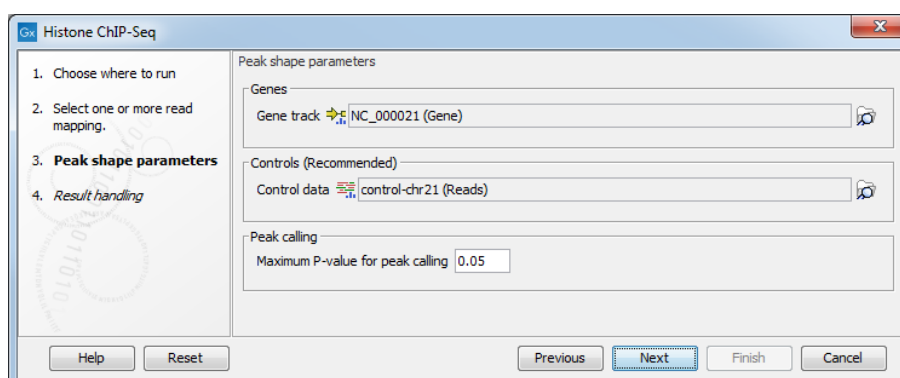


Figure 36.2: Setting peak shape parameters.

The output options are shown in figure 36.3.

- **Create a Quality Control (QC) report** with which you can check the quality of the reads. It lists the number of mapped reads, the normalized strand coefficient, and the relative

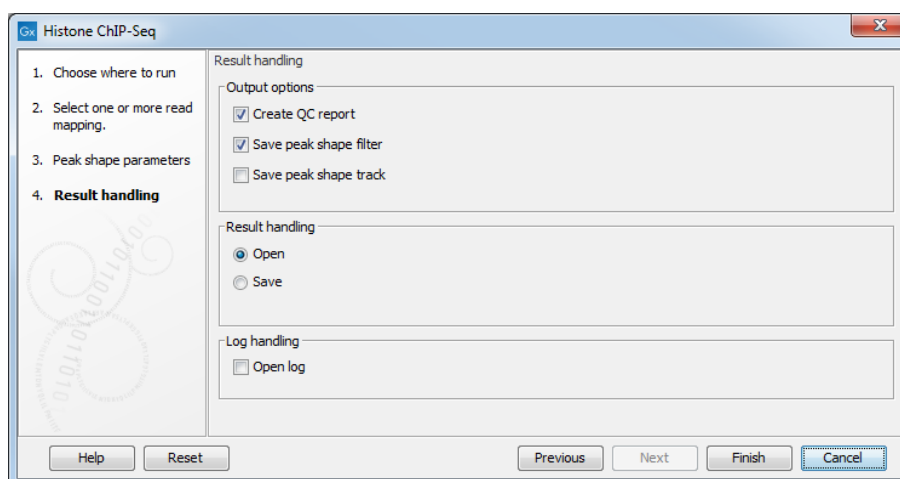


Figure 36.3: Setting up result handling.

strand correlation for each mapping. For each metric, the **Status** column will be **OK** if the experiment has good quality or **Low** if the metric is not as high as expected. Furthermore, the QC report will show the mean read length, the inferred fragment length, and the window size that the algorithm would need to be able to model the signal shape. In case the input contains paired-end reads, the report will also contain the empirical fragment length distribution.

- **Save the peak-shape filter** generated by the tool while processing. This filter can be used to identify genomic regions whose read coverage profile matches the characteristic peak shape, as well as to determine the statistical significance of this match. The filter implemented is called Hotelling Observer and was chosen because it is the matched filter that maximizes the AUCROC (Area Under the Curve of the Receiver Operator Characteristic), one of the most widely used measures for algorithmic performance. For a more detailed description of peak-shape filters, see section 36.2.2, or the white-paper explaining the algorithmic and statistical methods <https://digitalinsights.qiagen.com/files/whitepapers/whitepaper-chip-seq-analysis.pdf>. The peak-shape filter is then applied to the experimental data by scaling the coverage profile in every gene region to a unit-window. The score is obtained for each region by comparing this profile to the peak shape filter.
- **Save peak shape score graph track** (figure 36.4)

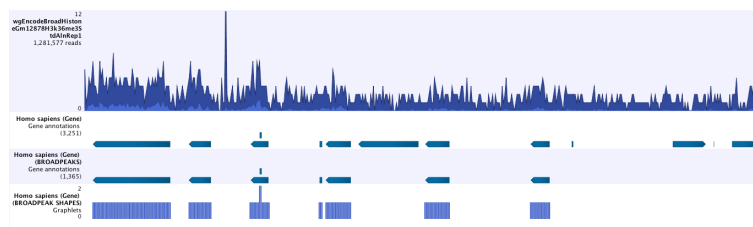


Figure 36.4: Example of Histone ChIP-Seq output.

The peak shape score is standardized and follows a standard normal distribution, so a p-value for each regions is calculated. After the peak shape score for all regions is calculated, regions where the peak shape score is greater than the given threshold are

copied to the output track. Hence the output only contains the gene regions where the coverage graph does match the peak-shape.

36.2 ChIP-Seq Analysis

Chromatin immunoprecipitation followed by high-throughput sequencing (ChIP-Seq) is a method for analyzing DNA-protein interactions. Peak finding is an essential post-processing step in the analysis and interpretation of ChIP-Seq data and many tools that have been developed specifically for this purpose.

Most peak callers are difficult to parametrize such that they accurately discriminate active and inactive promoter regions [Rye et al., 2011, Heydarian et al., 2014]. Nevertheless, these peaks are clearly visible to the human eye since they show a distinct shape. The ChIP-Seq Analysis tool takes a different approach, which is conceptually intuitive and modular by learning the shape of the signal from the data. The parametrization is done by giving positive and negative examples of peak shapes, making the parametrization process explicit and easily understandable.

The ChIP-Seq Analysis tool uses this approach to identify genomic regions with significantly enriched read coverage and a read distribution with a characteristic shape.

ChIP-Seq data analysis is typically based on identification of genomic regions where the signal (i.e. number of mapped reads) is significantly enriched. The detection of the enrichment is based on a background model or the comparison with a ChIP-Seq sample where the immunoprecipitation step is omitted. The shape of the signal from ChIP-Seq data depends on which protein was targeted in the immunoprecipitation reaction [Stanton et al., 2013, Kumar et al., 2013]. For example, the typical signal shape of a transcription factor binding site like NRSF shows a high concentration of forward reads followed by a high concentration of reverse reads (figure 36.5).

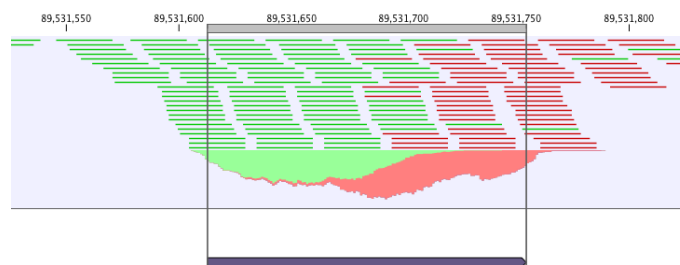


Figure 36.5: *Distribution of forward (green) and reverse (red) reads around a binding site of the transcription factor NRSF.*

The tool makes use of this characteristic shape to identify enriched regions (peaks) in ChIP-Seq data.

36.2.1 Quality Control of ChIP-Seq data

During the first step of the analysis, the ChIP-Seq Analysis tool analyzes the input to check if the input data satisfy the assumptions made by the algorithm and to compute several quality measures. The cross-correlation between reads mapping in the forward and in the reverse strand is often used to investigate the quality of ChIP-Seq experiments [Landt et al., 2012, Marinov et al., 2014]. The quality is determined with respect to the two main peaks of the cross-correlation plot (figure 36.6), the peak at the read length (often called a phantom peak [Landt et al., 2012]) and

the one at the fragment length. The peak at the fragment length is typically higher than the peak at the read length and the background (figure 36.6) for successful ChIP-Seq experiments.

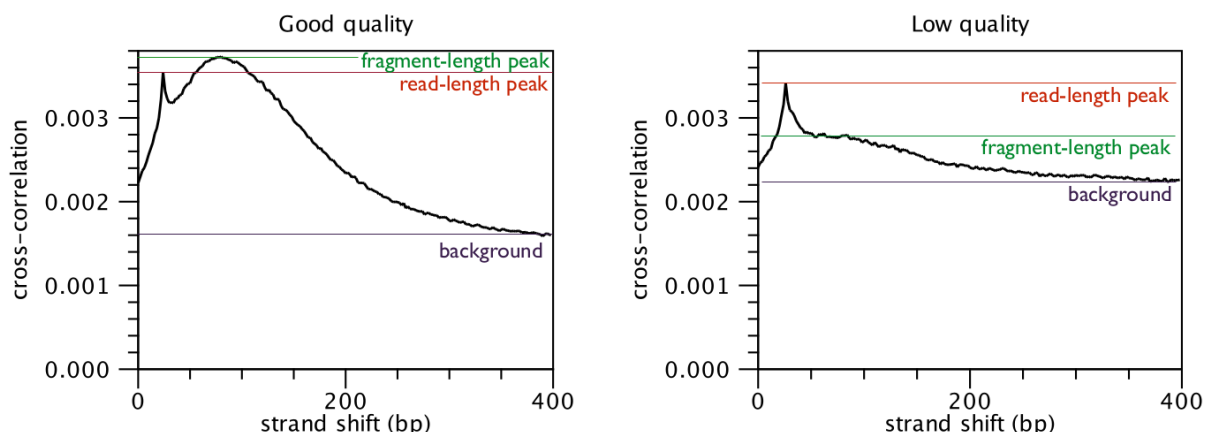


Figure 36.6: Difference in cross-correlation profiles in ChIP experiments of good and low quality.

For each input file of the analysis, the ChIP-Seq Analysis tool calculates and reports several quality measures. Those quality measures have been investigated by the modENCODE consortium and are described in more detail in [Landt et al., 2012]. The quality measures are:

Number of mapped reads For mammalian cells (e.g. human and mouse), this value should be at least 10 million reads. For smaller organisms such as worm and fly, this value should be at least 2 million reads.

Normalized strand coefficient The normalized strand coefficient describes the ratio between the fragment-length peak and the background cross-correlation values. This value should be greater than 1.05 for ChIP-Seq experiments.

Relative strand correlation The relative strand correlation describes the ratio between the fragment-length peak and the read-length peak in the cross-correlation plot. This value should be high (at least 0.8) for transcription factor binding sites, which have a concentrated signal. However, this value can be low even for successful ChIP-Seq experiments on histone modifications [Landt et al., 2012].

36.2.2 Learning peak shapes

In order to learn a characteristic shape from ChIP-Seq data, the ChIP-Seq Analysis tool analyzes the genomic coverage of the reads. For each read mapping, the 5' position of the reads mapping in the forward strand and the 3' position of the reads mapping in the reverse strand are extracted. Those values are then normalized to create a coverage value for the forward and the reverse strand. In order to learn the characteristic peak shape of ChIP-Seq data, the ChIP-Seq Analysis tool identifies a set of positive regions, i.e. regions with very apparent peaks. Those regions are easy to find and are typically found by every peak-caller. The ChIP-Seq Analysis tool identifies these regions by finding areas with very high coverage in the ChIP-Seq data. The average shape of the positive regions of the NRSF transcription factor is shown in for the forward and reverse strand in figure 36.7.

Next, the ChIP-Seq Analysis tool builds a filter, which can be used to identify genomic regions whose read coverage profile matches the characteristic peak shape and to determine the

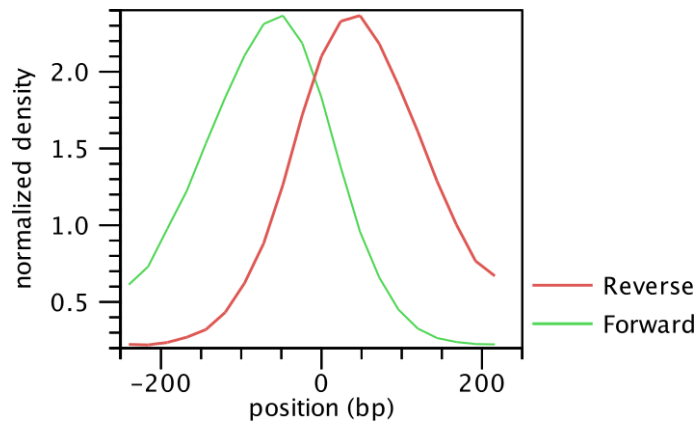


Figure 36.7: Average peak shape of the transcription factor NRSF.

statistical significance of this match. In order to build such a filter, examples of positive (e.g. ChIP-Seq peaks) and negative (e.g. background noise, PCR artifacts) profiles are needed as input. The ChIP-Seq Analysis tool uses regions with very high coverage in the experiment ChIP-Seq as positive examples. If control ChIP-Seq experiments are given, regions with high coverage in the control and low in the experimental ChIP-Seq data are used as negative examples, as they are probably originated from PCR artifacts. If there is no information to build a negative profile from, the profile is estimated from the sequencing noise.

Once the positive and negative regions have been identified, the ChIP-Seq Analysis tool learns a filter that matches the average peak shape, which we term peak shape filter. The filter implemented is called Hotelling Observer and was chosen because it is the matched filter that maximizes the AUCROC (Area Under the Curve of the Receiver Operator Characteristic), one of the most widely used measures for algorithmic performance.

The Hotelling observer h is defined as:

$$h = \left(\frac{R_p + R_n}{2} \right)^{-1} (\mathbb{E}[X_p] - \mathbb{E}[X_n]),$$

where $\mathbb{E}[X_p]$ is the average profile of the *positive* regions, $\mathbb{E}[X_n]$ is the average profile of the *negative* regions, while R_p and R_n denote the covariance matrices between the *positive* and *negative* profiles, respectively. The Hotelling Observer has already previously been successfully used for calling ChIP-Seq peaks [Kumar et al., 2013]. An example of Hotelling observer is shown in figure 36.8.

36.2.3 Applying peak shape filters to call peaks

The peak shape filter is then applied to the experimental data and a score is calculated at each genomic position. The score is obtained by extracting the genomic coverage profile of a window centered at the genomic position and then comparing this profile to the peak shape filter. The result of this comparison is defined as peak shape score. The peak shape score is standardized and follows a standard normal distribution, so a p-value for each genomic position can be calculated as $\text{p-value} = \Phi(-\text{Peak shape score of the peak center})$, where Φ is the standard normal cumulative distribution function.

Once the peak shape score for the complete genome is calculated, peaks are identified as genomic regions where the maximum peak shape score is greater than a given threshold. The

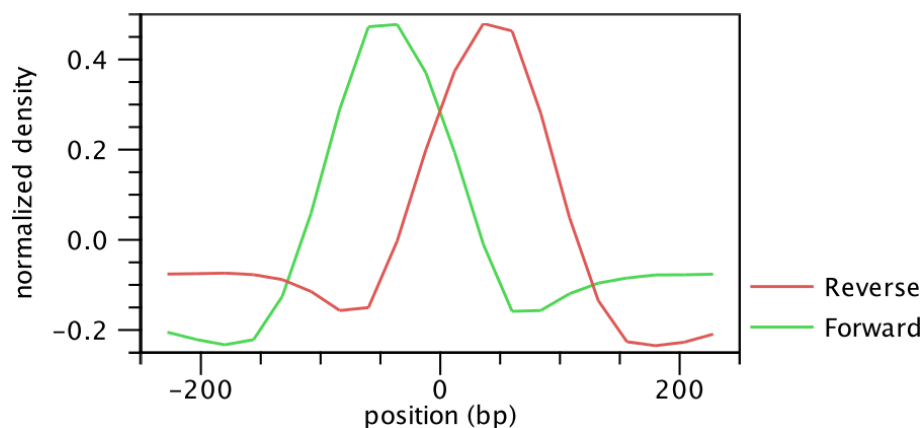







Figure 36.8: Peak shape filter for the transcription factor NRSF.

center of the peak is then identified as the genomic region with the highest peak shape score and the boundaries are determined by the genomic positions where the peak shape score becomes negative.

36.2.4 Running the Transcription Factor ChIP-Seq tool

To run the Transcription Factor ChIP-Seq tool:

Tools | Epigenomics Analysis  | **Transcription Factor ChIP-Seq** 

This will open up the wizard shown in figure 36.9 where you can select the input data (the mapped ChIP-Seq reads). Multiple inputs (e.g. replicate experiments) are accepted, provided that they refer to the same genome. Track based read mappings  and stand-alone read mappings  /  are both accepted.

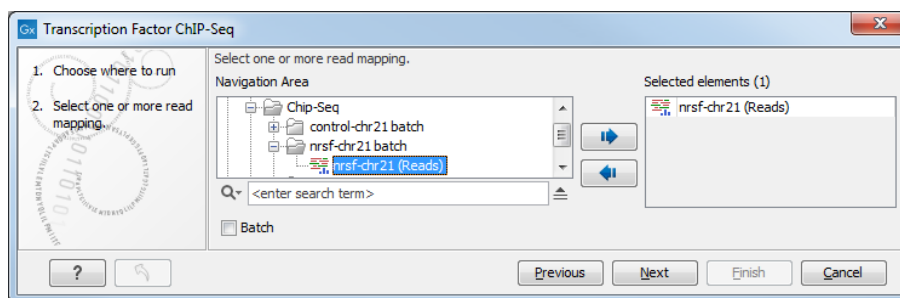


Figure 36.9: Select the input data for the Transcription Factor ChIP-Seq tool.

Click **Next** to go to the next wizard step (shown in figure 36.10).

In this wizard step you have the following options:

- **Control data** The control data, typically a ChIP-Seq sample where the immunoprecipitation step is omitted, can be specified in this option.
- **Maximum P-value for peak calling** The threshold for reporting peaks can be specified by this option.

Click **Next** to go to the wizard step shown in figure 36.11.

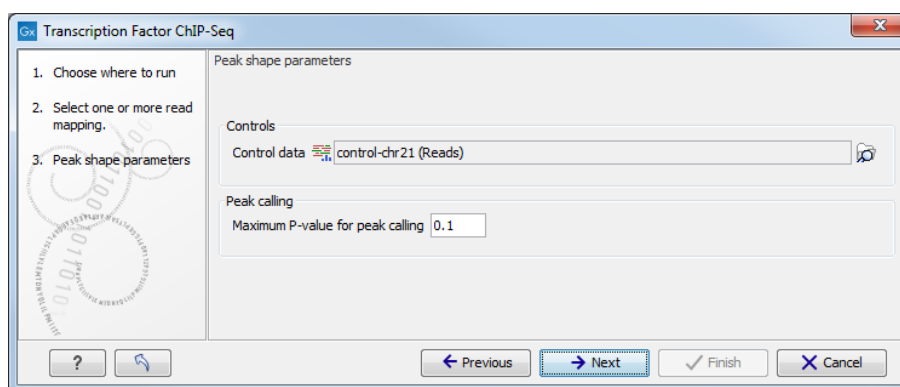


Figure 36.10: Options for the Transcription Factor ChIP-Seq tool.

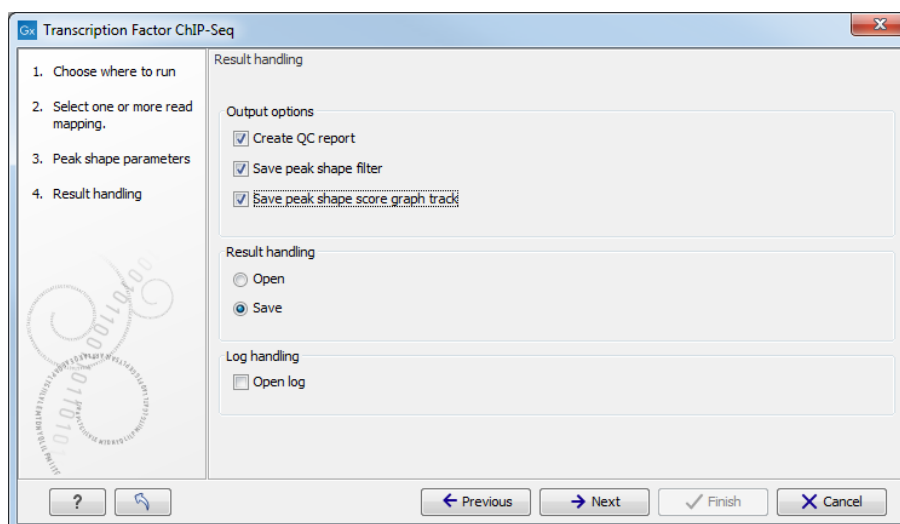






Figure 36.11: Output options for the Transcription Factor ChIP-Seq tool.

In addition to the annotation track with **Peak annotations**  that will always be generated by the algorithm, you can choose to select additional output types.


The options are:

- **QC report**  Generates a quality control report that allows you to check the quality of the reads. The QC report contains metrics about the quality of the ChIP-Seq experiment. It lists the number of mapped reads, the normalized strand coefficient, and the relative strand correlation for each mapping. For each metric, the **Status** column will be **OK** if the experiment has good quality or **Low** if the metric is not as high as expected. Furthermore, the QC report will show the mean read length, the inferred fragment length, and the window size that the algorithm would need to be able to model the signal shape. In case the input contains paired-end reads, the report will also contain the empirical fragment length distribution. The metrics and their definitions are described in more detail in section 36.2.1.
- **Peak shape filter**  The peak shape filter contains the Hotelling Observer filter that was learned by the Transcription Factor ChIP-Seq algorithm. For the definition of Peak shape, see section 36.2.3.

- **Peak shape score** () A graph track containing the peak shape score. The track shows the peak shape score for each genomic position. To save disk space, only peak shape scores greater than zero are reported. For the definition of peak shape score, see section 36.2.3.

Choose whether you want to open the results directly, or save the results in the **Navigation Area**. If you choose to save the results, you will be asked to specify where you would like to save them.

36.2.5 Peak track

The main output of the Transcription Factor ChIP-Seq tool is an annotation track containing the peaks. The table view (figure 36.12) can be opened by clicking on the table icon () in the lower left corner of the peak annotation track. For each peak, it contains:

- **Chromosome** The chromosome where the peak is located.
- **Region** The position of the peak.
- **Center of peak** The center position of the peak. This is determined as the genomic position that matches the peak shape filter best.
- **Length** The length of the peak.
- **Peak shape score** The peak shape score of the peak.
- **P-value** The p-value of the peak.

For more details on some of the values above, see section 36.2.3. Information about the genes located upstream and downstream of the peaks can be added to the table by using Annotate with Nearby Information, see section 27.5.2.

The peak annotation track is most informative when combined with the read mapping in a track list (figure 36.12), see section 27.2 for details.

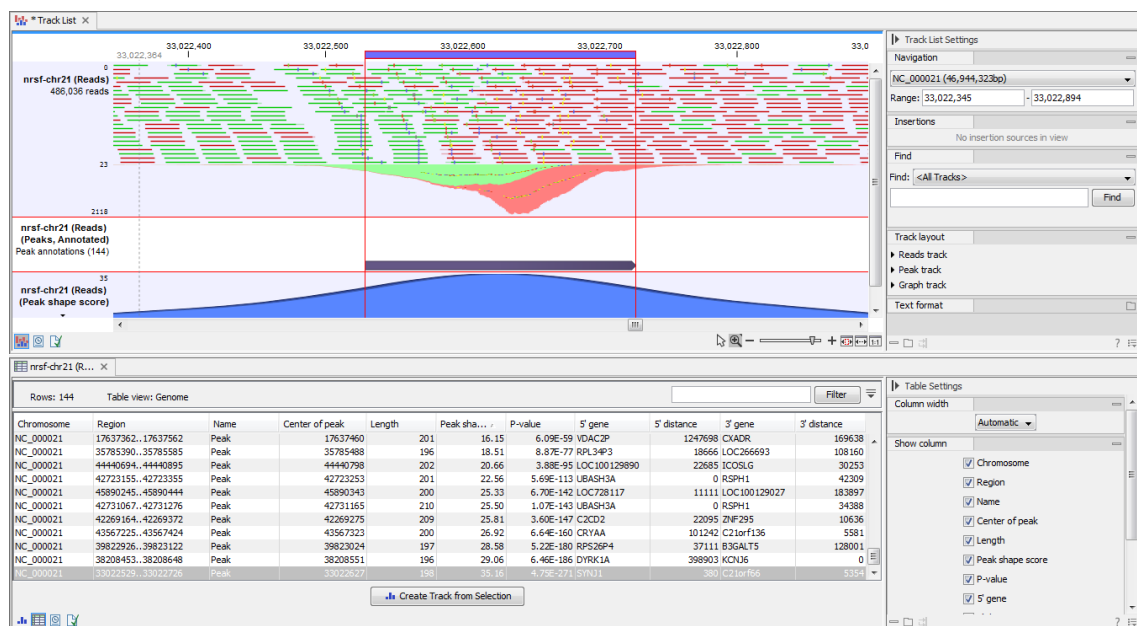


Figure 36.12: Top: Track list containing the mapped reads, the Peak track annotated with nearby genes, and the Peak shape score track. Bottom: Table view of the Peak track. Clicking a peak in the table will update the track list to show the selected peak.

36.3 Bisulfite Sequencing

36.3.1 Detecting DNA methylation

DNA-Methylation is one of the most significant epigenetic mechanisms for cell-programming. DNA-methylation alters the gene expression pattern such that cells can recall their cell type, essentially removing the necessity for continuous external signalling or stimulation. Even more, DNA-methylation is retained throughout the cell-cycle and thus inherited through cell division. DNA-Methylation involves the addition of a methyl group to the 5-position of the cytosine pyrimidine ring or the number 6 nitrogen of the adenine purine ring. DNA-methylation at the 5-position of cytosines typically occurs in a CpG dinucleotide context. CpG dinucleotides are often grouped in clusters called CpG islands, which are present in the 5' regulatory regions of many genes.

A large body of evidence has demonstrated that aberrant DNA-methylation is associated with unscheduled gene silencing, resulting in a broad range of human malignancies. Aberrant DNA-methylation manifests itself in two distinct forms: hypomethylation (a loss/reduction of methylation) and hypermethylation (an increase in methylation) compared to normal tissue. Hypermethylation has been found to play significant roles in carcinogenesis by repressing transcription of tumor suppressor genes, while hypomethylation is implicated in the development and the progression of cancer.

DNA methylation detection methods comprise various techniques. Early methods were based on restriction enzymes, cleaving either methylated or unmethylated CpG dinucleotides. These were laborious but suitable to interrogate the DNA-methylation status of individual DNA sites. It was later discovered that bisulfite treatment of DNA turns unmethylated cytosines into uracils while leaving methylated cytosines unaffected (figure 36.13). This discovery provided a more effective screening method that, in conjunction with whole genome shotgun sequencing of bisulfite converted DNA, opened up a broad field of genome-wide DNA methylation studies.

Ever since, bisulfite shotgun sequencing (BS-seq) is considered a gold-standard technology for genome-wide methylation profiling at base-pair resolution.

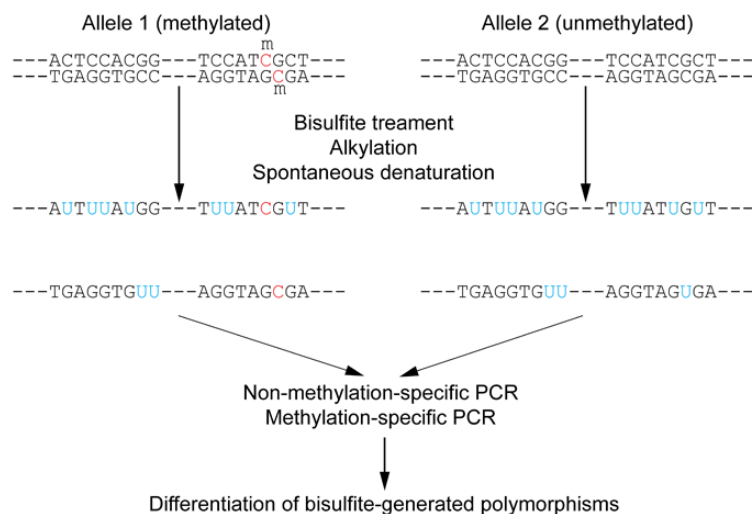


Figure 36.13: Outline of bisulfite conversion of sample sequence of genomic DNA. Nucleotides in blue are unmethylated cytosines converted to uracils by bisulfite, while red nucleotides are 5-methylcytosines resistant to conversion. Source: https://en.wikipedia.org/wiki/Bisulfite_sequencing

Figure 36.14 depicts the conceptual steps of the Bisulfite-sequencing:

1. **Genomic DNA** Genomic DNA is extracted from cells, sheared to fragments, end-repaired, size-selected (around 400 base pairs depending on targeted read length) and ligated with Illumina methylated sequencing adapters. End-repair involves either methylated or unmethylated cytosines, possibly skewing true methylation levels. Therefore, 3'- and 5'-ends of sequenced fragments should be soft-clipped prior to assessing methylation levels.
2. **Denaturation** Fragments must be denatured (and kept denatured during bisulfite conversion), because bisulfite can only convert single-stranded DNA.
3. **Bisulfite conversion** Bisulfite converts unmethylated cytosines into uracils, but leaves methylated cytosines unchanged. Because bisulfite conversion has degrading effects on the sample DNA, the conversion duration is kept as short as possible, sometimes resulting in non-complete conversions (*i.e.* not all unmethylated cytosines are converted).
4. **PCR amplification** PCR-amplification reconstructs the complementary strands of the converted single-stranded fragments and turns uracils into thymines.
5. **Strand discordance** Not an actual step of the workflow, but to illustrate that bisulfite converted single-stranded fragments are not reverse-complementary anymore after conversion.
6. **Paired-end sequencing** Directional paired-end sequencing yields read pairs from both strands of the original sample-DNA. The first read of a pair is known to be sequenced either from the original-top (OT) or the original-bottom (OB) strand. The second read of a pair is sequenced from a complementary strand, either ctOT or ctOB. It is a common

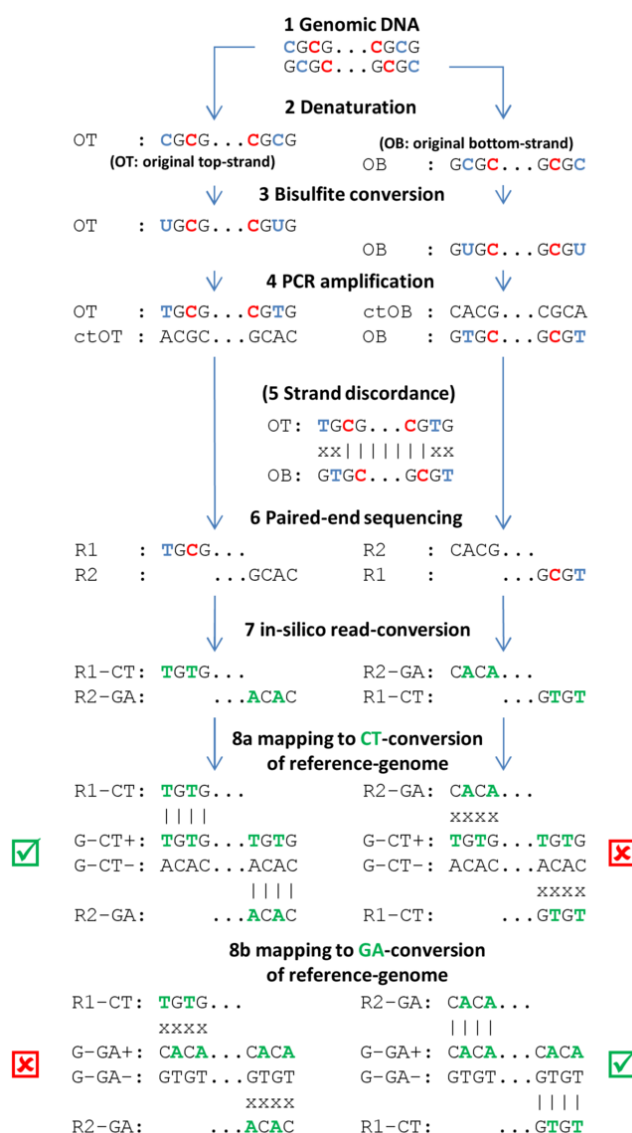


Figure 36.14: Individual steps of BS-seq workflow include denaturation of fragmented sample DNA, bisulfite conversion, subsequent amplification, sequencing and mapping of resulting DNA-fragments. (See text for explanations). Methylated cytosines are drawn in red, unmethylated cytosines and respective uracils/thymidines in blue. DNA-nucleotides that are in-silico converted (during read mapping) are given in green.

misunderstanding that the first read of a pair yields methylation information for the top-strand and the second read for the bottom-strand (or vice versa). Rather, both reads of a read pair report methylation for the same strand of sample DNA, either the top or the bottom strand. Individual read pairs can of course arise from both the top and the bottom strand, eventually yielding information for both strands of the sample DNA.

- 7. In silico read-conversion** The only bias-free mapping approach for BS-seq reads involves *in-silico* conversion of all reads. All cytosines within all first reads of a pair are converted to thymines and all guanines in all second reads of a pair are converted to adenines (complementary to C-T conversion).

8. **Mapping to CT- or GA-conversion of reference genome** The reference genome is also converted into two different *in silico* versions. In the first conversion all cytosines are replaced by thymines and, in the second conversion, all guanines are converted to adenines. The *in silico* converted read pairs are then independently mapped to the two conversions of the reference genome and the better of the two mappings is reported as final mapping result (see green checkboxes).

Note: with non-directional BS-seq, no assumptions regarding the strand origins of either of the reads of a pair can be made (see step 6). Therefore, two different conversions of the read pair need to be created: the first read of a converted pair consists of the CT-conversion of read 1 and the GA-conversion of read 2, and the second converted pair consists of the GA-conversion of read 1 and the CT-conversion of read 2. Both converted reads pairs are subsequently mapped to the two conversion of the reference genome. The best out of the four resulting mappings is then reported as the final mapping result.

36.3.2 Map Bisulfite Reads to Reference

Setting up the **Map Bisulfite Reads to Reference** tool is very similar to the usual read mapping, with a few differences. In particular,

- Some options such as 'Global alignment' are either not available or preset.
- Only a track version of mappings is produced in output.
- The bisulfite mappings have a special 'invisible' property set for them, to inform the downstream Call Methylation Levels tool, see section 36.3.3, about the correct type of input.

Please note that, because two versions of the reference sequence (C->T and G->A - converted) have to be indexed and used simultaneously for each read, the RAM requirements for the bisulfite mapper are double than those needed for a regular mapping against a reference sequence of the same size.

To start the read mapping:

Tools | Epigenomics Analysis  | Bisulfite Sequencing  | Map Bisulfite Reads to Reference 

Selecting reads and directionality

In the first dialog, select the sequences or sequence lists containing the sequencing data (figure 36.15). Please note that reads longer than 100,000 bases are not supported.

When the sequences are selected, click **Next** to specify what type of protocol you have used (directional or not).

A directional protocol yields reads from both strands of the original sample-DNA. The first read of a pair (or every read for single-end sequencing) is known to be sequenced either from the original-top (OT) or the original-bottom (OB) strand. The second read of a pair is sequenced from a complementary strand, either ctOT or ctOB. At the time of writing, examples of directional protocols include:

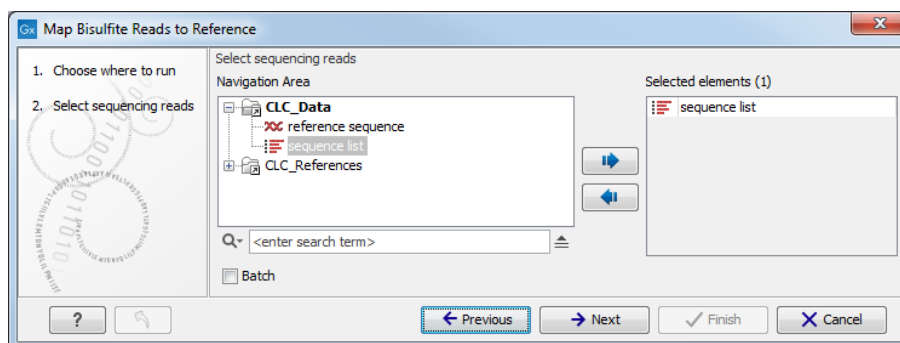


Figure 36.15: *Specifying the reads as input. You can also choose to work in batch.*

- Illumina TruSeq DNA Methylation Kit (formerly EpiGnome)
- Kits from the NuGen Ovation family of products
- Swift Accel-NGS Methyl-seq DNA Library Kit
- Libraries prepared by the 'Lister' method

In a non-directional protocol, the first read of a pair may come from any of the four strands: OT, OB, ctOT or ctOB. Examples include:

- QIAseq Methyl Library Kit <https://www.qiagen.com/us/shop/sequencing/qiaseq-solutions/qiaseq-methyl-library-kit>
- Zymo Pico Methyl-Seq Library Kit
- Bioo Scientific (Perkin Elmer) NEXTflex Bisulfite-Seq Kit
- Libraries prepared by the 'Cokus' method

If you are uncertain about the directionality of your protocol, contact the protocol vendor. Note that it is sometimes possible to infer the directionality by looking at the reads: in the absence of methylation, a directional protocol will have few or no Cs in the first read of each pair. We do not recommend however using this approach.

Selecting the reference

When the sequences and directionality are selected, click **Next**, and you will see the dialog shown in figure 36.16.

Click the **Browse and select element** (🔍) button to select either single sequences, a list of sequences or a sequence track as reference. Note the following constraints:

- single reference sequences longer than 2gb ($2 \cdot 10^9$ bases) are not supported.
- a maximum of 120 input items (sequence lists or sequence elements) can be used as input to a single read mapping run.

Including or excluding regions (masking)

The next part of the dialog shown in figure 36.16 lets you *mask* the reference sequences. Masking refers to a mechanism where parts of the reference sequence are not considered in the mapping. This can be useful for example when mapping data is captured from specific

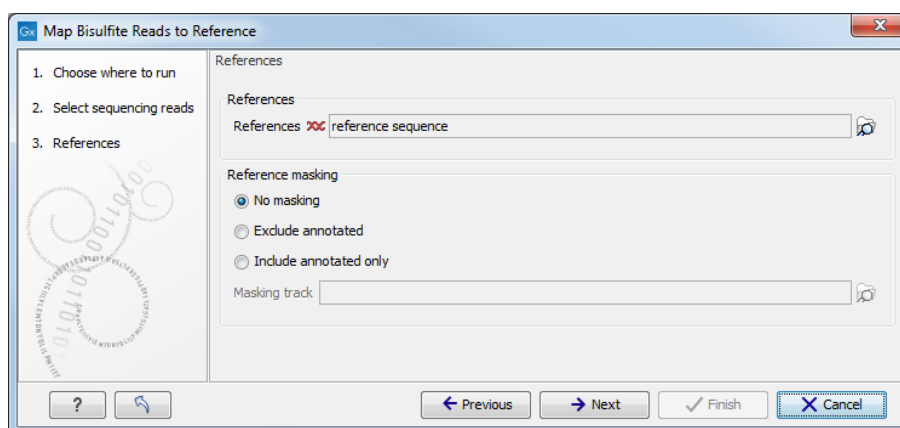


Figure 36.16: Specifying the reference sequences and masking.

regions (e.g. for amplicon resequencing). The read mapping will still base its output on the full reference - it is only the core read mapping that ignores regions.

Masking is performed by discarding the masked out nucleotides. As a result the reference is split into separate sequences, which are positioned according to the original unmasked reference sequence.

Note that you should be careful that your data is indeed only sequenced from the target regions. If not, some of the reads that would have matched a masked-out region perfectly may be placed wrongly at another position with a less-perfect match and lead to wrong results for subsequent variant calling. For resequencing purposes, we recommend testing whether masking is appropriate by running the same data set through two rounds of read mapping and variant calling: one with masking and one without. At the end, comparing the results will reveal if any off-target sequences cause problems in the variant calling.

Masking out repeats or using other masks with many regions is not recommended. Repeats are handled well and does not cause any slowdown. On the contrary, masking repeats is likely to cause a dramatic slowdown in speed, increase memory requirements and lead to incorrect read placement.

To mask a reference sequence, first click the **Include** or **Exclude** options, and second click the **Browse** (🔍) button to select a track to use for masking.

Mapping parameters

Clicking **Next** leads to the parameters for the read mapping (see figure 36.17).

The first parameter allows the mismatch cost to be adjusted:

- **Match score** The positive score for a match between the read and the reference sequence. It is set by default to 1 but can be adjusted up to 10.
- **Mismatch cost** The cost of a mismatch between the read and the reference sequence. Ambiguous nucleotides such as "N", "R" or "Y" in read or reference sequences are treated as a mismatches and any column with one of these symbols will therefore be penalized with the mismatch cost.

After setting the mismatch cost you need to choose between linear gap cost and affine gap cost, and depending on the model you chose, you need to set two different sets of

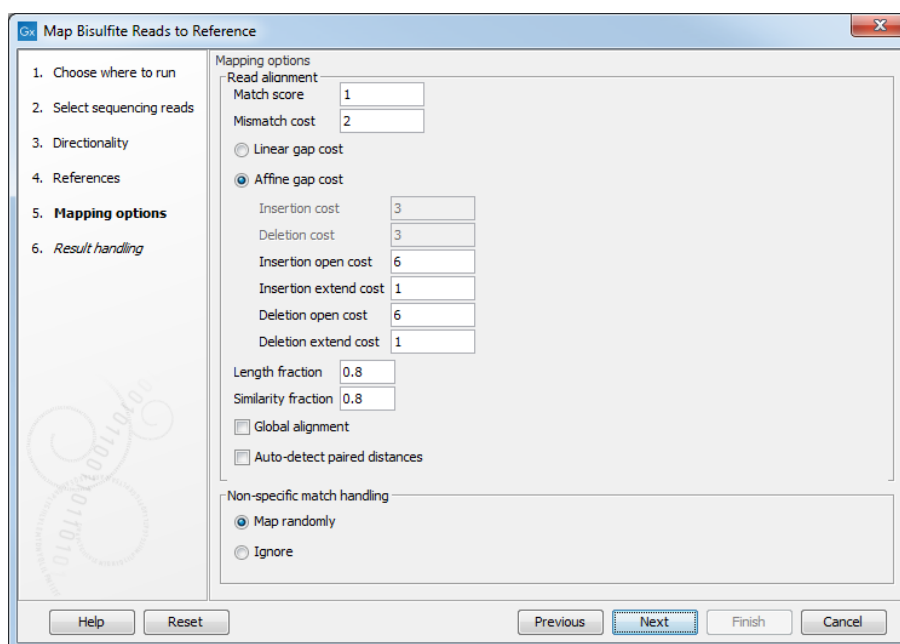


Figure 36.17: Setting parameters for the bisulfite read mapping.

parameters that control how gaps in the read mapping are penalized.

- **Linear gap cost** The cost of a gap is computed directly from the length of the gap and the insertion or deletion cost. This model often favors small, fragmented gaps over long contiguous gaps. If you choose linear gap cost, you must set the insertion cost and the deletion cost:

Insertion cost The cost of an insertion in the read (a gap in the reference sequence). The cost of an insertion of length ℓ will be $\ell \cdot \text{Insertion cost}$.

Deletion cost The cost of a deletion in the read (gap in the read sequence). The cost of a deletion of length ℓ will be $\ell \cdot \text{Deletion cost}$.

- **Affine gap cost** An extra cost associated with opening a gap is introduced such that long contiguous gaps are favored over short gaps. If you chose affine gap cost, you must also set the cost of opening an insertion or a deletion:

Insertion open cost The cost of opening an insertion in the read (a gap in the reference sequence).

Insertion extend cost The cost of extending an insertion in the read (a gap in the reference sequence) by one column.

Deletion open cost The cost of a opening a deletion in the read (gap in the read sequence).

Deletion extend cost The cost of extending a deletion in the read (gap in the read sequence) by one column.

Using affine gap cost, an insertion of length ℓ is penalized by
 $\text{Insertion open cost} + \ell \cdot \text{Insertion extend cost}$
 and a deletion of length ℓ is penalized by
 $\text{Deletion open cost} + \ell \cdot \text{Deletion extend cost}$

In this way long consecutive gaps get a lower cost per column than small fragmented gaps and they are therefore favored.

The score of a match between the read and the reference is usually set to 1. Adjusting the cost parameters above can improve the mapping quality, e.g. when the read error rate is high or the reference is expected to differ significantly from the sequenced organism. For example, if the reads are expected to contain many insertions and/or deletions, it can be a good idea to lower the insertion and deletion costs to allow more of such errors. However, one should also consider the possible drawbacks when adjusting these settings. For example, reducing the insertion and deletion cost increases the risk of mapping reads to the wrong positions in the reference.

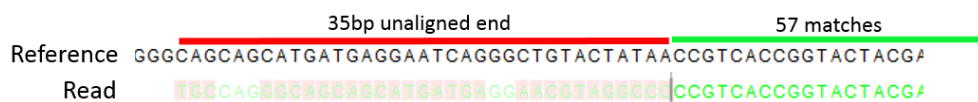


Figure 36.18: An alignment of a read where a region of 35bp at the start of the read is unaligned while the remaining 57 nucleotides matches the reference.

Figure 36.18 shows an example using linear gap cost where the read mapper is unable to map a region in a read due to insertions in the read and mismatches between the read and the reference. The aligned region of the read has a total of 57 matching nucleotides which result in an alignment score of 57 which is optimal when using the default cost for mismatches and insertions/deletions (2 and 3 respectively). If the mapper had aligned the remaining 35bp of the read as shown in figure 36.19 using the default scoring scheme, the score would become: $(26+1+3+57) * 1 - 5 * 2 - 8 * 3 = 53$

In this case, the alignment shown in Figure 36.18 is optimal since it has the highest score. However, if either the cost of deletions or mismatches were reduced by one, the score of the alignment shown in figure 36.19 would become 61 and 58, respectively, and thus make it optimal.

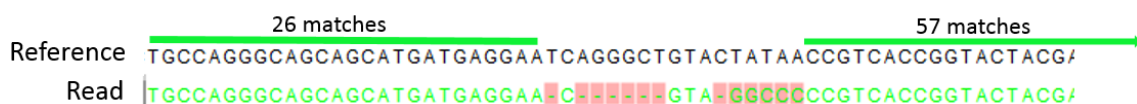


Figure 36.19: An alignment of a read containing a region with several mismatches and deletions. By reducing the default cost of either mismatches or deletions the read mapper can make an alignment that spans the full length of the read.

Once the optimal alignment of the read is found, based on the cost parameters described above, a filtering process determines whether this match is good enough for the read to be included in the output. The filtering threshold is determined by two factors:

- **Length fraction** The minimum percentage of the total alignment length that must match the reference sequence at the selected similarity fraction. A fraction of 0.5 means that at least half of the alignment must match the reference sequence before the read is included in the mapping (if the similarity fraction is set to 1). Note, that the minimal seed (word) size for read mapping is 15 bp, so reads shorter than this will not be mapped.
- **Similarity fraction** The minimum percentage identity between the aligned region of the read and the reference sequence. For example, if the identity should be at least 80% for the read to be included in the mapping, set this value to 0.8. Note that the similarity fraction relates to the length fraction, i.e. when the length fraction is

set to 50% then at least 50% of the alignment must have at least 80% identity (see figure 36.20).

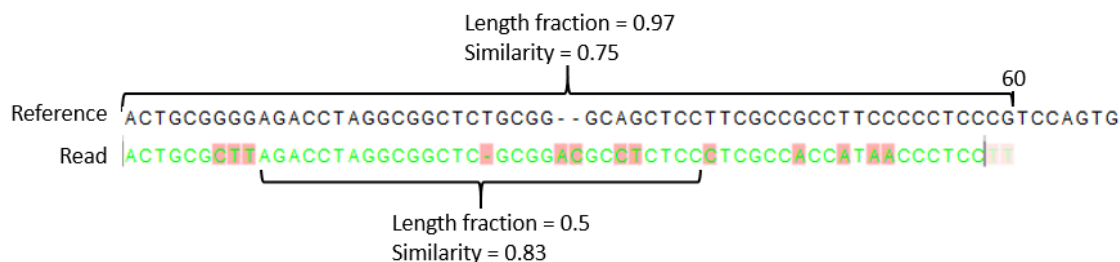


Figure 36.20: A read containing 59 nucleotides where the total alignment length is 60. The part of the alignment that gave rise to the optimal score has length 58 which excludes 2 bases at the left end of the read. The length fraction of the matching region in this example is therefore $58/60 = 0.97$. Given a minimum length fraction of 0.5, the similarity fraction of the alignment is computed as the maximum similarity fraction of any part of the alignment which constitute at least 50% of the total alignment. In this example the marked region in the alignment with length 30 (50% of the alignment length) has a similarity fraction of 0.83 which will satisfy the default minimum similarity fraction requirement of 0.8.

Mapping paired reads

- Global alignment** By default, mapping is done with local alignment of the reads to the reference. The advantage of performing local alignment instead of global alignment is that the ends are automatically left unaligned if there are many differences from the reference at the ends. For many sequencing platforms, the quality of the bases drop along the read, and a local alignment approach is desirable. By checking this option, the mapper is forced to look for the highest scoring alignment of the entire read, meaning that the read mapping generated will have no unaligned ends even when the end of the reads align to the wrong places.
- Auto-detect paired distances** If the sequence list used as input contains paired reads, this option will automatically be enabled - if it contains single reads, this option will not be applicable.

CLC Genomics Workbench offers as the default choice to automatically calculate the distance between the pairs. If this is selected, the distance is estimated in the following way:

1. A sample of 200,000 reads is extracted randomly from the full data set and mapped against the reference using a very wide distance interval.
2. The distribution of distances between the paired reads is analyzed using a method that investigates the shape of the distribution and finds the boundaries of the peak.
3. The full sample is mapped using this distance interval.
4. The **history** (📄) of the result records the distance interval used.

The above procedure will be run for each sequence list used as input, assuming that they do not necessarily share the same library preparation and could have different distributions

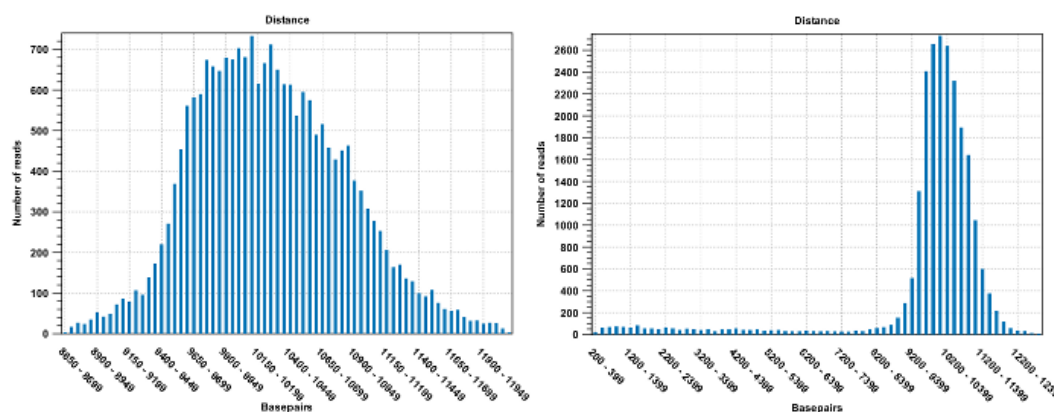


Figure 36.21: To the left: mapping with a narrower distance interval estimated by the workbench. To the right: mapping with a large paired distance interval (note the large right tail of the distribution).

of paired distances. Figure 36.21 shows an example of the distribution of intervals with and without automatic pair distance interval estimation.

Sometimes the automatic estimation of the distance between the pairs may return a warning "Few reads mapped as pairs so pair distance might not be accurate". This message indicates that the paired distance was chosen to spans all uniquely mapped reads. If in doubt, you may want to disable the option to automatically estimate paired distances and instead manually specify minimum and maximum distances between pairs on the input sequence list.

If the automatic detection of paired distances is not checked, the mapper will use the information about minimum and maximum distance recorded on the input sequence lists.

When a paired distance interval is set, the following approach is used for determining the placement of read pairs:

- First, all the optimal placements for the two individual reads are found.
- Then, the allowed placements according to the paired distance interval are found.
- If both reads can be placed independently but no pairs satisfies the paired criteria, the reads are treated as independent and marked as a **broken pair**.
- If only one pair of placements satisfy the criteria, the reads are placed accordingly and marked as uniquely placed even if either read may have multiple optimal placements.
- If several placements satisfy the paired criteria, the pair is treated as a non-specific match (see section 30.1.5 for more information.)
- If one read is uniquely mapped but the other read has several placements that are valid given the distance interval, the mapper chooses the location that is closest to the first read.

Non-specific matches

At the bottom of the dialog, you can specify how **Non-specific matches** should be treated. The concept of Non-specific matches refers to a situation where a read aligns at *more than one position with an equally good score*. In this case you have two options:

- **Random**. This will place the read in one of the positions randomly.

- **Ignore.** This will not include the read in the final mapping.

Note that a read is only considered non-specific when the read matches equally well at several alignment positions. If there are e.g. two possible alignment positions and one of them is a perfect match and the other involves a mismatch, the read is placed at the position with the perfect match and it is not marked as a non-specific match.

For paired data, reads are only considered non-specific matches if the entire pair could be mapped elsewhere with equal scores for both reads, or if the pair is broken in which case a read can be categorized as non-specific in the same way as single reads (see section 30.1.4).

When looking at the mapping, the default color for non-specific matches is yellow.

Gap placement

In the case of insertions or deletions in homopolymeric or repetitive regions, the precise placement of the insertion or deletion cannot be determined from the data. An example is shown in figure 36.22.

```

TTCTCAAACAAT
TTCTC-AACAAT
TTCTC-AACAAT
TTCTC-AACAAT
TTCTC-AACAAT
TTCTC-AACAAT
TTCTC-AACAAT

```

Figure 36.22: Three A's in the reference (top) have been replaced by two A's in the reads (shown in red). The gap is placed towards the 5' end, but could have been placed towards the 3' end with an equally good mapping score for the read.

In this example, three A's in the reference (top) have been replaced by two A's in the reads (shown in red). The gap is placed towards the 5' end (left side), but could have been placed towards the 3' end with an equally good mapping score for the read as shown in figure 36.23.

```

TTCTCAAACAAT
TTCTCAA-CAAT
TTCTCAA-CAAT
TTCTCAA-CAAT
TTCTCAA-CAAT
TTCTCAA-CAAT
TTCTCAA-CAAT

```

Figure 36.23: Three A's in the reference (top) have been replaced by two A's in the reads (shown in red). The gap is placed towards the 3' end, but could have been placed towards the 5' end with an equally good mapping score for the read.

Since either way of placing the gap is arbitrary, the goal of the mapper is to place the gaps consistently at the same side for all reads.

Bisulfite read mapping result handling

Click **Next** lets you choose how the output of the mapping should be reported. There are two independent output options available that can be (de-)activated in both cases:

- **Create report.** This will generate a summary report as described in section 30.1.10.
- **Collect unmapped reads.** This will collect all the reads that could not be mapped to the reference into a sequence list (there will be one list of unmapped reads per sample, and for paired reads, there will be one list for intact pairs and one for single reads where the mate could be mapped).

However, the main output is a reads track:

Reads track A reads track is very "lean" (i.e. with respect to memory requirements) since it only contains the reads themselves. Additional information about the reference, consensus sequence or annotations can be added and viewed alongside in the context of a Track List later (by adding, for example, a reference and/or annotation track, respectively). This kind of output is useful when working with tracks in general and especially for resequencing purposes this is recommended.

Note that the tool will output an empty read mapping and report when nothing mapped, and empty unmapped reads if everything mapped.

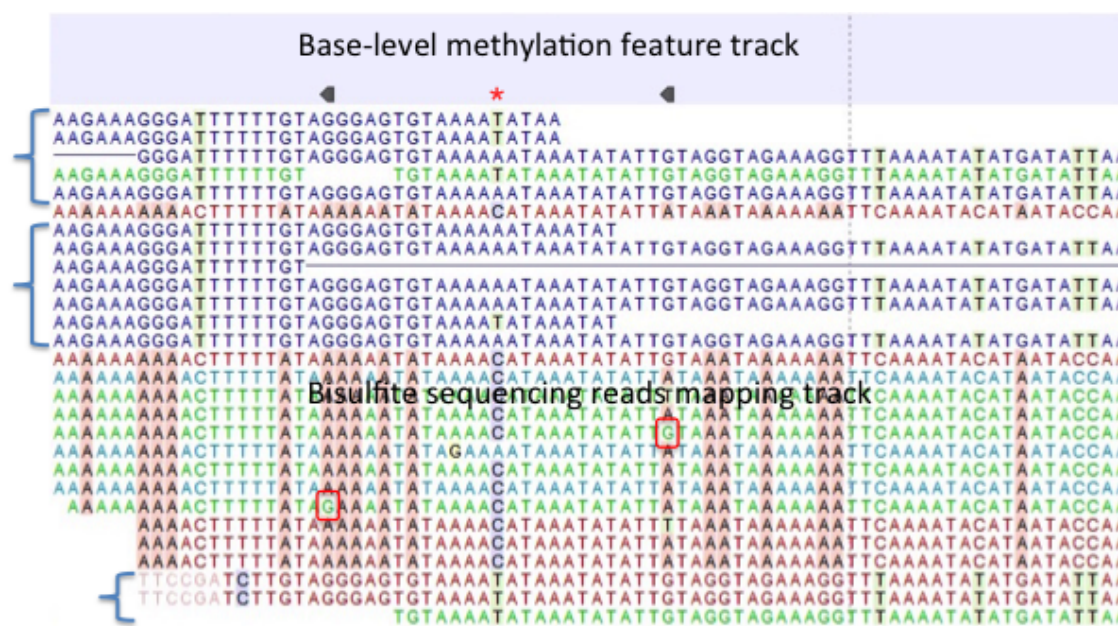


Figure 36.24: A typical directional shotgun BS-seq mapping, together with the base-level methylation calling feature track on top.

Figure 36.24 illustrates the view of a typical directional shotgun BS-seq mapping. As with any read mapping view, the color of the reads follows the usual CLC convention, that is green/red for forward/reverse reads, and dark/pale blue for paired reads. Independent of this orientation property, each read or read pair has an 'invisible' property indicating if it came from the original

top (OT), or original bottom (OB) strand. However, if the BS-sequencing protocol is truly 100%-directional, the orientation in the mapping and the OT/OB origin of reads/read pairs will be concordant.

In this figure, blocks of reads from the original top strand are marked with squiggly brackets on the left, while the rest are from the original bottom strand. In a mapping, they can be distinguished by a pattern of highlighted mismatches to reference. OT reads will have mismatches predominantly on 'T', due to C->T conversion; OB reads will have a pattern of mismatches on 'A' symbols, corresponding to G->A conversion as can be seen on a reverse-complementary strand. When methyl-C occurs in a sample, there will be a match in the reads, instead of an expected mismatch.

In this figure, there were two positions where such events occurred, both on the original bottom strand, and both supported by a single read only. 'G' symbols in those reads are shown in red boxes. The reverse direction of an arrowhead on a base-level methylation track also reflects the OB-position of a methylation event.

Note also that it appears that there may be a G/T heterozygous SNP (C/A in the OB strand) in the second position. While such occurrences may lead to underestimation of true methylation levels of cytosine alleles in heterozygous SNPs, our current tool does not attempt to compensate for such eventualities.

To understand and interpret BS-sequencing and mapping better, it may be helpful to examine the position (marked with a red asterisk) in between the two detected methylation events. There appears to be an additional A/C heterozygous SNP with C's in reads from OT strand fully converted to T's, i.e., showing no evidence of methylation at that heterozygous position.

36.3.3 Call Methylation Levels

The tool takes as input one or more read mappings created by the Map Bisulfite Reads to Reference tool, see section 36.3.2. If more than one mapping is used as input, various statistical options to detect differential methylation become available.

The tool will accept a regular mapping as input, but will warn about possibly inconsistent interpretation of results. Mapping done in a 'normal', not bisulfite mode, is likely to result in sub-optimal placement of reads due to a large number of C/T mismatches between bisulfite-converted reads and a reference. Also, this tool will consequently interpret majority of cytosines in a reference as methylated, creating possibly very large and misleading output files. The invisible 'bisulfite' property of a mapping may be erased if the original mapping is manipulated in the workbench with other tools - such as the **Merge Read Mappings** tool - in which case the warning should be ignored.

After selecting the relevant mapping(s), the wizard offers to set the parameters for base-level methylation calling, as shown in figure 36.25:

- The first three check boxes (**Ignore non-specific matches**, **Ignore duplicate matches**, **Ignore broken pairs**) enable control over whether or not certain reads will be included in base level methylation calling, and subsequent statistical analysis. The recommended option is to have them turned on.
 - **Ignore non-specific matches** Reads or matches mapped ambiguously will be ignored.
 - **Ignore duplicate matches** Multiple reads with identical mapping coordinates will be counted once only.

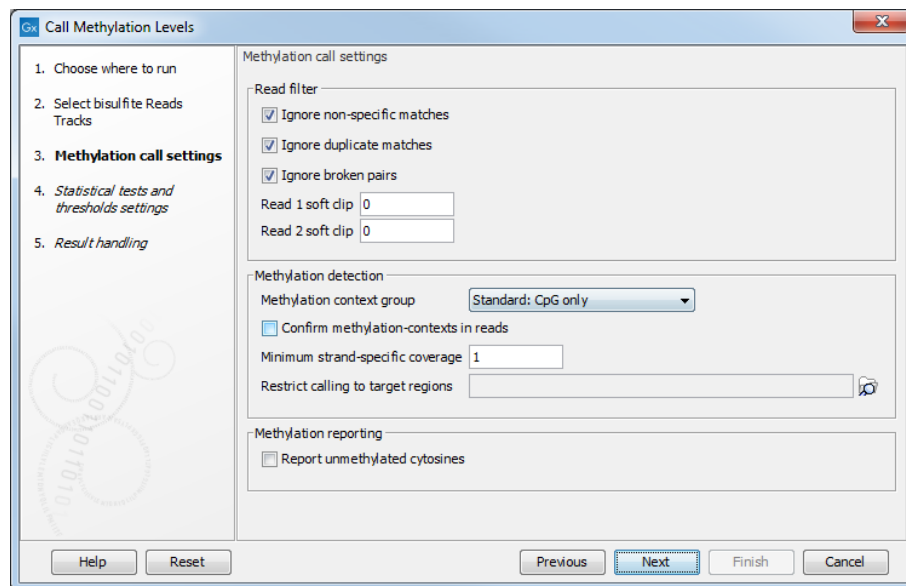


Figure 36.25: Methylation call settings.

- **Ignore broken pairs** Pairs reads mapped as broken pairs will be ignored.
- **Read 1 soft clip, Read 2 soft clip:** sets a number of bases on a 5'-end of a read that will be ignored in methylation calling. It is common for bisulfite data to have a technical bias in amplification and sequencing, making a small number of bases at the beginning of a read (usually not more than 5) unreliable for calling. Setting a parameter to 0 (default), and inspecting a graph in the report may help determine the specific number for a certain dataset, if a bias is suspected.
- **Methylation context group** popup menu controls in which context the calls will be made.
 - Standard contexts include:
 - CpG** Detects 5-methylated cytosines in CpG contexts
 - CHG** Detects 5-methylated cytosines in CHG contexts (H = A/C/T)
 - CHH** Detects 5-methylated cytosines in CHH contexts
 - NOME-seq contexts [Kelly et al., 2012] include:
 - GCH** Detects enzymatic methylation in GCH contexts
 - HCG** Detects endogenous methylation in HCG contexts
 - GCG** Detects ambiguous methylation in GCG contexts
 - Exhaustive
 - Detects 5-methylated cytosines independently of their nucleotide-context
- **Confirm methylation contexts in reads** checkbox controls if a selected context(s) is present in a read itself, and not just a reference sequence, before a call is made. This is useful if a sample has a variant that can affect a context of a call, so that reads representing a variant allele that breaks a context will be excluded, if the box is checked.
- **Minimum strand-specific coverage** sets a lower limit of coverage for the top, or a bottom strand, to filter out positions with low coverage.

- **Restrict calling to target regions** enables selection of a feature track to limit calling to defined regions. In addition to genes, CDSs and other annotation tracks that can be generated or imported into the workbench, Create RRBS-fragment Track tool, see section 36.3.4, can be used to generate fragments of pre-selected size predicted for restriction digest of a reference genome with commonly used frequent cutters that target common methylation contexts, such as MspI.
- **Report unmethylated cytosines** ensures that methylation levels are reported at all sites with coverage, rather than sites with some methylation. Both methylated and unmethylated cytosines will be reported in the optional methylation levels track, while the detection of differentially methylated regions remains unaffected. With this option on, the methylation levels track will include some fully unmethylated cytosines with "methylation level = 0" and "methylated coverage = 0", provided that they have context coverage ≥ 1 .

Statistical tests and thresholds settings

The next set of parameters depends on experimental setup, and a number of samples in the input, as shown in 36.26.

Figure 36.26: The statistical tests and thresholds settings.

Statistical test

- **Statistic mode** pop-up menu offers the following choices.
- **Fisher exact:** Compares methylation-levels of a case/control sample-pair; multiple case/-control samples are merged before pair-wise comparison. Context-specific coverage, and methylated coverage are summed separately for case and control mappings in a window of pre-set size. If more than one case, or control samples are provided, the values within each set are simply added. The contingency table is used to evaluate the hypergeometric cumulative distribution probability for methylated coverage in case sample(s) to be equal or greater than in controls, given the context-specific coverage in a window. Therefore, this test reports statistically significant HYPER-methylation in cases, compared to controls, in a

given window. To identify regions that are hypo-methylated compared to controls with this test, simply reverse case and control when specifying the inputs.

- **Chi-squared:** Analyses the inter-individual methylation-level variability across a cohort of samples; no controls are supported. A contingency table is constructed for a window, where each row corresponds to an input sample, with coverage counted for methylated and unmethylated cytosines within strand/context. Expected values for those, given sample coverage in a window, are calculated from aggregate for all samples, and deviation is evaluated with a Chi-squared test. This statistic tells if an input group of samples has methylation heterogeneity between them, in a given window.
- **ANOVA:** Assesses differential methylation by comparing a case-sample group versus a control-sample group; requires at least two case-samples and two control-samples. It tests if variability in methylation levels within each group is less than between groups, in a window of interest.
- **No test:** No test will be performed and only methylation levels will be produced for each input sample; remaining options on that screen will be grayed out.
- **Maximum p-value** sets the limit of probability calculated in a statistical test of choice, at which a window will be accepted as significant, and included in the output.
- **Control samples** menu is used to select bisulfite mappings that are required to serve as controls in either Fisher exact, or Anova statistics.

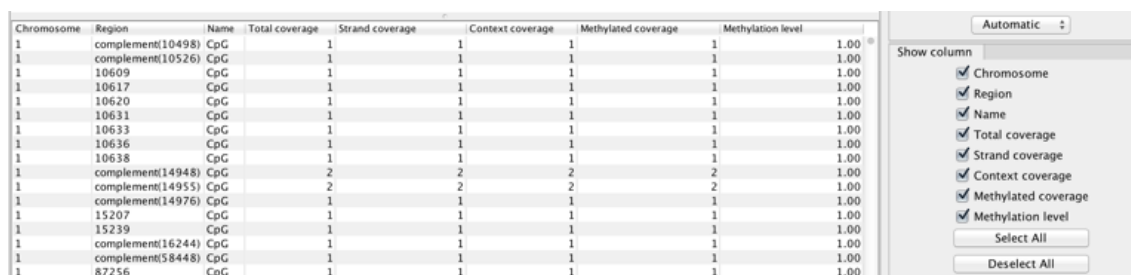
Window thresholds

- **Window length** When no window track was chosen in the previous step for focusing the analysis, examine differential methylation in windows of this fixed size. Defines the size of the window in the genome track within which methylation levels in case and control samples are compared, and statistical significance of difference, if any, is calculated and reported. Windows are evaluated sequentially along the reference.
- **Minimum number of samples** A window will be skipped, if less than this number of samples in a group have coverage at or above the **Minimum strand-specific coverage** in a minimum number of sites, as defined below.

Sample thresholds

- **Minimum high-confidence site-coverage** A site with at least this coverage is considered a high confidence site.
- **Minimum high-confidence site-count** Exclude sample from a current window, if it has fewer than this number high-confidence methylation-sites.
- **Maximum mean site coverage** Exclude sample from current window, if it has a higher mean site coverage. The default "0.0" setting does not filter any.

The tool produces a number of feature tracks and reports. Select the outputs you are interested in during the last wizard step of the tool. The **Create track of methylated cytosines** option is



Chromosome	Region	Name	Total coverage	Strand coverage	Context coverage	Methylated coverage	Methylation level
1	complement(10498)	CpG	1	1	1	1	1.00
1	complement(10526)	CpG	1	1	1	1	1.00
1	10609	CpG	1	1	1	1	1.00
1	10617	CpG	1	1	1	1	1.00
1	10620	CpG	1	1	1	1	1.00
1	10631	CpG	1	1	1	1	1.00
1	10633	CpG	1	1	1	1	1.00
1	10636	CpG	1	1	1	1	1.00
1	10638	CpG	1	1	1	1	1.00
1	complement(14948)	CpG	2	2	2	2	1.00
1	complement(14955)	CpG	2	2	2	2	1.00
1	complement(14976)	CpG	1	1	1	1	1.00
1	15207	CpG	1	1	1	1	1.00
1	15239	CpG	1	1	1	1	1.00
1	complement(16244)	CpG	1	1	1	1	1.00
1	complement(58448)	CpG	1	1	1	1	1.00
1	87256	CpG	1	1	1	1	1.00

Figure 36.27 shows a screenshot of a software interface. On the left is a table with 8 columns: Chromosome, Region, Name, Total coverage, Strand coverage, Context coverage, Methylated coverage, and Methylation level. The table contains 18 rows of data for various CpG sites on chromosome 1. On the right is a 'Show column' panel with a list of columns and checkboxes. All checkboxes are checked. Below the list are 'Select All' and 'Deselect All' buttons.

Figure 36.27: Output table.

chosen by default. It will provide a base level methylation track for each read mapping supplied, i.e., case or control (see figure 36.27 for a table view of the track).

In the table, each row corresponds to a cytosine that conforms to a context (such as 'CpG' in this example) and which has non-zero methylated coverage.

The columns of the methylation levels track table view indicate:

- **Chromosome** chromosome in which the methylated cytosine is found
- **Region** position of the mapping where the methylated cytosine is found. Rows with 'Region' values that start with 'complement' represent methylated Cs in reads that come from the original bottom strand of reference.
- **Name** of the context in which methylation is detected (see tooltip of the wizard for the names and definition of the various contexts available.)
- **Total coverage** total reads coverage of the position. May be calculated after filtering for non-specific, broken, and duplicate reads if these options are enabled.
- **Strand coverage** of the total coverage, how many reads are in the same direction than the strand in which the methylated C is Fdetected (original top, or original bottom)
- **Context coverage** of the strand coverage, how many reads conform to the selected methylation context
- **Methylated coverage** how many reads support evidence of methylation in this position, i.e., retained Cs instead of conversion to Ts
- **Methylation level** "Methylated coverage" divided by "Context coverage"

For each mapping, you can also generate an optional summary report by selecting the **Create methylation reports** option. This report includes statistics of direction of mapping of reads/read pairs, chosen contexts, and useful graphs. The graphs can help detect any bias in called methylation levels that commonly occurs at the start of BS-seq reads due to end-repair of DNA fragments in library preparation. This facilitates setting the correct trimming parameters for Read 1 soft clip, Read 2 soft clip.

Note that positions where no methylation was detected are filtered from the final output and are not reported in the 'Methylation levels' feature track. However they are included in the intermediate calculations for differential methylation detection.

When the statistical test is performed, a feature track is produced. If more than one methylation context is chosen, a separate feature track is produced for each individual context, i.e., for CpG, CHH, etc. The table view of such track for **Fisher exact** test is shown in figure 36.28.

Chromosome	Region	Name	Cytosines	Case samples	Case coverage	Case coverage mean	Case methylated	Case methylation level	Control samples	Control coverage	Control coverage mean	Control methylated	Control methylation level	p-value
1	540001..541000		22	1	14	0.64	12	0.86	1	13	0.59	5	0.38	0.02
1	567001..568000		48	1	56	1.17	9	0.16	1	25	0.52	0	0.00	0.03
1	833001..834000		15	1	7	0.59	0	0.00	1	11	0.61	2	0.18	9.05E-3
1	855001..856000		17	1	12	0.71	10	0.83	1	10	0.59	4	0.40	0.05
1	911001..912000		25	1	10	0.40	6	0.60	1	21	0.84	2	0.10	5.92E-3
1	951001..952000		12	1	10	0.83	9	0.90	1	2	0.17	0	0.00	0.05
1	1005001..1006000		19	1	6	0.32	6	1.00	1	15	0.79	5	0.33	8.51E-3
1	1072001..1073000		48	1	25	0.52	5	0.20	1	33	0.69	1	0.03	0.05
1	1079001..1080000		24	1	17	0.71	10	0.59	1	16	0.67	1	0.06	1.67E-3

Figure 36.28: Example of table with statistical test output.

The columns of the differential methylation feature track table indicate:

- **Name** column not used
- **Cytosines** total number of cytosines in the region
- **Case samples** number of samples in the case group
- **Case coverage** sum of "Total coverage" values of the region in the case group
- **Case coverage mean** sum of "Context coverage" in the region divided by the number of covered Cs in context in the region in the case group
- **Case methylated** sum of "Methylated coverage" in the region in the case group
- **Case methylation level** "Case methylated" divided by "Case coverage mean"
- **Control samples** number of samples in the control group
- **Control coverage** sum of "Total coverage" values of the region in the control group
- **Control coverage mean** sum of "Context coverage" in the region divided by the number of covered Cs in context in the region in the control group
- **Control methylated** sum of "Methylated coverage" in the region in the control group
- **Control methylation level** "Case methylated" divided by "Case coverage mean" for the control group
- **p-value** probability of no difference in methylation levels between case and control in the region, given the data and the statistical test applied

For the highlighted window region 833001..834000, the relevant values used in the hypergeometric test are 6 (the number of methylated cytosines in the case sample) out of 7 (total number of cytosines), while the control sample had 11 covered context-conforming cytosines in the region, of which only 2 were methylated. If there are no case/control difference in methylation, the probability (p-value) of such hypermethylation in the case sample is calculated as 9.05×10^{-3} , below the threshold.

36.3.4 Create RRBS-fragment Track

Create RRBS-fragment Track can be used to generate fragments of pre-selected size based on the restriction digest of a reference genome with commonly used frequent cutters targeting methylation contexts such as *MspI*. If such track is provided to the **Call Methylation Levels** tool, its features are used to calculate statistical tests instead of consecutive windows of pre-set size. The input object is a reference genome of interest, in a track format.

Figure 36.29 shows the options for generating a track with predicted, and selected features corresponding to restriction digest of the selected genome.

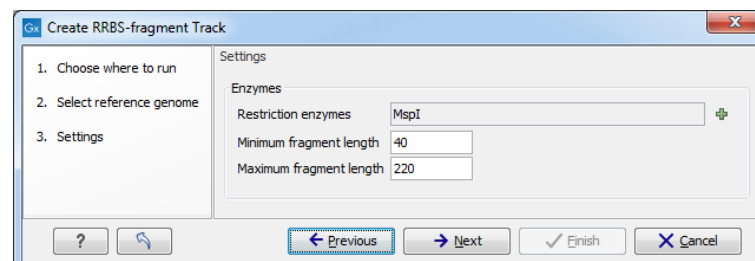


Figure 36.29: Create RRBS-fragment track settings.

Restriction enzymes The enzymes used in reduced-representation bisulfite sequencing can be selected here, with the most common for the CpG islands, *MspI*, pre-selected by default.

Parameters **Minimum fragment length** and **Maximum fragment length** define the acceptable size range, out of the all possible predicted fragments after full digest of reference DNA, will be included in the output of the tool.

The output of the tool is a feature track.

36.4 Advanced Peak Shape Tools

The Advanced Peak Shape Tools folder provides advanced tools to manipulate peak shapes and find regions in sequencing data that exhibit a distinct shape. These tools are used by the ChIP-Seq algorithm (see section 36.2), but they can also be used interactively to refine the results of ChIP-Seq peak calling or to perform shape-based analyses of sequencing data.

These tools are available under:

Tools | Epigenomics Analysis (📁) | **Advanced Peak Shape Tools** (📁)

The **Learn Peak Shape Filter** tool allows you to build a new peak shape filter from sequencing data and a set of positive and negative regions.

The **Apply Peak Shape Filter** tool allows you to apply a peak shape filter to sequencing data to discover regions (peaks), which match a given peak shape.

The **Score Regions** tool allows you to apply a peak shape filter to score genomic regions according how they match a given peak shape.

36.4.1 Learn Peak Shape Filter

The Learn Peak Shape Filter tool allows you to build a new peak shape filter from sequencing data and a set of positive and negative regions. The resulting filter can be used to identify genomic regions whose read coverage profile matches the characteristic shape of the positive examples and does not match the shape of the negative examples. See section 36.2.2 for the procedure used to build the peak shape filter.

An example of such filter is shown in figure 36.30.

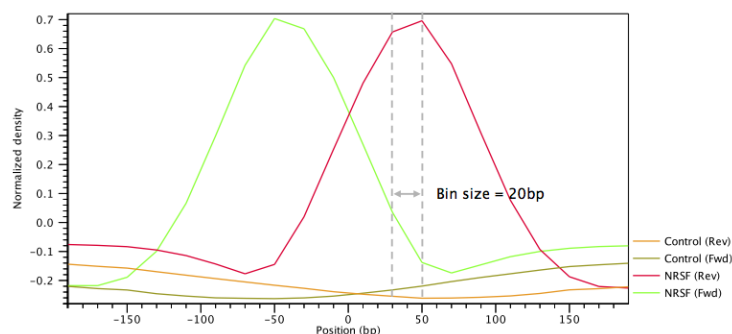





Figure 36.30: Example of a peak shape filter with a window size of 400bp made up of 20 bins of size 20bp each. The filter was built from ChIP-Seq data of the transcription factor NRSF and a control ChIP-Seq experiment.

To run the Learn Peak Shape Filter tool:

Tools | Epigenomics Analysis  | **Advanced Peak Shape Tools**  | **Learn Peak Shape Filter** 

This will open up the wizard shown in figure 36.31 where you can select the input data (for example the mapped ChIP-Seq reads). Track based read mappings  and stand-alone read mappings  /  are both accepted. Multiple inputs are accepted, provided that they refer to the same genome.

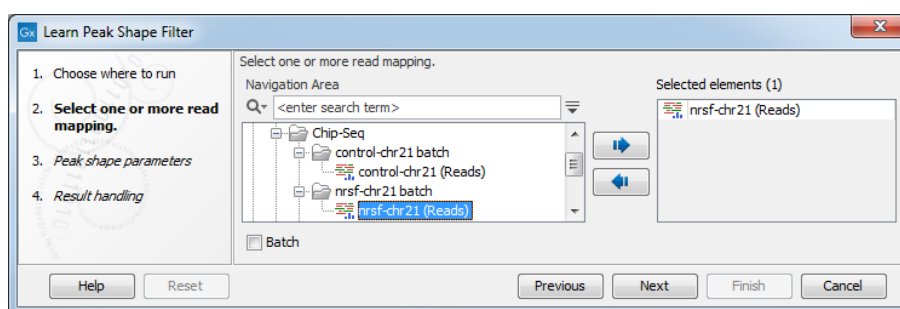



Figure 36.31: Select the input data for the Learn Peak Shape Filter tool.

Click **Next** to go to the next wizard step (shown in figure 36.32).

In this wizard step you have the following options:

- **Location of positive regions** An annotation track  containing the location of the positive regions (e.g. ChIP-Seq peaks) that will be used to build the peak shape filter. The set

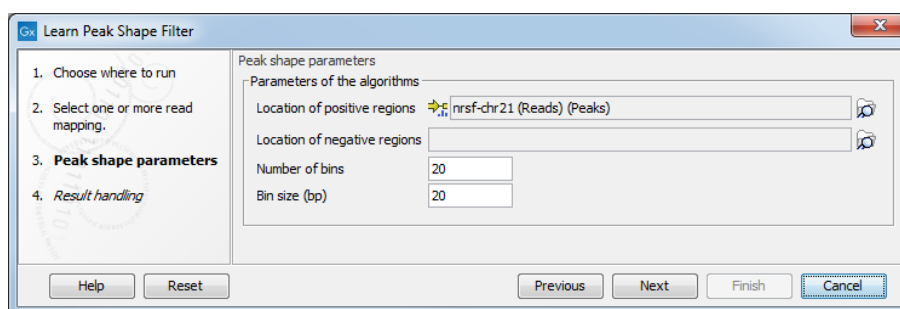


Figure 36.32: Options for Learn Peak Shape Filter.

of positive regions should include examples where the shape is clearly exhibited. It is preferable to have fewer peaks with high quality rather than a large amount of ambiguous peaks. Typically, a number of positive peaks greater than 5-10 times the number of bins is sufficient to learn a well-defined shape.

- **Location of negative regions** An annotation track (📁) containing the location of the negative regions that will be used to build the peak shape filter (e.g. background, PCR artifacts or examples of bad peaks from a previous run of the ChIP-Seq analysis tool). If no annotation track is provided, a negative profile will be derived from sequencing noise.
- **Number of bins** The number of bins to use to build the filter. The default value of 20 for the Number of bins parameter should be satisfactory for most uses. Higher values may be useful when the shape to be learned is particularly complex. Note that if the chosen number of bins is very large, the learned peak shape filter may not be smooth and could over-fit the input data. If only few positive regions are available, reducing the number of bins may be helpful.
- **Bin size** The size of each bin in base pairs. The bin size is related to the window size (i.e. the length of the shape to be learned) by the formula $\text{Window size} = \text{Bin size} \times \text{Number of bins}$ (see figure 36.30).

The result of the algorithm will be a **Peak shape filter** (📊), which can then be applied to call peaks or score regions using Apply Peak Shape Filter. After clicking on the button labeled **Next**, you can choose whether you want to open the result directly, or save the results in the **Navigation Area**. If you choose to save the results, you will be asked to specify where you would like to save them.

36.4.2 Apply Peak Shape Filter

The Apply Peak Shape Filter tool allows you to apply a peak shape filter to sequencing data to discover regions (peaks), which match a given peak shape (see section 36.2.3).

To run the Apply Peak Shape Filter tool:

Tools | Epigenomics Analysis (📁) | **Advanced Peak Shape Tools** (📁) | **Apply Peak Shape Filter** (📊)

This will open up the wizard shown in figure 36.33 where you can select the input data (e.g. mapped ChIP-Seq reads). Track based read mappings (📊) and stand-alone read mappings (📄) / (📄) are both accepted. Multiple inputs are accepted, provided that they refer to the same

genome. At least 2 input files must be provided (sample + control for example) if the tool is to be used with a filter that was built from a sample and a control. Note: the order of the inputs should be the same when building the filter and applying the filter, i.e., if the filter was generated from Sample 1 + control, with control being the second object transferred over in the wizard, then when running the Apply Peak Shape Filter tool, the control should also be the second object used in the wizard.

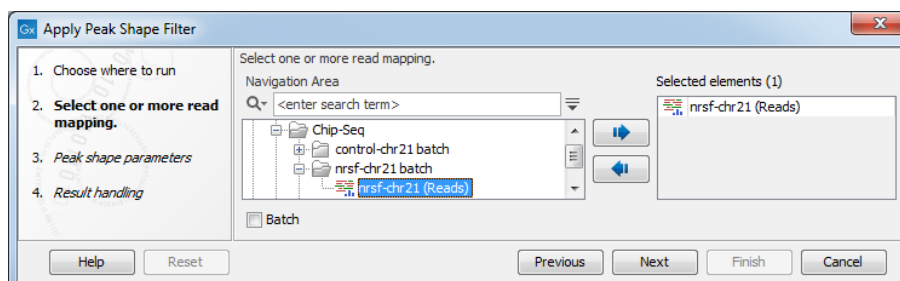


Figure 36.33: Select the input data for Apply Peak Shape Filter.

Click **Next** to go to the next wizard step (shown in figure 36.34).

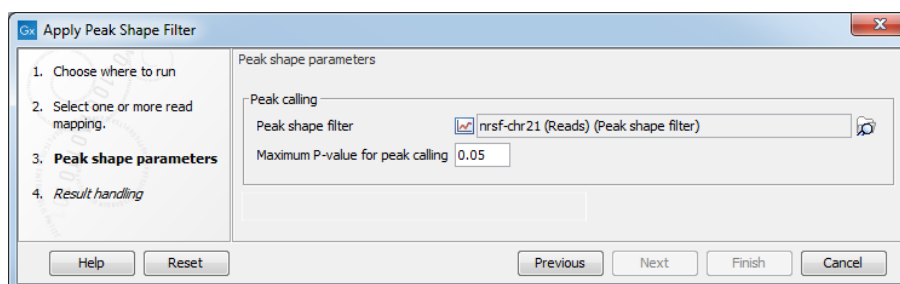


Figure 36.34: Options for Apply Peak Shape Filter.

In this wizard step you have the following options:

- **Peak shape filter** The peak shape filter (📊) to apply to the data. Peak shape filters can be obtained as the result of the ChIP-Seq Analysis tool. If no filter is given, a filter is derived from the input data.
- **Maximum P-value for peak calling** The threshold for reporting peaks can be specified by this option.

Click **Next** to go to the wizard step shown in figure 36.35.

In addition to the annotation track with **Peak annotations** (📊) that will always be generated by the algorithm, you can choose to select an additional output type:

- **Peak shape score** (📊) A graph track containing the peak shape score. The track shows the peak shape score for each genomic position. To save disk space, only scores greater than zero are reported. For the definition of peak shape score.

Choose whether you want to open the results directly, or save the results in the **Navigation Area**. If you choose to save the results, you will be asked to specify where you would like to save them.

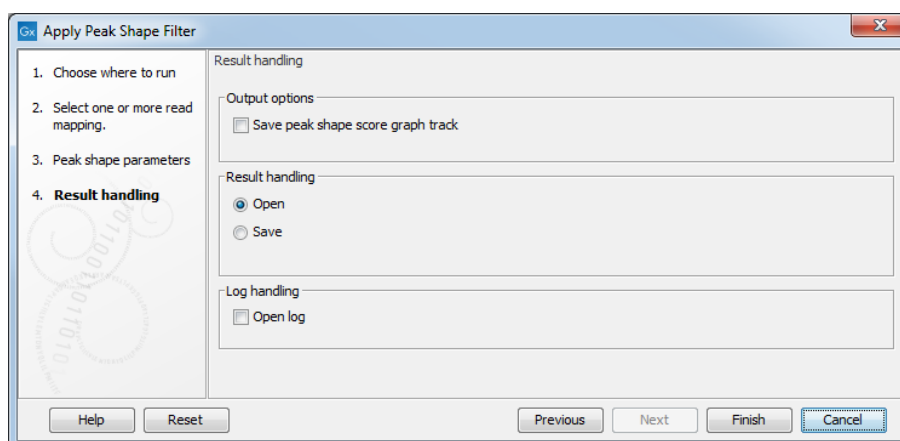


Figure 36.35: Output options for Apply Peak Shape Filter.

For more information on the **Peak track** (📊), see section 36.2.5.

36.4.3 Score Regions

The Score Regions tool allows you to apply a new peak shape filter to score genomic regions according how they match a given peak shape.

To run the Score Regions tool:

Tools | Epigenomics Analysis (📁) | **Advanced Peak Shape Tools** (📁) | **Score Regions** (📊)

This will open up the wizard shown in figure 36.36 where you can select the input data (e.g. mapped ChIP-Seq reads). Multiple inputs are accepted, provided that they refer to the same genome. Track based read mappings (📊) and stand-alone read mappings (📄) / (📄) are both accepted.

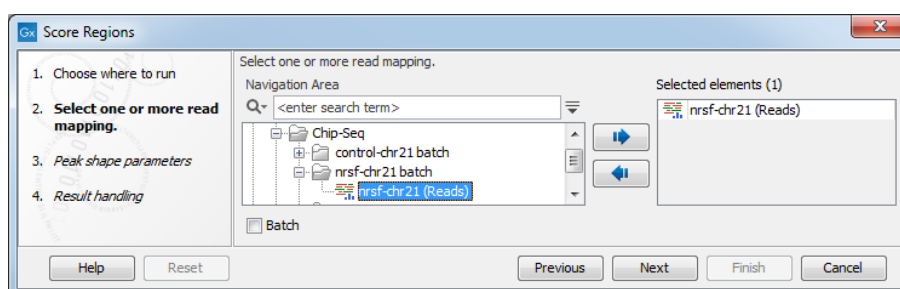


Figure 36.36: Select the input data for Score Regions.

Click **Next** to go to the next wizard step (shown in figure 36.37).

In this wizard step you have the following options:

- **Peak shape filter** The peak shape filter (📊) to apply to the data. Peak shape filters can be obtained as the result of the ChIP-Seq Analysis tool.
- **Regions to score** An annotation track (📊) containing the regions where the peak shape

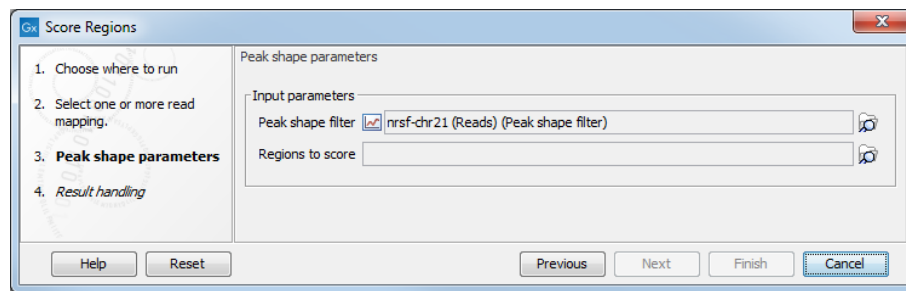


Figure 36.37: Options for Score Regions.

will be applied. The peak shape filter will be applied to every genomic position within the interval and the maximum values will be used to score the region.

The result of the algorithm will be an annotation track (📊) of the same type as the regions to score annotation track, where the columns of type **Peak shape score**, **P-value** and **Center of peak** will be added or replaced.

After clicking **Next**, you can choose whether you want to open the result directly, or save the results in the Navigation Area. If you choose to save the results, you will be asked to specify where you would like to save them.

Chapter 37

Utility tools

Contents

37.1 Extract Annotated Regions	1181
37.2 Extract Reads	1183
37.3 Filter on Custom Criteria	1187
37.4 Merge Overlapping Pairs	1189
37.5 Combine Reports	1192
37.5.1 Combine Reports output	1196
37.6 Create Sample Report	1198
37.6.1 Create Sample Report output	1204
37.7 Create Report from Table	1205
37.8 Modify Report Type	1207
37.8.1 Modifying report types in workflows	1209
37.9 Track tools	1213
37.10 Create Sequence List	1213
37.11 Update Sequence Attributes in Lists	1213
37.12 Split Sequence List	1217
37.13 Subsample Sequence List	1219
37.14 Rename Elements	1221
37.15 Rename Sequences in Lists	1225

37.1 Extract Annotated Regions

Using **Extract Annotated Regions**, parts of a sequence (or several sequences) can be extracted based on annotations. Lengths of flanking regions can be specified if desired. The output is a sequence list that contains sequences carrying the annotation specified.

Some examples of the use of this tool:

- Extract all tRNA gene sequences from a genome.
- Extract sequences for regions where annotations contain particular text

- Extract sequences of differentially expressed regions by using RNA-Seq statistical comparisons as the annotation source.

To launch **Extract Annotated Regions**, go to:

Tools | Utility Tools (🔧) | Extract Annotated Regions (👉)

This opens the wizard. In the first step (figure 37.1) you can select one or more annotated sequences, annotation tracks, variant tracks, or statistical comparison tracks.

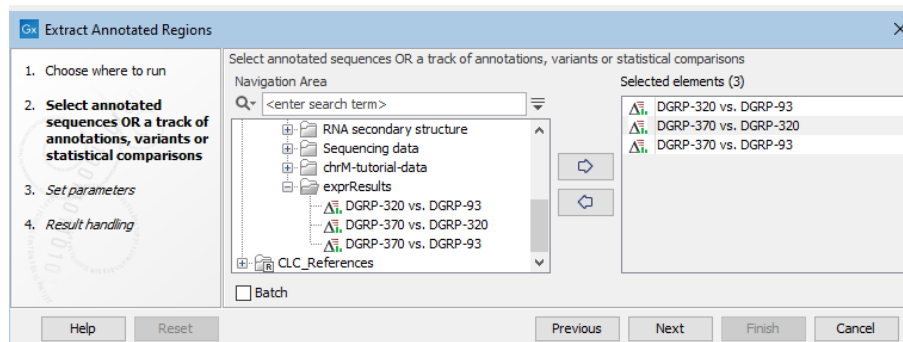


Figure 37.1: *Selecting input. Here, statistical comparison tracks have been selected.*

If you selected tracks as input, you must enter a reference sequence track in the next wizard step. In that step, you can also specify particular annotations where regions should be extracted, and flanking region lengths to be included, if desired (figure 37.2).

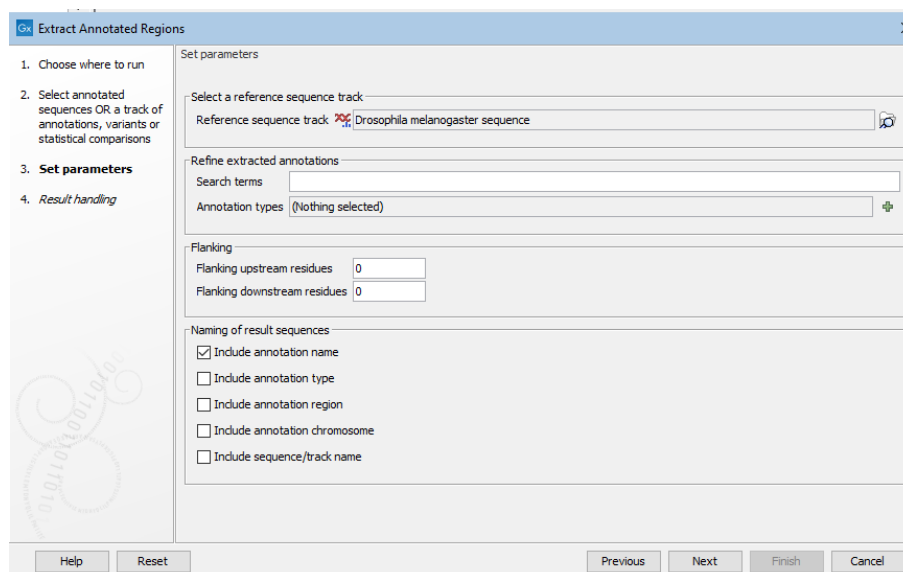


Figure 37.2: *Specifying a reference sequence track after track-based data was selected as input.*

- **Search terms** All annotations and attached information for each annotation will be searched for the entered term. It can be used to make general searches for search terms such as "Gene" or "Exon", or it can be used to make more specific searches. For example, if you have a gene annotation called "MLH1" and another called "MLH3", you can extract both annotations by entering "MLH" in the search term field. If you wish to enter more specific search terms, separate them with commas: "MLH1, Human" will find annotations where both "MLH1" and "Human" are included.

- **Annotation types** If only certain types of annotations should be extracted, this can be specified here.

The sequence of interest can be extracted with flanking sequences:

- **Flanking upstream residues** The output will include this number of extra residues at the 5' end of the annotation.
- **Flanking downstream residues** The output will include this number of extra residues at the 3' end of the annotation.

The sequences that are created can be named after the annotation name, type, etc:

- **Include annotation name** This will use the name of the annotation in the name of the extracted sequence.
- **Include annotation type** This corresponds to the type chosen above and will put this information in the name of the resulting sequences. This is useful information if you have chosen to extract "All" types of annotations.
- **Include annotation region** The region covered by the annotation on the original sequence (i.e. not including flanking regions) will be included in the name.
- **Include sequence/track name** If you have selected more than one sequence as input, this option enables you to discern the origin of the resulting sequences in the list by putting the name of the original sequence into the name of the resulting sequences.

Click **Finish** to start the tool.

37.2 Extract Reads

Extract Reads takes a stand-alone read mapping or reads track as input and extracts the mapped reads to a reads track or sequence list. All reads or a subset of reads can be extracted. Subsets can be specified based on location relative to a set of regions and/or based on specified characteristics of the reads.

To launch **Extract Reads**, go to:

Tools | Utility Tools  | **Extract Reads** 

Extracting reads based on their location relative to a set of regions can be configured in the 'Select overlap (optional)' wizard step (figure 37.3):

Overlap options

- **Overlap tracks** Extract only reads that overlap one or more regions in the provided overlap track(s). The reference genome of the input and overlap tracks must be compatible.
- **Type of overlap** (figure 37.4).
 - **Any overlap.** Extract reads that overlap any region.

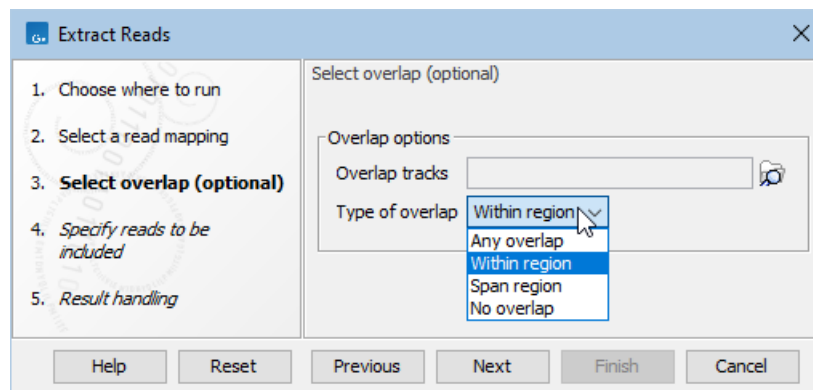


Figure 37.3: Overlap tracks can be used for extracting reads mapped to particular areas of the reference genome.

- **Within region.** Extract reads fully within an overlap region. Reads overlapping region boundaries are not extracted.
- **Span region.** Extract reads with residues that align on both sides of an overlap region. For paired reads, fragments that span a region are extracted. The option **Only include matching read(s) of read pairs** (see below) can be used to solely extract individual reads of a pair that span a region.
- **No overlap.** Extract reads that do not overlap any region in the provided overlap track(s).

The nature of the extracted reads can be specified in the 'Specify reads to be included' wizard step (figure 37.5). Note that reads in read mappings are colored according to their characteristics, see section 30.3.1.

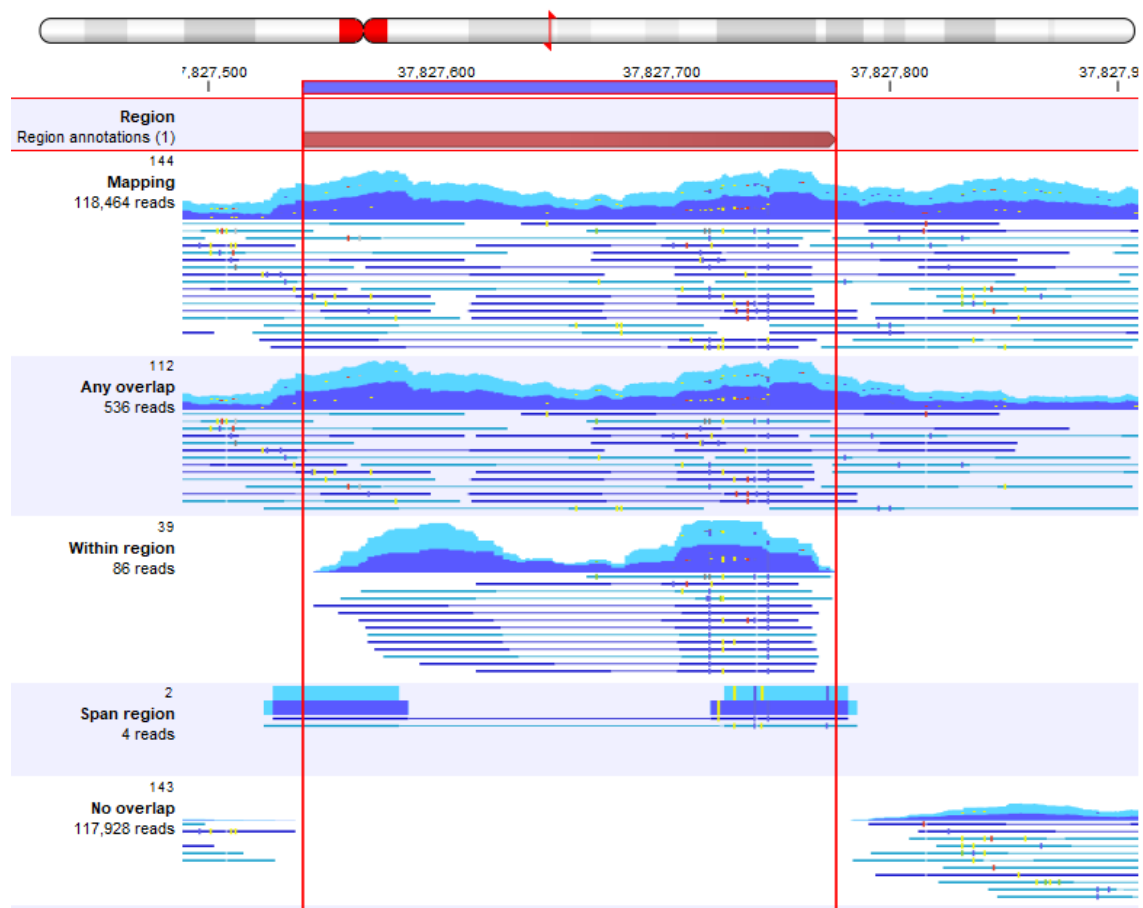


Figure 37.4: A track list including, from top to bottom: an overlap track, a read mapping, and read mappings with extract reads for 'Type of overlap': 'Any overlap', 'Within region', 'Span region', and 'No overlap'.

Match specificity

- **Include specific matches** Reads that mapped best to just a single position of the reference genome.
- **Include non-specific matches** Reads that have multiple, equally good alignments to the reference genome. These reads are colored yellow by default in read mappings.

Alignment quality

- **Include perfectly aligned reads** Reads where *the full read* is perfectly aligned to the reference genome. Reads that extend beyond the end of the reference are not considered perfectly aligned, because part of the read does not match the reference.
- **Include reads with less than perfect alignment** Reads with mismatches, insertions or deletions, or with unaligned ends.

Spliced status

- **Include spliced reads** Reads mapped across an intron.
- **Include non spliced reads** Reads not mapped across an intron.

Figure 37.5: Options to include or exclude specific types of reads.

Paired status

- **Include intact paired reads** Paired reads mapped within the specified paired distance.
- **Include reads from broken pairs** Paired reads where only one of the reads mapped, either because only one read in the pair matched the reference, or because the distance or relative orientation of its mate was wrong.
- **Include single reads** Reads marked as single reads (as opposed to paired reads). Reads from broken pairs are not included. Reads marked as single reads after trimming paired sequence lists are included.
- **Only include matching read(s) of read pairs** If only one read of a read pair matches the criteria, then only include the matching read as a broken pair. For example if only one of the reads from the pair is inside the overlap region, then this option only includes the read found within the overlap region as a broken read. When both reads are inside the overlap region, the full paired read is included. Note that some tools ignore broken reads by default.


Orientation

- **Include forward reads** Reads mapped in the forward direction.









- **Include reverse reads** Reads mapped in the reverse direction.

Note that excluding forward or reverse reads will generate broken pairs if reads in pairs are mapped in opposite directions, regardless of the **Only include matching read(s) of read pairs** option.

37.3 Filter on Custom Criteria

Filter on Custom Criteria takes as input an element, for which data can be displayed as rows in a table view () , and a set of filter criteria. As many criteria as desired can be defined. The tool outputs an element containing only the rows that satisfy the specified criteria.


Many element types are accepted as input, including:

- Annotation track ()
- Expression track ()
- IsomiR table ()
- miRNA seeds table ()
- Sequence list ()
- Statistical comparison table ()
- Statistical comparison track ()
- Variant track ()


To run **Filter on Custom Criteria**, go to:

Tools | Utility Tools () | **Filter on Custom Criteria** ()

After selecting the input element, the criteria to filter on are defined. A criterion is composed of an attribute, a comparison operator, and a value to compare to (figure 37.6).

To make defining criteria easier, the **Load Attributes** button can be used to populate a drop down list of the attributes found in the element selected in the **Template element** field. By default, the tool's input is preselected. Use the browse () button (figure 37.6) to select a different element. We recommend selecting as template a small element containing all necessary attributes, as loading attributes from large elements can take a long time.

If a template element is not used for populating a drop down list, attributes can be entered by typing directly in the **Attribute** field.

The **Add** button adds additional criteria. The X () button removes that criterion. The **Clear** button removes all defined criteria.

Below the criteria definitions, there is an option to choose whether all criteria should be fulfilled (**Match all**) or if just one needs to be fulfilled (**Match any**), for a row to be included in the output.

Criteria using attributes not present in the input element are ignored.

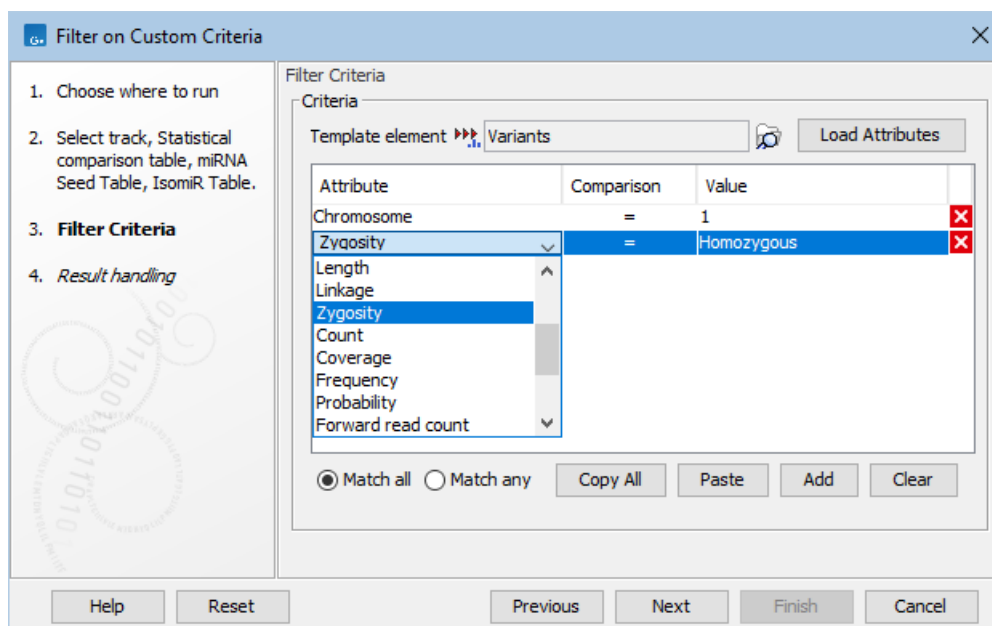


Figure 37.6: Filter criteria to extract the homozygous variants found on chromosome 1. The drop down menu shows some of the attributes populated using the Load Attributes button.

Reusing filter criteria

Previously defined filter criteria can be reused in subsequent runs, saving time and making it easy to apply a set of criteria consistently.

Copied filter criteria can be pasted into the 'Filter Criteria' wizard step using the **Paste** button. Pasting appends the copied criteria to any already defined criteria.

Filter criteria can be copied in two ways:

- The criteria defined in the the 'Filter Criteria' wizard step can be copied using the **Copy All** button. This can be pasted into a text file for later use.
- The criteria used in previous runs of **Filter on Customer Criteria** can be copied from the History view (📄) of the output (figure 37.7).

Additional criteria can then be added, and unwanted criteria removed, as described above.

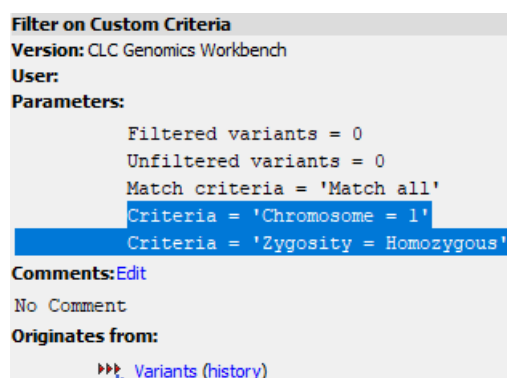


Figure 37.7: The history of an element output by Filter on Custom Criteria includes the criteria used. This can be copied and then pasted into the tool in a subsequent run.

37.4 Merge Overlapping Pairs

Some paired end library preparation methods using relatively short fragment size will generate data with overlapping pairs. This type of data can be handled as standard paired-end data in the CLC Genomics Workbench. (For variant detection related information, see section 31.1.3).

In some situations it can be useful to merge each overlapping pair into a single sequence read. This generates longer reads, and can improve the quality, as normally the quality drops towards the end of a read. By overlapping the ends of two reads, the resulting consensus read reflects two read ends instead of just one.

Merge overlapping paired reads by going to:

Tools | Utility Tools (🔧) | **Sequence Lists** (📄) | **Merge Overlapping Pairs** (🔗)

As input, select one or more sequence lists containing paired reads in forward-reverse orientation.

The resulting merged reads will be in the forward orientation.

Note: Length trimming of reads can be done before or after merging, however the merged read's 3' is the 5' of the reverse read, so that trimming the original reads using the **Remove 5' terminal nucleotides** option corresponds to trimming the merged reads using both the **Remove 5' terminal nucleotides** option and the **Remove 3' terminal nucleotides** option.

The options available in the launch wizard are shown in figure 37.8.

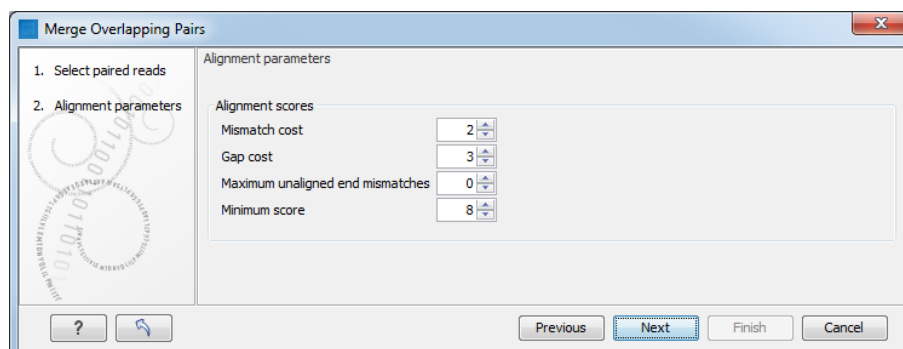


Figure 37.8: Merging overlapping pairs configuration options.

The tool options are best explained in the context of the merging algorithm: Because the fragment

size is not an exact number of base pairs and is different from fragment to fragment, an alignment of the two reads has to be performed. If the alignment is *good and long enough*, the reads will be merged. *Good enough* in this context means that the alignment has to satisfy some user-specified score criteria (details below). Because of sequencing errors that typically are more abundant towards the end of the read, the alignment is not expected always to be perfect, and the user can decide how many errors are acceptable. *Long enough* in this context means that the overlap between the reads has to be non-coincidental. Merging two reads that do not really overlap, leads to errors in the downstream analysis, thus it is very important to make sure that the overlap is big enough. If only a few bases overlap was required, some read pairs will match by chance, so this has to be avoided.

The following options define what is *good enough* and *long enough*

- **Mismatch cost** The alignment awards one point for a match, and the mismatch cost is set by this parameter. The default value is 2.
- **Gap cost** This is the cost for introducing an insertion or deletion in the alignment. The default value is 3.
- **Max unaligned end mismatches** The alignment is local, which means that a number of bases can be left unaligned. If the quality of the reads is dropping to be very poor towards the end of the read, and the expected overlap is long enough, it makes sense to allow some unaligned bases at the end. However, this should be used with great care which is why the default value is 0. As explained above, a wrong decision to merge the reads leads to errors in the downstream analysis, so it is better to be conservative and accept fewer merged reads in the result.
- **Minimum score** This is the minimum score of an alignment to be accepted for merging. The default value is 10. As an example: with default settings, this means that an overlap of 13 bases with one mismatch will be accepted (12 matches minus 2 for a mismatch).

Please note that even with the alignment scores above the minimum score specified in the tool setup, the paired reads also need to have the number of end mismatches below the "Maximum unaligned end mismatches" value specified in the tool setup to be qualified for merging.

After configuring the merging options, the option to create a report is offered.

The main result will be two sequence lists for each list in the input: one containing the merged reads (marked as single end reads), and one containing the reads that could not be merged (still marked as paired data). Since the *CLC Genomics Workbench* handles a mix of paired and unpaired data, both of these sequence lists can be used in the further analysis. However, please note that low quality can be one of the reasons why a pair cannot be merged. Hence, the list of reads that could not be paired is more likely to contain more reads with errors than the one with the merged reads.

Quality scores come into play in two different ways when merging overlapping pairs.

First, if there is a conflict between the reads in a pair (i.e. a mismatch or gap in the alignment), quality scores are used to determine which base the merged read should have at a given position. The base with the highest quality score will be the one used. In the case of gaps, the average of the quality scores of the two surrounding bases will be used. In the case that two conflicting

bases have the same quality or both reads have no quality scores, an [IUPAC ambiguity code](see the appendix section [H](#)) representing these bases will be inserted.

Second, the quality scores of the merged read reflect the quality scores of the input reads.

We assume independence of errors in calculating the new quality score for a merged position and carry out the following calculations:

- When the two reads agree at a position, the two quality scores are summed to form the quality score of the base in the new read. The score is capped at the maximum value on the quality score scale which is 64. Phred scores are log scores, so their sums represent the multiplication of the original error probabilities.
- If the two bases disagree at a position, the quality score of the base in the new read is determined by subtracting the lowest score from the highest score of the input reads. Similar to the addition of scores when bases are the same, this adjusts the error probability to reflect a decreased certainty that the base reported at that position is correct.

Thus, if two bases at a given position of an overlapping region are different, and each of those bases was originally given a high phred score, the score assigned to the merged base will be very low. This reflects the fact that the base at this position is unreliable.

If a base at a given position in one read of an overlapping region has a very low quality score and the base at that position in the other read has a high score, it is likely that the base with the high quality score is correct. The adjusted quality score for this position in the merged read would reflect that there is less certainty in the base at that position than before. However, such a position would still be assigned quite a high quality, as the base call is still likely to be correct.

Figure [37.9](#) shows an example of the report generated when merging overlapping pairs.

It contains three sections:

- A summary showing the numbers and percentages of reads that have been merged.
- A plot of the alignment scores. This can be used to guide the choice of minimum alignment score as explained in section [37.4](#).
- A plot of read lengths. This shows the distribution of read lengths for the pairs that have been merged:
 - The length of the overlap.
 - The length of the merged read.
 - The combined length of the two reads in the pair before merging.

1 Summary

	Number of reads	Percentage
Merged	20,105,092	44.53%
Not merged	25,044,608	55.47%
Total	45,149,700	100%

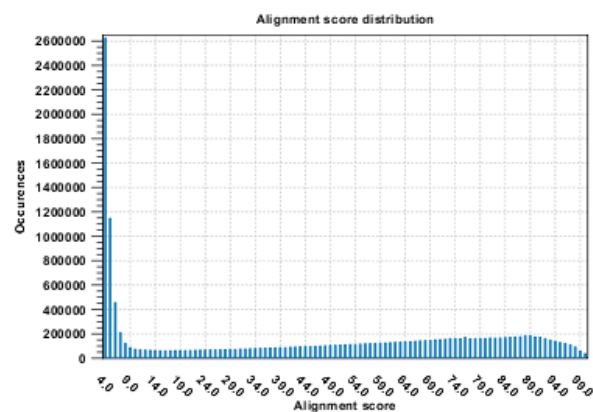
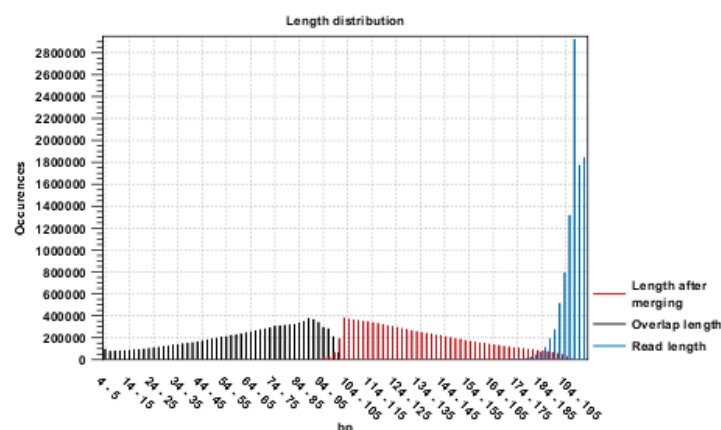
2 Alignment score distribution**3 Length distribution**

Figure 37.9: Report of overlapping pairs.

37.5 Combine Reports

Combine Reports takes reports as input and summarizes their contents. A single summary report is generated. The report content and its order is configurable.

To obtain a sample report with an overview of a single sample, use the **Create Sample Report** tool (see section 37.6). **Combine Reports** can take sample reports as input, allowing the generation of comprehensive, cross-sample summary reports.

Creating a combined report

To run **Combine Reports**, go to:

Tools | **Utility Tools** (🔧) | **Reports** (📄) | **Combine Reports** (🔗)

To see a list of tools that produce reports that can be used as input, click on the (i) icon in

the top right corner of the input selection wizard (figure 37.10).

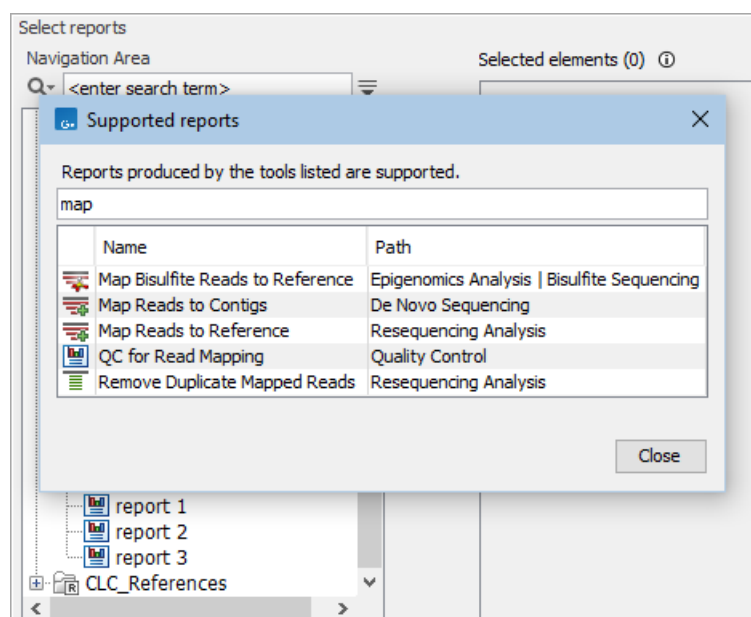


Figure 37.10: Clicking on the info icon at the top right corner of the input selection wizard opens a window showing a list of tools that produce supported reports. Text entered in the field at the top limits the list to just tools with names containing the search term.

One section is included in the combined report for each report type provided as input, with the type determining the title of the corresponding section. See **Report types and combined report content** below for further details.

Specifying the order of the sections in the report

When more than one report type is provided as input, the order of the sections in the output report is configurable in the "Set order" wizard step:

- **Order of inputs** Use the order that the input reports were specified in.
- **Define order** Explicitly define the section order by moving items up and down in the listing. Defining the order is recommended when the tool is being launched in batch mode with folders of reports provided as the batch units. Doing this avoids reliance on the order of the elements within the folders being the same.

When **Combine Reports** is included in a workflow, sections are ordered according to the order of the inputs. See section 14.1.3 for information about ordering inputs in workflows.

Specifying the content of the report

The content of the report is configured in the "Set contents" wizard step (figure 37.12):

- Under "Report contents", the following options are available:

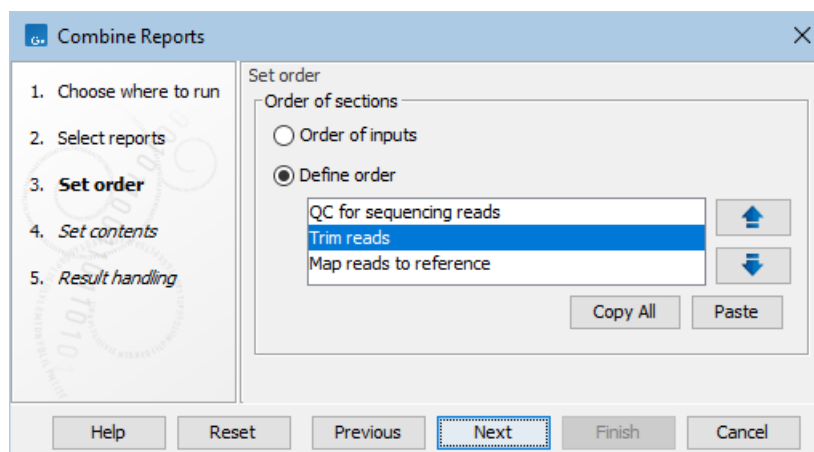


Figure 37.11: When more than one report type is provided as input, the order of the sections can be configured in the "Set order" wizard step.

- **Show summary items as plots** When checked, summary items are displayed as box plots instead of tables, where possible.
- **Include tables for outliers** When checked, samples detected as outliers for each table/box plot are added to an outliers table, which is printed under the table/box plot.
- Under "Include", the sections to be added to the report are specified. The combined report contains a general summary section that appears at the top of the report when included. Where available, individual subsections and summary items can also be specified. Where only some subsections or summary items are excluded, the checkbox for the parent section(s) are highlighted for visibility.

See section [37.5.1](#) for information about the generated report.

Reusing configurations

Configurations defined previously can be used in subsequent runs.

Configurations can be copied in two ways:

- Copy the configuration defined in the relevant wizard step using the **Copy all** button.
- Copy the configuration used in previous runs of the tool from the History (🕒) view of the output, described further below.

Copied configurations can be pasted into a text file for later use.

A copied configuration can be pasted into the wizard step using the **Paste** button.

Any existing settings in that wizard step will be overwritten.

The history of a report output by **Combine Reports** contains both the order of the sections (Order reports) and the excluded sections/subsections/summary items (Exclude) (figure [37.13](#)). These

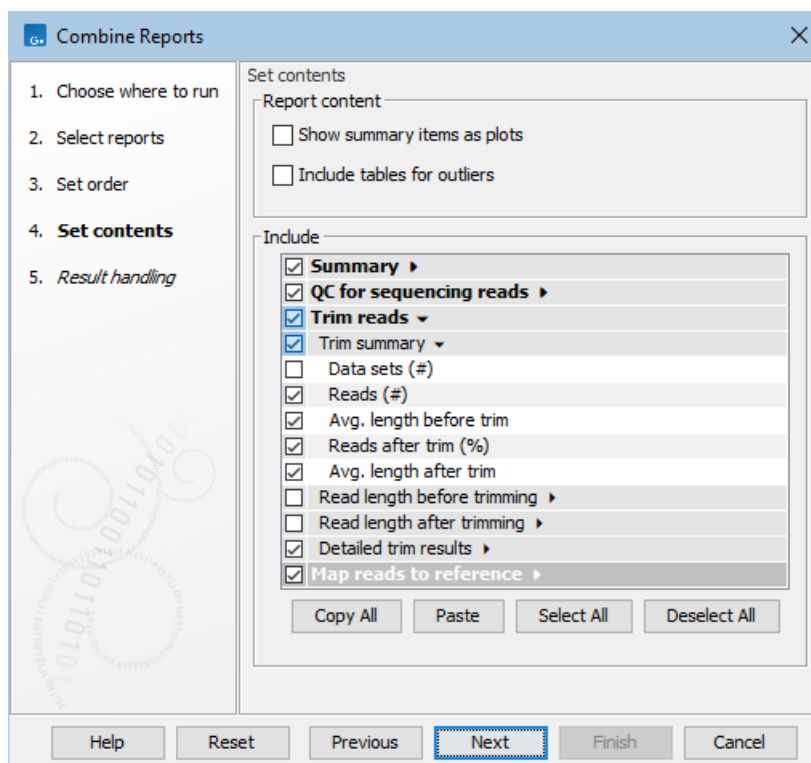


Figure 37.12: The content of the combined report is configured in the "Set contents" wizard step. Sections with a check in the box are included, while those without a check are excluded from the combined report. For visibility, sections where some contents have been excluded have checkboxes highlighted.

can be selected, copied, and then pasted into the "Set order"/"Set contents" wizard steps, respectively, in a subsequent run. Alternatively, the entire history can be selected, copied, and then pasted in each wizard step. Only the relevant configuration is pasted into each step.

Report types and combined report content

Reports that can be supplied as input have a report type. The type of a report affects the placement of its summary information in the output. Specifically:

- Reports with the *same type* that are generated by *the same tool* are summarized into a single section, named according to the type.

This is useful when the aim is to compare the values from those reports, for example results from different samples or different analysis runs. However, if a particular tool has been used more than once in an analysis, for different purposes, then placing the summary of these results in different sections may be desirable. This can be done by editing the report type in some of the reports (see section 37.8.).

- Reports with *different types* are summarized in separate sections, named according to the types.

The report type assigned by a particular tool is unique, so reports generated by different tools have different types.

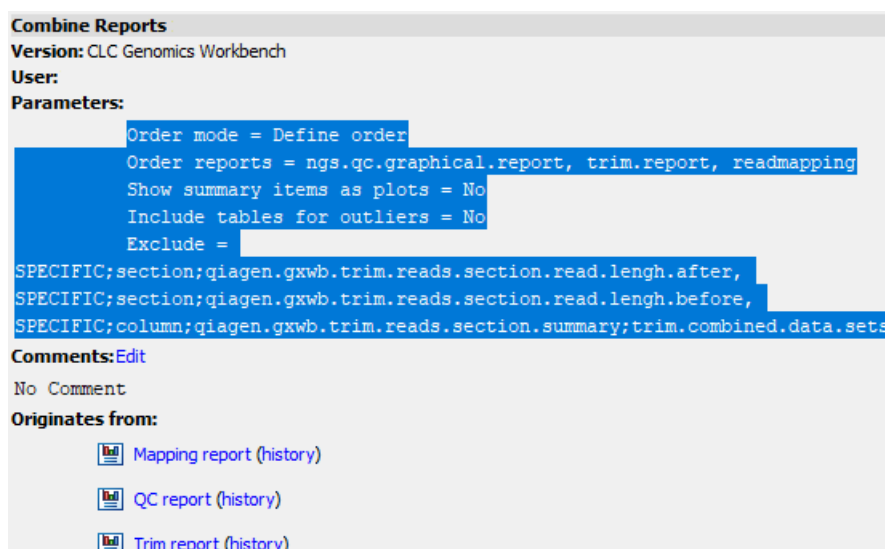



Figure 37.13: The history of a report output by Combine Reports with the parameters selected, ready to be copied.

If reports generated by different tools are later modified so their report types are the same, those reports will still be summarized in different sections, although each of these sections will have the same name.

The type of a report can be seen in the Element Info () view for that report.

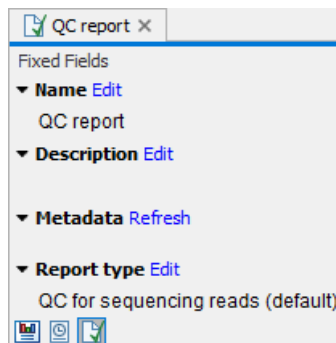



Figure 37.14: The type of a report can be found in the Element Info view of reports that are supported as input for tools that summarize reports.

37.5.1 Combine Reports output

Combine Reports generates a single report () containing summary items and other selected information from the reports provided as input.

The report contains one section per input report type, as described in section 37.5. Summary items are displayed in table format.

Note: The summaries for reports produced by **Trim Sequences** do not follow the format described below.

The tables contain one row per input report and one column per summary item. The last rows, shaded in pale gray, report the minimum, median, maximum, mean and standard deviation for all numeric summary items (figure 37.15).

The first column indicates the sample name, which is either the name of the input report or the sample name as determined by **Create Sample Report**, see section 37.6. The combined report contains links to the input reports and clicking on the sample name selects the corresponding report in the Navigation Area.

When **Show summary items as plots** is checked, tables are displayed as box plots, wherever possible. Each numeric summary item will be represented as one box in the plot (figure 37.15). Tables containing summary items that are not numbers, or numbers with very different ranges (for example, a percentage and the number of inputs) cannot be displayed as box plots.

Highlighted cells

Table cells are highlighted (figure 37.15):

- In yellow if they are detected as outliers. For each numeric summary item, the lower quartile - 1.5 IQR (interquartile range) to upper quartile + 1.5 IQR range is calculated using all the values for the summary item. Samples with values outside this range are considered outliers.
- In pink if they are considered problematic. A potential problem has been identified that is explained underneath the table.
- In red if both an outlier and problematic.

When **Include tables for outliers** is checked, tables/plots containing the summary items will be followed by an additional table containing the identified outliers, with one row for each summary item containing outliers (figure 37.15).

Summary section

By default, combined reports contain a summary section, offering a quick overview of samples that have been identified as outliers and/or problematic. The summary section is only present if it was included when configuring the report content (see section 37.5) and it only contains summaries those sections/subsections/summary items that are also included in the combined report.

Combined sample reports

When sample reports with "Quality control" subsections generated using **Create Sample Report** are used as input to the **Combine Reports** tool, the combined report will include a "Quality control" section. One table is added for each summary item from the sample reports "Quality control" subsections. The tables contain the sample name, values and thresholds as columns, and have one row per input report. Samples are color coded, just like in the individual sample reports (see section 37.6).

The combined report icon summarizes the overall status:

2.6 Distribution of biotypes

The table is based on 8 samples.

Sample name	protein_coding	rRNA	processed_transcript
sample1	96.05	3.95	0
sample2	100.00	0.00	0
sample3	95.50	4.50	0
sample4	96.01	3.99	0
sample5	69.93	30.07	0
sample6	89.06	10.94	0
sample7	57.84	42.16	0
sample8	0.00	100.00	0
Minimum	0.00	0.00	0.00
Median	92.28	7.72	0.00
Maximum	100.00	100.00	0.00
Mean	75.55	24.45	0.00
Standard deviation	33.95	33.95	0.00

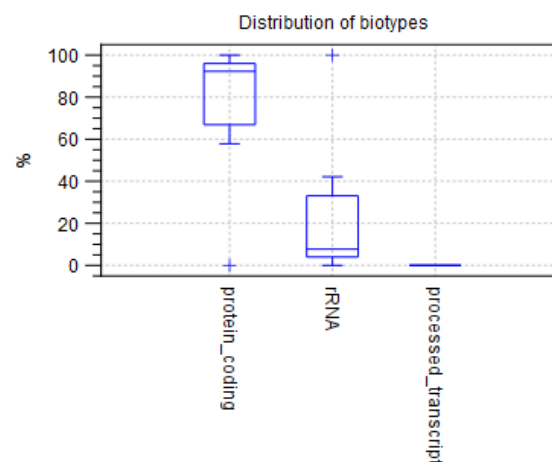
rRNA: >15% of reads mapped to rRNA. This suggests that the poly-A enrichment / rRNA depletion protocol failed. This sample can still be used for differential expression and variant calling, but expression values such as TPM and RPKM may not be comparable to those of other samples.

- If an rRNA depletion kit has been used, check that the kit matches the species being studied.
- In future experiments, check for rRNA depletion prior to library preparation.

	Outliers
protein_coding	sample8
rRNA	sample8

2.6 Distribution of biotypes

The box plot is based on 8 samples.



rRNA: >15% of reads mapped to rRNA. This suggests that the poly-A enrichment / rRNA depletion protocol failed. This sample can still be used for differential expression and variant calling, but expression values such as TPM and RPKM may not be comparable to those of other samples.

- If an rRNA depletion kit has been used, check that the kit matches the species being studied.
- In future experiments, check for rRNA depletion prior to library preparation.

	Outliers
protein_coding	sample8
rRNA	sample8

Figure 37.15: Summary items are reported in tables (left) or box plots (right) when the "Show summary items as plots" option is checked. Cells are highlighted when identified as outliers (yellow), as problematic (pink) or as both outliers and problematic (red). Text for the "rRNA" summary item describes the identified problems. Tables containing the detected outliers are present when the "Include tables for outliers" option is checked.

- At least one sample failed the quality check (🚫). Where none failed, then:
- The quality of at least one sample was uncertain (🟡). Where none were uncertain, then:
- All samples passed the quality check (✅). Or:
- No relevant quality control thresholds have been defined (🔧).

37.6 Create Sample Report

Create Sample Report takes reports relating to a single sample as input and summarizes their contents. A single summary report is generated containing an overview of the analysis of that sample. The report content and its order is configurable. Summary items can be added to a "Quality control" subsection of the sample report, offering a quick overview of the most important quality control metrics.

Creating a sample report

To create a sample report, go to:

Tools | Utility Tools (🔧) | Reports (📄) | Create Sample Report (📄)

To see a list of tools that produce reports that can be used as input, click on the (📘) icon in the top right corner of the input selection wizard (figure 37.16).

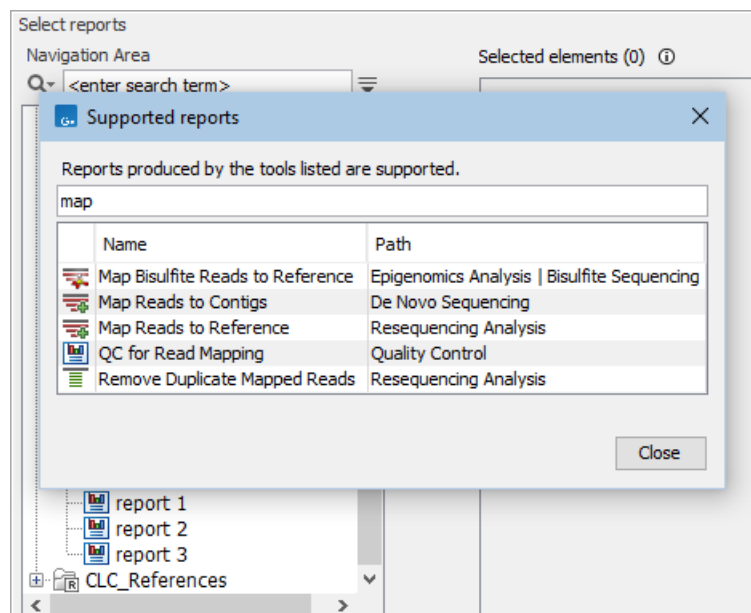


Figure 37.16: Clicking on the info icon at the top right corner of the input selection wizard opens a window showing a list of tools that produce supported reports. Text entered in the field at the top limits the list to just tools with names containing the search term.

One section is included for each report type provided as input, with the type determining the title of the corresponding section. See **Report types and sample report content** below for further details.

Specifying the order of the sections in the report

When more than one report type is provided as input, the order of the sections in the output report is configurable in the "Set order" wizard step:

- **Order of inputs** Use the order that the input reports were specified in.
- **Define order** Explicitly define the section order by moving items up and down in the listing. Defining the order is recommended when the tool is being launched in batch mode with folders of reports provided as the batch units. Doing this avoids reliance on the order of the elements within the folders being the same.

The section order is also reflected in the "Quality control" subsection.

When **Create Sample Report** is included in a workflow, sections are ordered according to the order of the inputs. See section 14.1.3 for information about ordering inputs in workflows.

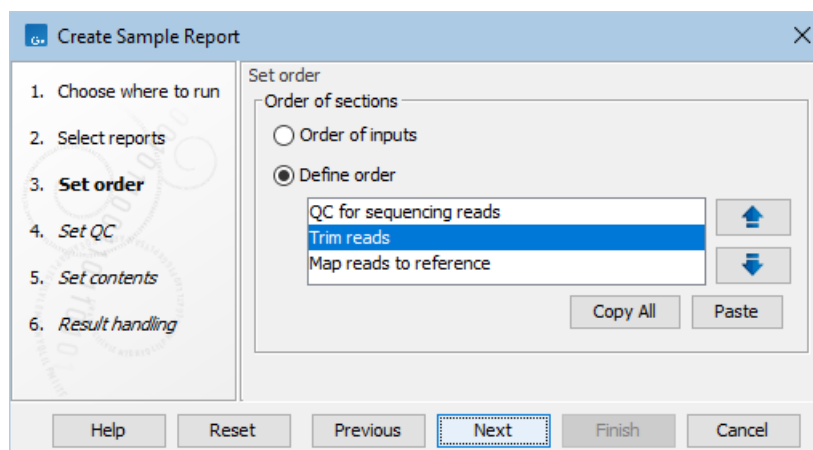


Figure 37.17: When more than one report type is provided as input, the order of the sections can be configured in the "Set order" wizard step.

Defining quality control metrics

The "Quality control" subsection can be configured in the "Set QC" wizard step (figure 37.19). The section will not be included in the sample report when no summary items are added. The **Add** button opens a dialog where summary items can be selected (figure 37.18).

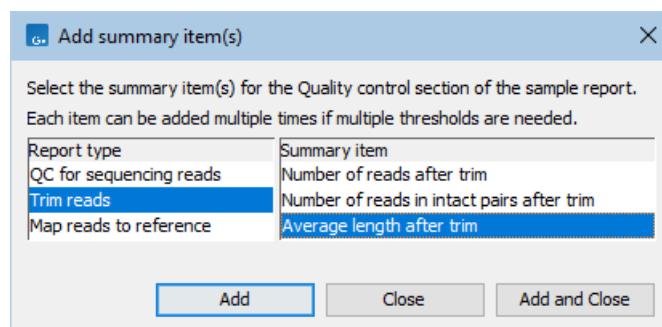


Figure 37.18: Adding summary items to the "Quality control" subsection. To the left are the report types for which summary items can be added. To the right are the available summary items for the selected report type.

Quality control conditions can be specified for summary items if desired (figure 37.19). When a condition is met, that item will be highlighted in green in the "Quality control" subsection of the sample report. If the condition is not met, that item will be highlighted in yellow or red, depending on how the condition has been configured.

To configure a condition for a summary item, supply a threshold value and the operator to use to perform the comparison, and the color to use in the report if the condition is not met.

Summary items can be added more than once, and different conditions can be configured for each instance.

An example: The following would be added to the "Quality control" subsection for the configuration shown in figure 37.19:

- **Number of reads in data set** Since no condition is specified, colors will not be used.

Create Sample Report

1. Choose where to run
2. Select reports
3. Set order
4. **Set QC**
5. Set contents
6. Result handling

Set QC

Quality control conditions (optional)

Summary item	Threshold	When not met
QC for sequencing reads		
Number of reads in data set	≥	
Trim reads		
Number of reads after trim	≥ 260000	Yellow
Average length after trim	≥ 80	Red
Average length after trim	≥ 60	Red
Average length after trim	≤ 200	Green
Map reads to reference		
% of reads mapped	≥ 80	Red

Copy All Paste Add... Clear

Sample name
Sample name
Press Shift + F1 for options

Help Reset Previous Next Finish Cancel

Figure 37.19: Configuring conditions for the "Quality control" subsection. Summary items can be included without conditions (e.g., "Number of reads in data set"), or have multiple conditions defined (e.g. "Average length after trim").

- **Number of reads after trim** The sample will be marked yellow if there are less than 260,000 reads left after trimming, and green otherwise.
- **Average length after trim** Multiple conditions are specified and the sample will be marked
 - Red, if the length is less than 60.
 - Yellow, if the length is less than 80 (but at least 60) or above 200.
 - Green, if the length is between 80 and 200.
- **Percentage of reads mapped** The sample will be marked red if less than 80% of the reads mapped, and green otherwise.

See section 37.6.1 for more details.

Setting the sample name

The name of the sample for which the sample report is created is used when combining sample reports (see section 37.5). This can be customized in the "Set QC" wizard step (figure 37.19):

- **Sample name** If empty and the tool is run
 - From the Tools menu, {basename} is used: the sample name is set to the basename of the first input report, i.e. the prefix of the name until the last (for names that end in (...).

- As part of a workflow, {metadata} is used: the sample name is set to the batch unit identifier.

The sample name can be set to

- An explicit name, such as S234.
- A pattern, such as Sample {basename}. Different patterns are available when running the tool from the Tools menu or as part of a workflow. Consult the tooltip and see section 14.2.4 for details.

Specifying the content of the report

The sections to include in the report are configured in the "Set contents" wizard step (figure 37.20). This includes a general summary section that appears at the top of the report when included.

Where available, individual subsections and summary items can also be specified. Where only some subsections or summary items are excluded, the checkbox for the parent section(s) are highlighted in for visibility.

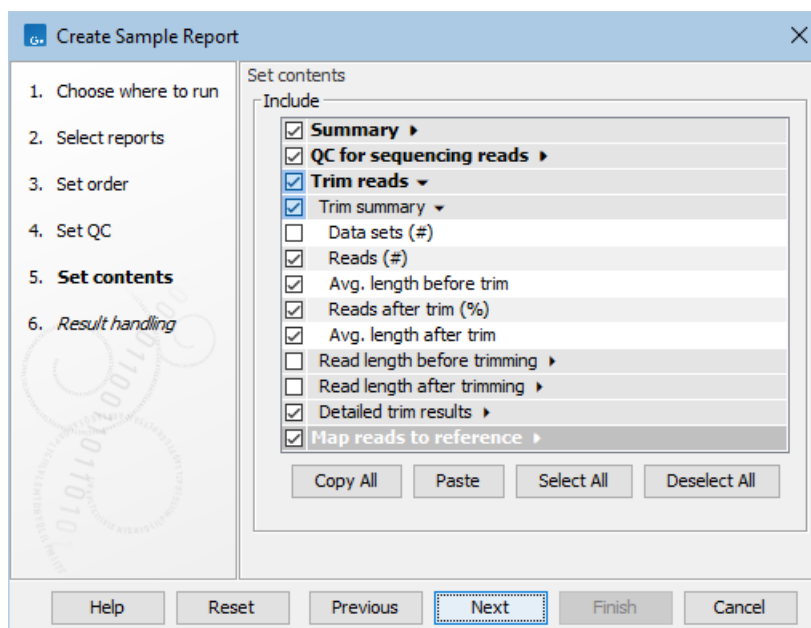


Figure 37.20: The content of the sample report is configured in the "Set contents" wizard step. Sections with a check in the box are included, while those without a check are excluded from the sample report. For visibility, sections where some contents have been excluded have checkboxes highlighted.

Reusing configurations

Configurations defined previously can be used in subsequent runs.

Configurations can be copied in two ways:

- Copy the configuration defined in the relevant wizard step using the **Copy all** button.

- Copy the configuration used in previous runs of the tool from the History (📄) view of the output, described further below.

Copied configurations can be pasted into a text file for later use.

A copied configuration can be pasted into the wizard step using the **Paste** button.

Pasting has the following effects:

- In the "Set QC" wizard step: Conditions referring to summary items included in any input report are *added* to any conditions already configured. Conditions that refer to summary items not present in any input report are not added.
- In the "Set order" and "Set contents" wizard steps: All existing configuration is overwritten with the new information.

The history of a report output by **Create Sample Report** contains the order of the sections (Order reports), the summary items to be added to the "Quality control" subsection (Metrics), and the excluded sections/subsections/summary items (Exclude) (figure 37.21). These can be selected, copied, and then pasted into the "Set order"/"Set QC"/"Set contents" wizard steps wizard steps, respectively, in a subsequent run. Alternatively, the entire history can be selected, copied, and then pasted in each wizard step. Only the relevant configuration is pasted into each step.

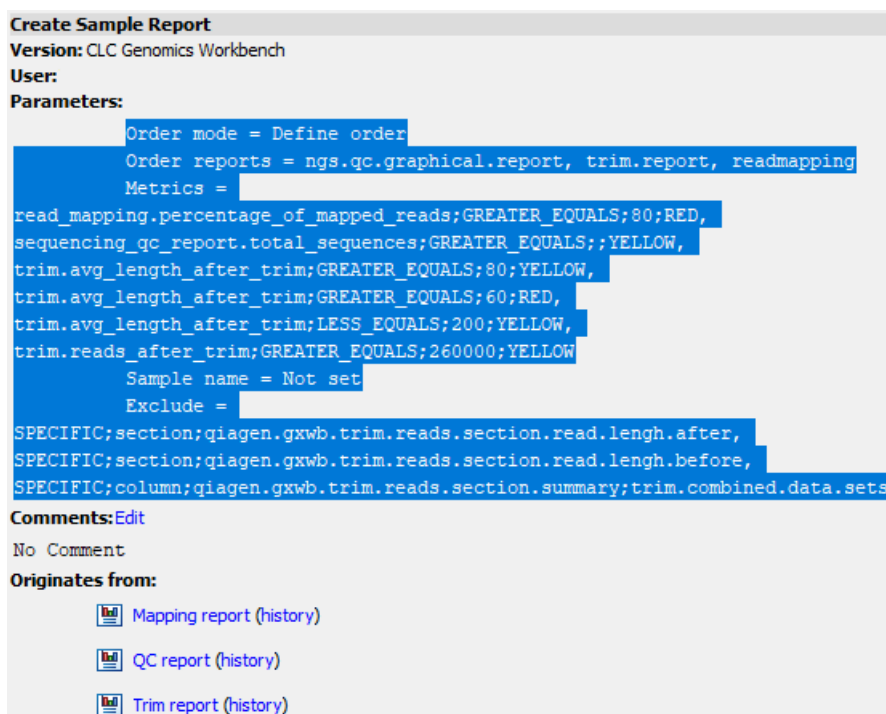


Figure 37.21: The history of a report output by Create Sample Report with the parameters selected, ready to be copied.

Report types and sample report content

Reports that can be supplied as input have a report type. The type of a report affects the placement of its summary information in the output. Specifically:

- Reports with the *same type* that are generated by *the same tool* are summarized into a single section, named according to the type.

This is useful when the aim is to compare the values from those reports, for example results from different samples or different analysis runs. However, if a particular tool has been used more than once in an analysis, for different purposes, then placing the summary of these results in different sections may be desirable. This can be done by editing the report type in some of the reports (see section 37.8.).

- Reports with *different types* are summarized in separate sections, named according to the types.

The report type assigned by a particular tool is unique, so reports generated by different tools have different types.

If reports generated by different tools are later modified so their report types are the same, those reports will still be summarized in different sections, although each of these sections will have the same name.

The type of a report can be seen in the Element Info () view for that report.

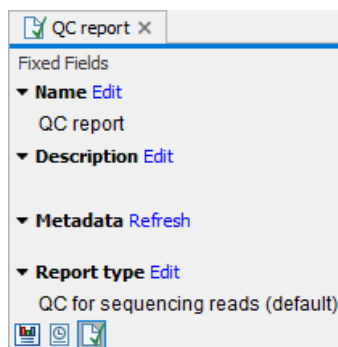



Figure 37.22: The type of a report can be found in the Element Info view of reports that are supported as input for tools that summarize reports.

37.6.1 Create Sample Report output

A single report () is generated, containing summary items and other selected information from the supported reports provided as input.

The report contains one section per input report type, as described in section 37.6. Summary items are displayed in table format.

The summary items are displayed in tables with two columns. The first row indicates the report from which the information has been extracted. The sample report contains links to the input reports and clicking on the report name selects the corresponding report in the Navigation Area.

Table cells are highlighted in pink if they are considered problematic: a potential problem has been identified that is explained underneath the table.

Summary section

By default, sample reports contain a summary section, offering a quick overview of the reports where potential problems have been identified. The summary section is only present if it was included when configuring the report content (see section 37.5) and it only contains summaries those sections/subsections/summary items that are also included in the combined report.

If summary items were added to the "Set QC" wizard step (figure 37.19), a "Quality control" subsection is added to the summary section. The summary items are displayed in a table with three columns: the selected summary item, its value, and the set threshold, if any. When conditions are set, the value is colored accordingly (figure 37.23). If multiple conditions are set for one summary item, the value is colored:

- Red if a condition with a red color is not met. Where all such conditions are met, then:
- Yellow if a condition with a yellow color is not met. Where all such conditions are met, then:
- Green, i.e. all conditions are fulfilled.

1.2 Quality control

Summary item	Value	Threshold
Number of reads in data set	264,200	N/A
Number of reads after trim	252,555	≥ 260,000
Average length after trim	60.00	≥ 60.00
Percentage of reads mapped	99.99	≥ 80.00

Figure 37.23: The "Quality control" subsection.

Note that rounding of values should be taken into account when interpreting results. For example, the extract in figure 37.23 is from a report generated using the configuration shown in figure 37.19. The `Average length after trim` is reported as `60.00`, but is marked in red, even though the condition specified the value should be `>= 60.00`. The true value in the report was, in fact, `59.996`. When necessary, the values without rounding can be checked by exporting the report to JSON format, see section 8.1.10.

The sample report icon summarizes the overall status:

- The sample failed the quality check, because at least one red condition is not met (🚫). Where all such conditions are met, then:
- The quality of the sample is uncertain, because a yellow condition is not met (⚠️). Where all such conditions are met, then:
- The sample passed the quality check, because all conditions are met (✅). Or:
- No relevant quality control thresholds have been defined (🔍).

37.7 Create Report from Table

The **Create Report from Table** tool allows any element that can be exported to "Tab delimited text" to be turned into a report with a single table.

Many element types are accepted as input, including:

- variant track (📊)
- annotation track (📌)
- expression track (📈)
- statistical comparison track (📊)

Note that tables with many rows will create very long reports. Consider filtering the table first. Filtering can be done manually when in table view, see section 9.2. Alternatively, use filtering tools such as **Filter on Custom Criteria**, **Filter Based on Overlap**, **Filter Against Known Variants**.

To run **Create Report from Table**, go to:

Tools | Utility Tools (🔧) | Reports (📄) | Create Report from Table (📄)

After selecting the input element, the columns to include in the report must be defined. Column definitions consist of four parts:

- **Column.** The original name of the column in the table.
- **New name.** The name this column should have in the report. Shorter names are often preferred in reports, because PDF exports have limited width. If the name is left blank, the column will not be renamed.
- **Sort.** Whether the column in the table should be sorted in ascending or descending order. When left blank, the sorting will match that of the input table.
- **Sort order.** The order in which sorting should be applied (only relevant when sorting on multiple columns). Sorting is applied to columns in order from smallest to largest sort order i.e., column with sort order 1 is sorted on before column with sort order 2, and so on. It is not possible to manually enter a sort order. Instead the order is populated automatically according to the order in which columns are chosen for sorting. To change an existing sort order, toggle sorting of the affected columns off and on, such that they receive new sort orders.

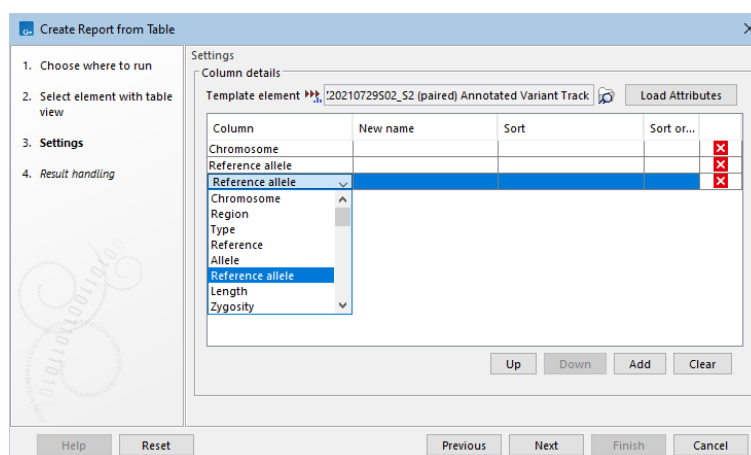




Figure 37.24: Define the columns to be included in the report.

To make defining columns easier, the **Load Attributes** button can be used to populate a dropdown list of the columns found in the element selected in the **Template element** field (figure 37.24). By default, the tool's input is preselected. Use the browse  button to select a different element.

If a template element is not used for populating a dropdown list, columns can be entered by typing directly in the **Column** field.

The **Add** button adds additional columns while pressing the X  button to the right of a column removes it. It is possible to reorder columns using the **Up** and **Down** buttons. The **Clear** button removes all defined columns.

If a column is defined that is not present in the input element, then an empty column with that name will be placed in the report.

Sample and combined reports


The report from this tool can be used in **Create Sample Report**. The name of the section in the sample report can be customized by running the **Modify Report Type** tool prior to running **Create Sample Report**, see section 37.8.

Report sections from this tool cannot be used in **Combine Reports**, because the contents of the tables may be specific to each sample.

37.8 Modify Report Type


Report types affect the placement of information in reports generated by tools designed to summarize other reports. For information about the effect of report type on summary report content see:

- **Combine Reports**: section 37.5.
- **Create Sample Report**: section 37.6.

A report's type can be edited directly in the Element Info () tab or the **Modify Report Type** tool can be used. Both options are described in this section. Note that report types are case sensitive. E.g. 'Trim by Quality' and 'Trim by quality' are interpreted as different types.

The report type assigned by a particular tool is unique, so reports generated by different tools have different types. The term "(default)" at the end of a report type suggests the type has not been modified since the report was created.

Directly updating a report type

The report type is shown in the Element Info () view of that report (figure 37.25). It can be edited in this view by clicking on "Edit".

Running the Modify Report Type tool

To run **Modify Report Type**, go to:

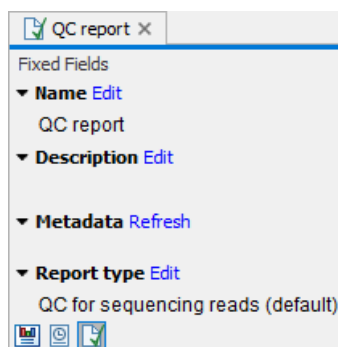


Figure 37.25: Reports types can be seen in the Element Info view of a report.

Tools | Utility Tools (🔧) | Reports (📄) | Modify Report Type (📄)

To see a list of tools that produce supported report types, click on the (ℹ️) icon in the top right corner of the input selection wizard (figure 37.26).

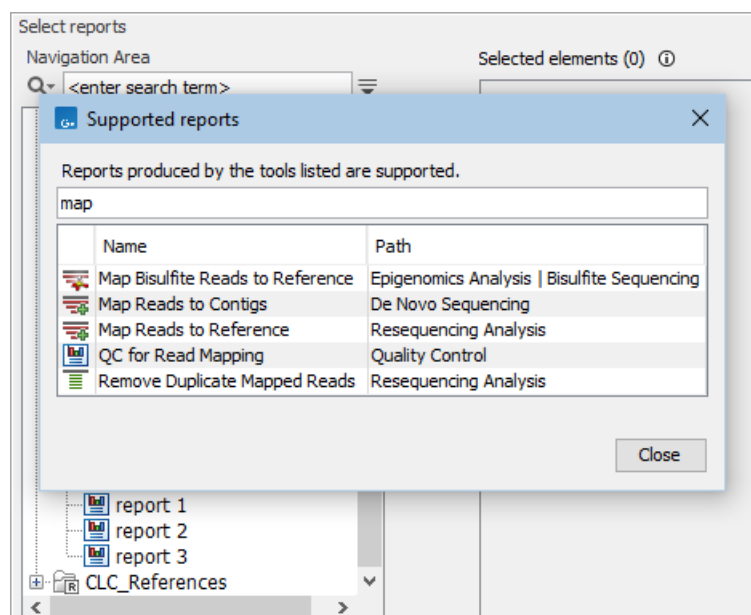


Figure 37.26: Clicking on the info icon at the top right corner of the input selection wizard opens a window showing a list of tools that produce supported report types. Text entered in the field at the top limits the list to just tools with names containing the search term.

The type of the report is set in the **Report type** text field in the "Parameters" wizard step. The type specified here is used for the title of the corresponding section of summary reports generated using this report as input. If left blank, the report type reverts back to the original "(default)" report type.

Modify Report Type outputs a report with the updated report type, but that is otherwise identical to the original report.

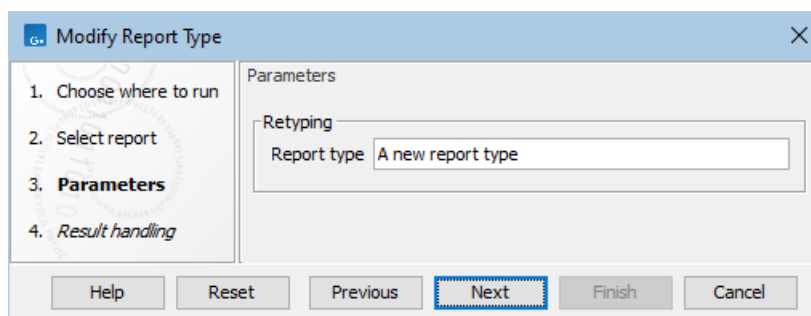


Figure 37.27: Enter the report type to assign in the "Report type" field.

37.8.1 Modifying report types in workflows

Modify Report Type can be added to workflows, where it can be particularly useful in cases where the same tool is present in the workflow multiple times.

Consider the example workflow in figure 37.28.

- Two **Trim Reads** workflow elements, named "Trim by Quality" and "Trim by Ambiguous", to reflect the type of trimming performed.
- Two **Modify Report Type** workflow elements, named "Modify Report Type to Trim by Quality" and "Modify Report Type to Trim by Ambiguous", to reflect which reports it modifies and the type it sets.
- One **Create Sample Report** workflow element, which uses the two trim reads reports with modified types.

Considerations when designing workflows

- **Links within summary reports to input reports**

Where report types are modified, we recommend that the individual reports are output after the report type modification has been done, i.e. link an output element to the **Modify Report Type** element, as illustrated in figure 37.28. This preserves the links found within the summary reports to the input reports with modified types, see section 37.5.1 and section 37.6.1.

- **Quality control summary items**

When adding summary items for the "Quality control" subsection in the sample report:

- Check **Apply to custom report type** and specify the report type, as configured for the respective "Modify Report Type" workflow element, for which the summary item should be used (figure 37.29).
In the example (figure 37.28), distinct summary items are defined for each modified report type (figure 37.30).
- Uncheck **Apply to custom report type** to use the summary item for the reports with default types.

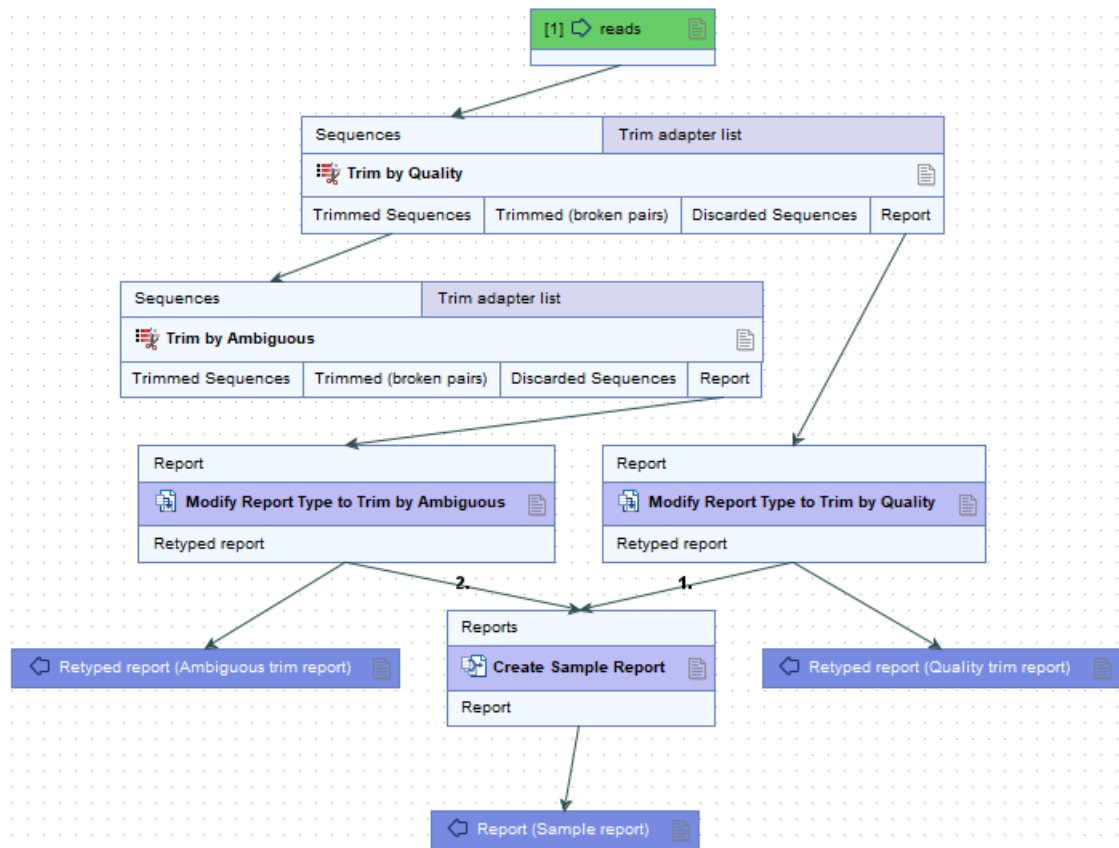


Figure 37.28: An example workflow running two trimming jobs. The name of each trim element is different but the underlying tool is the same, so the reports generated have the same type. The report types are then modified, and reports with the modified type are used as input to the next step.

- **Content of summary reports**

The content of the summary reports can only be defined for the default report types and applies too all reports, even those with a modified type, that originally had that report type (figure 37.31).

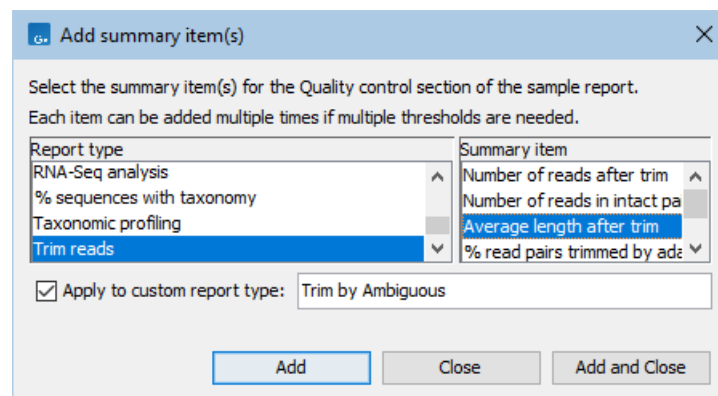


Figure 37.29: Adding summary item for the "Average length after trim" for the trim report with report type "Trim by Ambiguous".

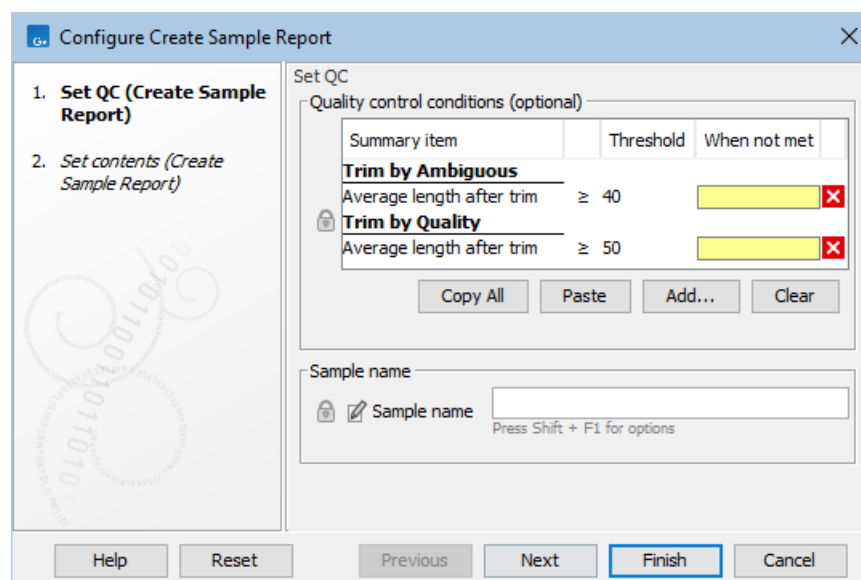


Figure 37.30: Defining different "Average length after trim" thresholds for the trim reports with report types "Trim by Ambiguous" and "Trim by Quality", respectively.

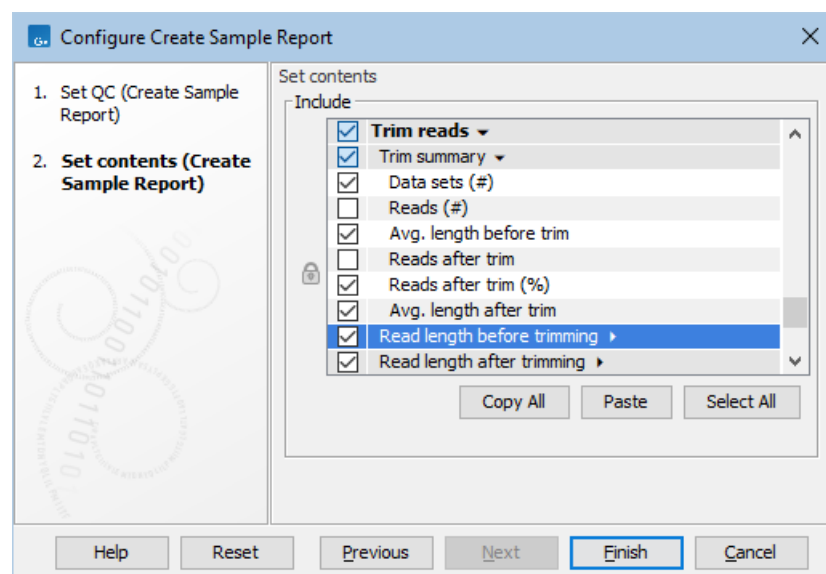


Figure 37.31: Defining the contents for trimming applies to all reports produced by the trimming tool, regardless of their report type.

37.9 Track tools

Utility tools for working with tracks are provided under:

Tools | Utility Tools  | **Tracks** 

These tools are documented in chapter 27.

A quick way to navigate to the documentation for a particular tool is to launch the tool and click on the "Help" button in the bottom, left side of the wizard.

37.10 Create Sequence List

For information on creating sequence lists, please see section 15.1.

37.11 Update Sequence Attributes in Lists

Update Sequence Attributes in Lists updates information about sequences in a Sequence List. Information can be added to existing attributes and new attribute types can be added.

The tool takes attribute information from an Excel file (.xls/xlsx), a comma separated text file (.csv), or a tab separated text file (.tsv). Each sequence is updated with the relevant information by matching the content of a particular column in the file, specified when launching the tool, with the contents of a column of the same name in the Sequence List.

The columns to take information from are specified when launching the tool. Column names are used as attribute names. If a column name matches an existing attribute in the Sequence List, the information from that column can be added to the existing attribute (details below). When a column name does not match an existing attribute, a new attribute is added to the Sequence List.

Additional notes:

- This tool is recommended when updating information for many sequences. However, attributes can also be updated individually, either directly in the Table view of the Sequence List (see section 15.1.3), or, by opening the sequence from the Sequence List and editing attributes in its Element info view. Opening a sequence can be done from the Sequence List view (right-click on the sequence name and choose "Open Sequence") or from the Table view (right-click in the row for that sequence and choose the option "Open This Sequence").
- Attributes relating to characteristics of the sequence itself, such as its length or the start of the sequence, cannot be updated using this tool, nor by directly editing the Sequence List.

To launch the **Update Sequence Attributes in Lists** tool, go to:

Tools | Utility Tools  | **Sequence Lists**  | **Update Sequence Attributes in Lists** 

and select one or more Sequence Lists of the same type (nucleotide or peptide) as input (figure 37.32).

Note: Sequences in all inputs provided will be worked upon as a single entity. A single Sequence List containing all sequences is output.

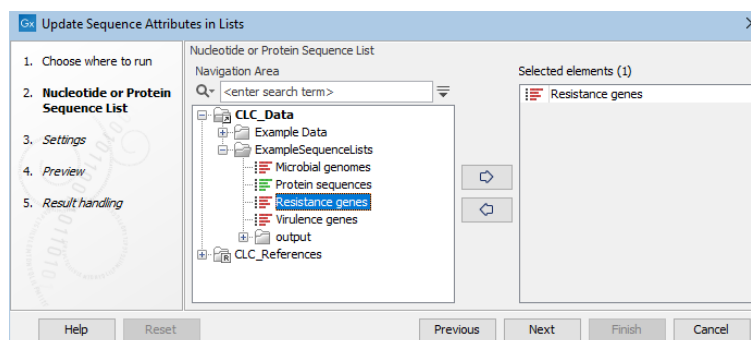


Figure 37.32: Select one or more Sequence Lists as input.

In the Settings wizard step, the file containing attribute information is specified, along with details about how to handle that information (figure 37.33).

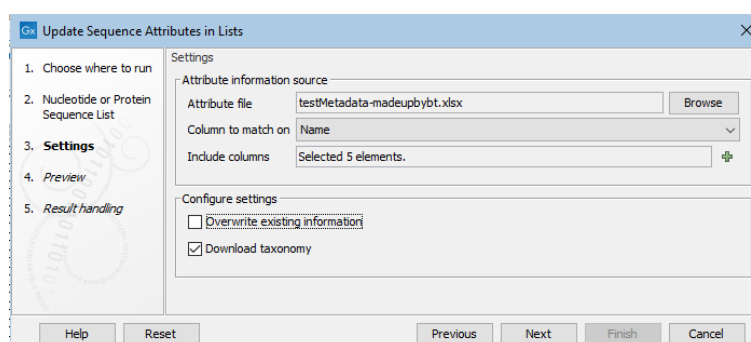


Figure 37.33: Information in the attribute file will be matched with the relevant sequence based on contents of the Name column in the file and in the Sequence List. Five columns containing relevant attribute information have been selected. The option to overwrite existing information has been left unchecked.

Attribute information source fields

- **Attribute file** Select an Excel file (.xls/.xlsx), a comma separated text file (.csv), or a tab separated text file (.tsv) containing attribute information. Column names are used as attribute names, so a header row is required. One column in the file must contain information that can be matched with information already present in the Sequence List (see "Column to match on", below).
- **Column to match on** Specify the column in the attribute file to use to match each row with the relevant sequence(s) in the Sequence List. When a value in this column matches a value in the column of the same name in the Sequence List, information from that row in the file is added to the attribute information for that sequence. Only information from specified columns will be added (see "Include columns", below.)

When matching based on sequence names, the column in the file containing the names must be called Name.

- **Include columns** Select the columns in the file containing the information to be updated or added to the Sequence List as well as the column specified in the "Column to match on" field.

When the name of a column does not match existing attribute name in the Sequence List, a new attribute will be added.

Configure settings checkboxes

- **Overwrite existing information** When this option is checked, existing sequence attribute values will be overwritten by values for the corresponding attributes in the attribute file. When no corresponding value is present in the attributes file, no change is made to the value in the Sequence List.

When left unchecked, existing attribute values in the Sequence List are not overwritten with new information from the file.

- **Download taxonomy** Check this box to download a 7-step taxonomy from the NCBI into an attribute called "Taxonomy". To use this option, there must be a column in the attributes file called `TaxID` containing valid taxonomic identifiers. See the "Column headings and value validation" section below for further details.

The "Taxonomy" attribute will be listed in the Preview wizard step, alongside the columns selected for inclusion.

The result of the choices made in the Settings step are reflected in the Preview wizard step (figure 37.34). In the upper pane is a list of the attribute types to be updated or added, as well as the attribute to be used to match sequences with the relevant information. How particular columns will be handled is indicated in the "Content handling" column, including whether validation will be applied. The columns subject to validation checks are described later in this section.

Shown in the lower pane is a small subset of the incoming information from the attribute file, based on the choices made in the Settings wizard step. Click on the "Previous" button to go back to that step if anything needs to be adjusted.

Column headings and value validation

Certain column names are recognized by the software and validation rules are applied to these. When the contents pass the validation checks, entries in those columns may be further processed.

In most cases, this further processing involves adding hyperlinks to online data resources. However, the contents of columns with the following names trigger different handling:

- **TaxID** When valid taxonomic identifiers are found in a column called `TaxID`, and the **Download taxonomy** checkbox was checked in the Settings wizard step, then a 7-step taxonomy is downloaded from the NCBI.

Examples of valid identifiers for `TaxID` attribute are those found in `/db_xref="taxon` fields in Genbank entries. For example, for `/db_xref="taxon:5833`, the expected value in the `TaxID` column would be 5833.

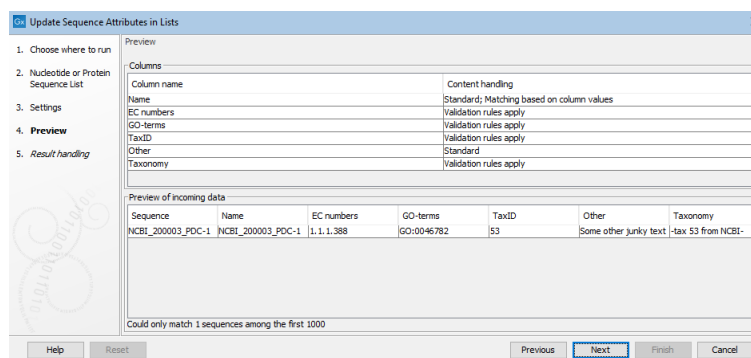


Figure 37.34: The Preview wizard steps shows information about how columns from the attribute file will be handled, and whether any problems were detected. Where validation checks are carried out, if any had failed, a yellow exclamation mark in the bottom pane would be shown for that column. Here, all entries pass. The "Other" column is not subject to validation checks. Only one sequence in the list is being updated in this example.

If a given sequence has a value already set for the Taxonomy attribute, then that existing value remains in place unless the "Overwrite existing information" box was checked in the Settings wizard step.

- **Gene ID** The following identifiers in a Gene ID column are added as attribute values and hyperlinked to the relevant online database:
 - Ensembl Gene IDs
 - HUGO Gene IDs
 - VFDB Gene IDs

Any other values in a Gene ID column are added as attributes to the relevant sequences, but are not hyperlinked to an online data resource. Note that this is different to how other non-validated attribute values are handled, as described below.

Multiple identifiers in a given cell, separated by commas, will be added as multiple Gene ID attributes for the relevant sequence. If any one of those identifiers is not recognized as one of the above types, then none will be hyperlinked.

Other columns where contents are validated are those with the headings listed below. If a value in such a column cannot be validated, it is **not** added nor used to update attributes.

If you wish to add information of this type but do not want this level of validation applied, use a heading other than the ones listed below.

- **COG-terms** COG identifiers
- **Compound AROs** Antibiotic Resistance Ontology identifiers
- **Compound Class AROs** Antibiotic Resistance Ontology identifiers
- **Confers-resistance-to ARO** Antibiotic Resistance Ontology identifiers
- **Drug ARO** Antibiotic Resistance Ontology identifiers

- **Drug Class ARO** Antibiotic Resistance Ontology identifiers
- **EC numbers** EC identifiers
- **GenBank accession** Genbank accession numbers
- **Gene ARO** Antibiotic Resistance Ontology identifiers
- **GO-terms** Gene Ontology (GO) identifiers
- **KO-terms** KEGG Orthology (KO) identifiers
- **Pfam domains** PFAM domain identifiers
- **Phenotype ARO** Antibiotic Resistance Ontology identifiers
- **PubMed IDs** Pubmed identifiers
- **TIGRFAM-terms** TIGRFAM identifiers
- **Virulence factor ID** Virulence Factors of Pathogenic Bacteria identifiers

Updating Location-specific attributes

Location-specific attributes can be created, which are the present for all elements created in that CLC File Location. Such attributes can be updated using the **Update Sequence Attributes in Lists** tool.

Of note when working with such attributes:

- Because these attributes are tied to the Location, they will not appear until the updated Sequence List has been saved.
- The updated Sequence List must be saved to the same File Location as the input for these attributes and their values to appear.
- If this tool is run on an unsaved Sequence List , or using inputs from more than one File Location at the same time, Location-specific attributes will not be updated. Information in the preview pane reflects this.

Location-specific attributes are described in section 3.3.

37.12 Split Sequence List

Split Sequence List can split up nucleotide or peptide sequence lists. The output can be a specified number of lists, lists containing a specified number of sequences, or lists containing sequences with particular attribute values, such as terms in the description,

To launch the **Split Sequence List** tool, go to:

Tools | Utility Tools  | **Sequence Lists**  | **Split Sequence List** 

Select one or more sequence lists as input. If multiple lists are selected, then to process each input list separately, check the "Batch" checkbox. When unchecked, sequences in all the lists are considered together as a single input.

In the next wizard step, the basis upon which to do the splitting and any necessary settings for that option are specified (figure 37.35).

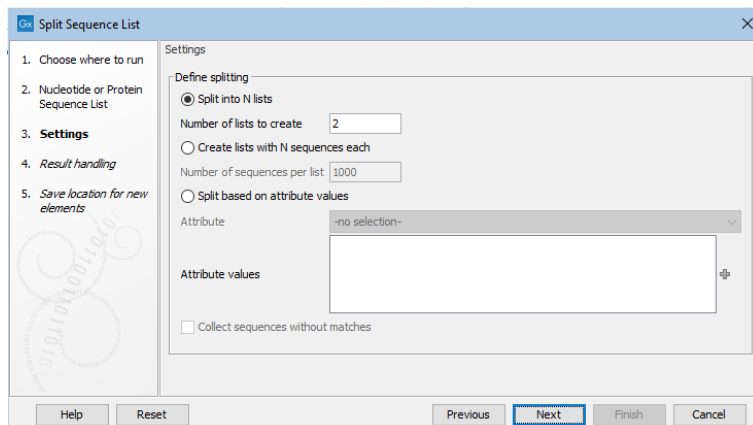


Figure 37.35: Sequence lists can be split into a set number of groups, or into lists containing particular numbers of sequences, or split based on attribute values.

The options are:

- **Split into N lists** In the "Number of lists to create" box, enter the number of lists to split the input into.
- **Create lists with N sequences each** In the "Number of sequences per list" box, enter the relevant number. The final sequence list in the set created may contain fewer than this number.
- **Split based on attribute values** Specify the attribute to split upon from the drop-down list. Columns in the table view of a sequence list equate to the attributes that the list can be split upon.

If no information is entered into the "Attribute values" field, a sequence list is created for each unique value of the specified attribute. If values are provided, a sequence list is created for each of these where at least one sequence has that attribute value. For example, if 3 values are specified, and sequences were found with attributes matching each of these values, 3 sequence lists would be created. If no sequences were found containing 1 of those attribute values, then only 2 sequence lists would be created. Check the "Collect sequences without matches" box to additionally produce a sequence list containing the sequences where no match to a specified value was identified.

Attribute values should be added in a new-line separated list (figure 37.36).

Matching of values is case specific. An asterisk should be added before and/or after the term if the value is part of a longer entry (figure 37.36).

Attribute values are matched in the order they are listed. For example, if the attribute values shown in figure 37.36 were used, and a sequence had a description with both terms in it, that sequence would be placed in the list containing sequences with "Putative" in their descriptions. This is because *Putative* is listed first in the list of values provided to the tool.

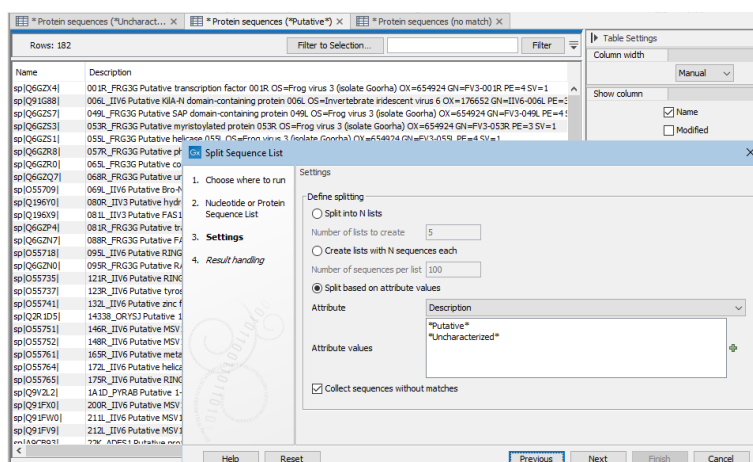


Figure 37.36: With the settings shown here, 3 sequence lists were created. These lists are open in the background tabs shown. One contains sequences with descriptions that include the term "Putative", one contains sequences with descriptions that include the term "Uncharacterized", and one contains sequences containing neither term in the description.

37.13 Subsample Sequence List

The **Subsample Sequence List** tool allows you to sample a nucleotide or protein sequence list by specifying either an absolute number or a percentage of the sequences to extract¹.

Having very high numbers of sequences can be resource intensive without necessarily leading to higher quality results. In such cases subsampling a sequence list can be helpful. Examples of such situations are provided at the end of this section.

To run **Subsample Sequence List**:

Tools | Utility Tools (🔧) | Sequence Lists (📁) | Subsample Sequence List (🔍)

In the first wizard step, select the sequence list to sample from (figure 37.37).

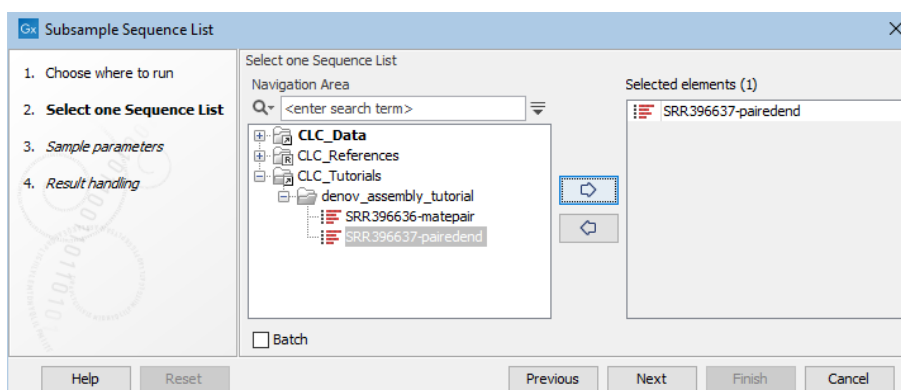


Figure 37.37: Select the sequence list to sample from.

In the next wizard step, configure the tool's settings (figure 37.38).

Sample size options

Sample an absolute number Extract the number of sequences specified in the "Sample size"

¹Prior to CLC Genomics Workbench 22.0, this tool was called Sample Reads

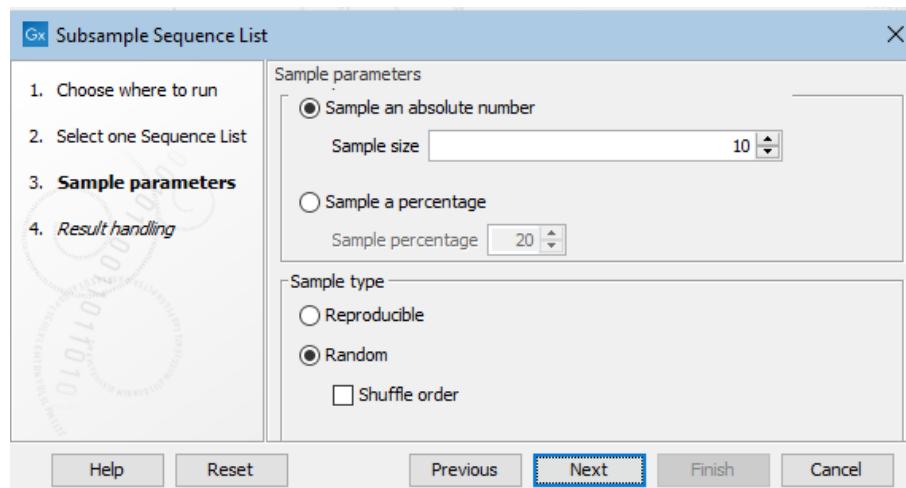


Figure 37.38: Details of the sample of sequences to extract are specified at the Sample parameters step.

field from the sequence list provided as input.

Sample a percentage Extract a percentage of the sequences from the sequence list provided as input. The percentage to extract is specified in the "Sample percentage" field.

Note: When working with paired-end reads, each pair is considered as 2 reads. Only complete pairs will be returned. For example, if a sequence list contains 3 paired reads (6 sequences), and a sample of 50% was requested, then the 2 sequences of a single pair would be returned. If an odd number of reads is requested, an even number would be returned. For example, if 3 reads were requested, 2 would be returned. By contrast, if 6 single end reads were provided as input, 3 reads would be returned in both cases.

Sample type options

Reproducible Return the same set of sequences each time the tool is run with the same input and sample size options specified.

Random Return a different subset of sequences each time the tool is run with the same input and sample size options specified.

Shuffle When unchecked, sequences returned using the Random option will in the same order as they appeared in the original sequence list. When checked, the order of the sequences in the output is shuffled.

Examples of situations to subsample sequence lists

Mapping reads with a very deep coverage can lead to slow response times, especially when the mappings will be investigated in detail. For example, the speed of edit operations in stand-alone mappings after running a tool like **Map Reads to Contigs** can be slowed down by excessive read depth. Subsampling very large sequence lists before mapping can thus lead to better performance, often without loss of important information, depending on the situation.

For de novo assembly, very high coverage in a location increases the probability that the information in that region is interpreted as being associated with sequencing errors. Reducing

the coverage to a maximum average of 100x can thus be a good idea for this application. The following calculation can be used for this purpose:

1. Multiply the estimated size of the genome you intend to assemble by 100 to give the total number of bases to use as input for the de novo assembly.
2. Divide this total number of bases by the average length of the reads.
3. Specify the result of this calculation as the absolute number of reads to sample from the sequence list.

See chapter 35 for further details about running a de novo assembly and mapping reads to contigs.

37.14 Rename Elements

Using **Rename Elements**, you can add or remove characters from element names. You can also replace parts of names, optionally using regular expressions to define the sections to replace.

Replacing spaces or other special characters in element names can be particularly useful before exporting to systems where filenames with those characters are problematic to work with.

Other methods of changing or controlling element names exist: Individual elements can be renamed directly in the Navigation Area by doing a slow double-click on the name, or selecting the element and choosing the menu item Edit | Rename. In workflows, the naming of outputs can be configured, as described in section 14.2.4.

To launch the **Rename Elements** tool, go to:

Tools | Utility Tools (🔧) | Renaming (🔄) | Rename Elements (🔄)

In the first wizard step, select the elements to rename (figure 37.39).

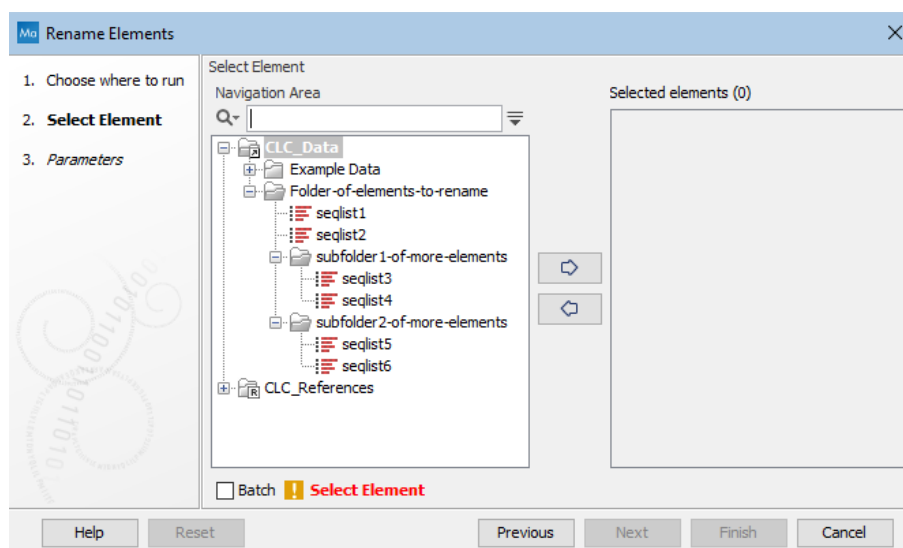


Figure 37.39: Select the elements or folders to be renamed.

Alternatively, you can right-click and select "Add folder contents" or, select "Add folder contents (recursively)" to rename all elements in a folder structure recursively. (figure 37.40).

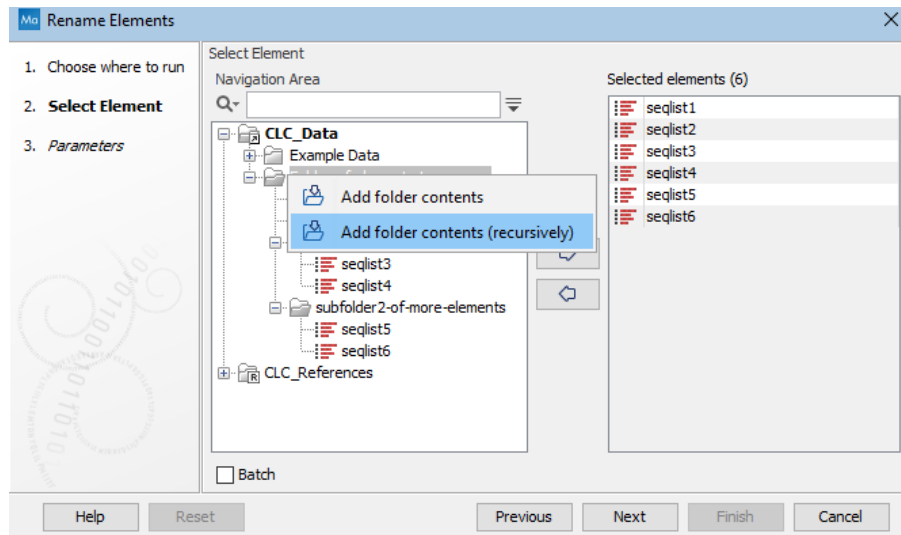


Figure 37.40: Right-click for options to add the contents of folders as inputs. Here, the "Add folder contents (recursively)" option was selected. If "Add folder contents" had been selected, only the elements `seqlist1` and `seqlist2` would have been added to the Selected elements list on the right.

When multiple elements are supplied as input, each will be renamed. There is no need to check the Batch checkbox for this. However, if you wish to rename only certain elements from the full selection, then checking the Batch box provides the opportunity to do that. When checked, the next wizard step shows the batch overview, where elements can be explicitly included or excluded from those to be renamed, based on text patterns. This step is described in more detail at section 12.3).

Checking the Batch checkbox for this tool also has the following effect when a folder is selected as input:

- With the Batch option checked, the top level contents of that folder will be renamed.
- With the Batch option unchecked, the folder itself will be renamed.

Important notes about renaming elements

- Renaming elements **cannot be undone**. To alter the names further, the elements must be renamed again.
- The renaming action is recorded in the **History** (📄) for the element, but the "Originates from" entries lists the changed element name, rather than the original element name.

Renaming options

This wizard step presents various options for the renaming action (figure 37.41). The **Rename Elements** is used for illustration in this section, but the options are the same for the **Rename Sequences in Lists** tool.

- **Add text to name** Select this option to add text at the beginning or the end of the existing name.

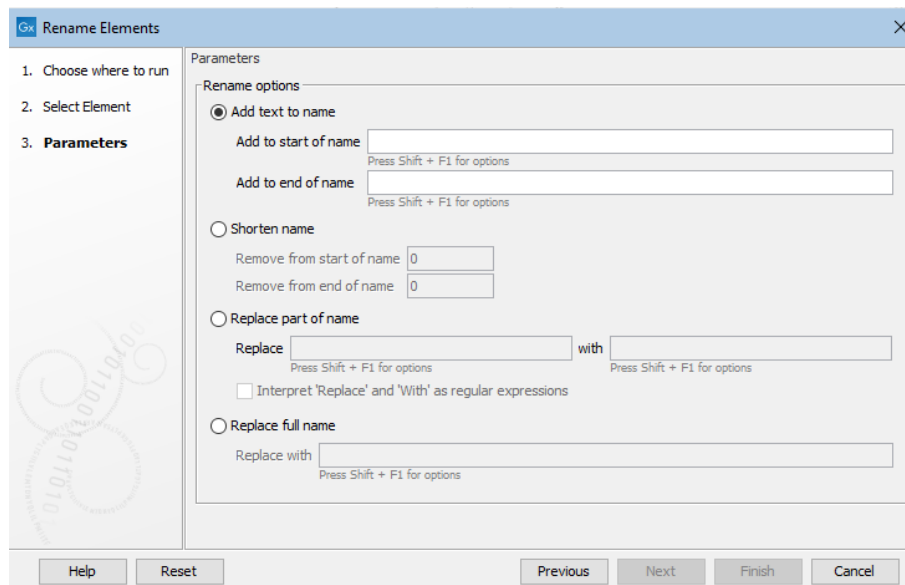


Figure 37.41: Text can be added, removed or replaced in the existing names.

You can add text directly to these fields, and you can also include placeholders to indicate certain types of information should be added. Multiple placeholders can be used, in combination with other text if desired (figure 37.42). The available placeholders are:

- {name} The current name of the element. Usually used when defining a new naming pattern for replacing the full name of elements.
- {shortname} Truncates the original name to 10 characters. Usually used when replacing the full names of elements.
- {Parent folder} The name of the folder containing the element.
- {today} Today's date in the form YYYY-MM-DD
- {enumeration} Adds a number to the name. This is intended for use when multiple elements are selected as input. Each is assigned a number, starting with the number 1, (added as 0000001) for the first element that was selected, 2 (added as 0000002) for the second element selected, and so on.

Click in a field and use Shift + F1 (Shift + Fn + F1 on Mac) to show the list of available placeholders, as shown in figure 37.42. Click on a placeholder in that list to have it entered into the field.

- **Shorten name** Select this option to shorten a name by removing a specified number of characters from the start and/or end of the name.
- **Replace part of name** Select this option to specify text or regular expressions to define parts of the element names to be replaced. By default, the text entered in the fields is interpreted literally. Check the "Interpret 'Replace' as regular expression" option to indicate that the terms provided in the "Replace" field should be treated as regular expressions. Information on regular expressions can be found at <https://docs.oracle.com/javase/tutorial/essential/regex/>.

By clicking in either the "Replace" or "with" field and pressing Shift + F1 (Shift + Fn + F1 on Mac), a drop down list of renaming possibilities is presented. The options listed for

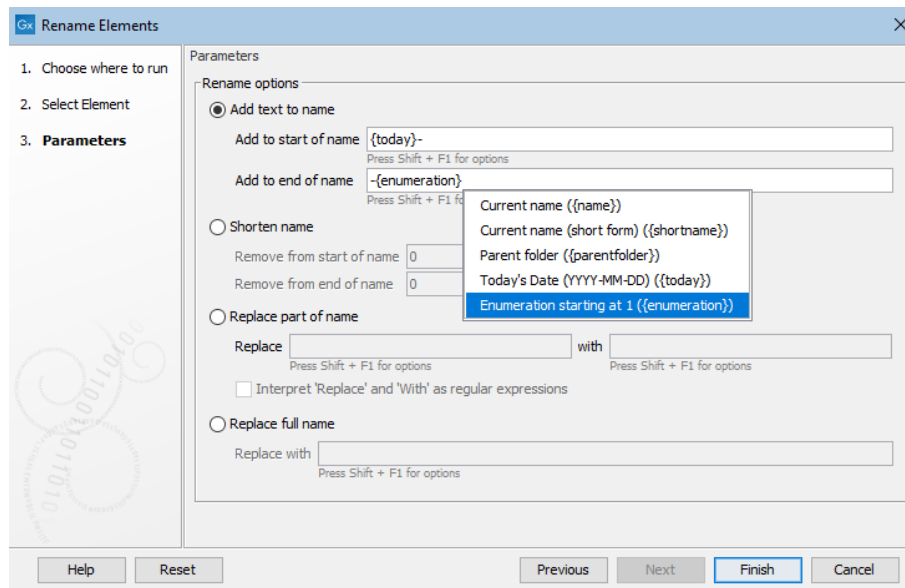


Figure 37.42: Click on Shift+F1 (Shift + Fn + F1 on Mac) to reveal a drop-down list of placeholders that can be used. Here, today's date and a hyphen would be prepended, and a hyphen and ascending numeric value appended, to the existing names.

the Replace field are some commonly used regular expressions. Other standard regular expressions are also admissible in this field. The placeholders described above for adding text to names are available for use in the "with" field. **Note:** We recommend caution when using these placeholders in combination with regular expressions in the Replace field. Please run a small test to ensure it works as you intend.

- **Replace full name** Select this option to replace the full element name. Text and placeholders can be used in this field. The placeholders described above for adding text to names are available for use. Use Shift + F1 (Shift + Fn + F1 on Mac) to see a list.

Examples using regular expressions to replace parts of element names

The following are examples to illustrate using regular expressions for renaming. The same principles apply for renaming using the **Rename Elements** and **Rename Sequences in Lists** tools.

- Replacing part of an element's name with today's date and an underscore. Details are shown in figure 37.43.
- Rename using the first 4 non-whitespace characters from names that start with 2 characters, then have a space, then have multiple characters following, such as 1N R1_0001.
 - Check the box beside "Interpret 'Replace' and 'with' as Java regular expressions".
 - Enter `([\w]{2})\s([\w]{2}).*` into the "Replaces" field.
 - Enter `$1$2` into the "with" field.
- Keep only the last 4 characters of the name.
 - Check the box beside "Interpret 'Replace' and 'with' as Java regular expressions".

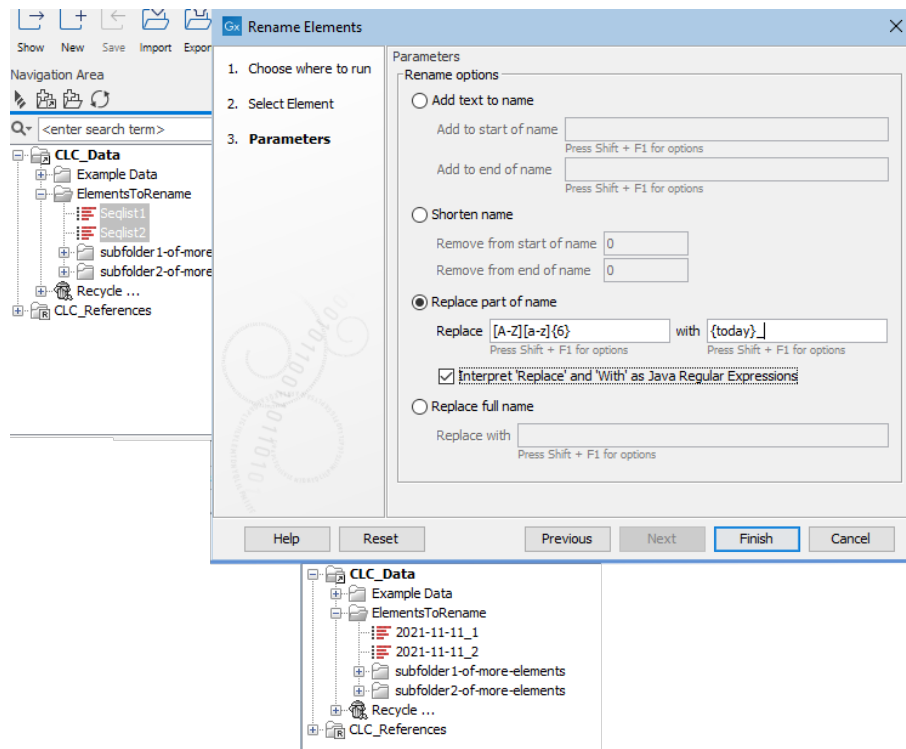


Figure 37.43: Elements with names *Seqlist1* and *Seqlist2* each start with a capital letter, followed by 6 small letters. Using the settings shown, their names are updated to be the date the renaming was done, followed by a hyphen, and the remaining parts of the original name, here, the integer at the end of each name.

- Enter `(.*) (. {4} $)` into the "Replaces" field.
- Enter `$2` into the "with" field.
- Replace a set pattern of text with the name of the parent folder. Here, we start with the name `p140101034_1R_AMR` and replace the first letter and 9 numbers with the parent folder name.
 - Check the box beside "Interpret 'Replace' and 'with' as Java regular expressions".
 - Enter `([a-z] \d {9}) (.*)` into the "Replaces" field.
 - Enter `{parentfolder} $2` into the "with" field.
- Rename using just the text between the first and second underscores in `1234_sample-code_5678`.
 - Check the box beside "Interpret 'Replace' and 'with' as Java regular expressions".
 - Enter `(^[^_]+) _ ([^_]+) _ (.*)` into the "Replaces" field.
 - Enter `$2` into the "with" field.

37.15 Rename Sequences in Lists

Using **Rename Sequences in Lists**, you can add or remove characters from the names of sequences within sequence lists. You can also replace parts of sequence names, optionally using regular expressions to define the sections to replace.

Individual sequences in a sequence list can also be renamed manually. Open the sequence list, right-click on the sequence name to change, and select the "Rename Sequence..." option.

To launch the **Rename Sequences in Lists** tool, go to:

Tools | Utility Tools (🔧) | Renaming (🔄) | Rename Sequences in Lists (📄)

In the first wizard step, select a sequence list. To rename sequences in multiple sequence lists using the same renaming pattern, check the Batch checkbox (figure 37.44).

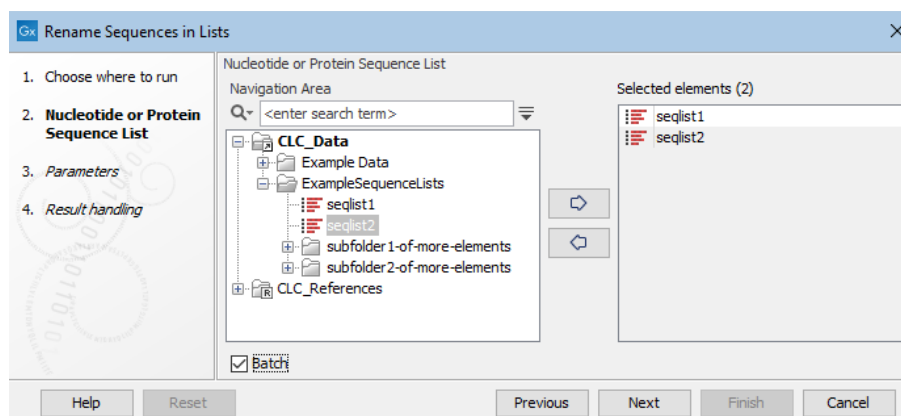


Figure 37.44: When the sequences in more than one list should be renamed, check the Batch checkbox.

The text "Renamed" is added within parentheses to the name of sequence lists output by this tool. E.g. with an input called "seqlist2", the sequence list containing the renamed sequences will be called "seqlist2 (Renamed)".

Renaming options

This wizard step presents various options for the renaming action (figure 37.45). The **Rename Elements** is used for illustration in this section, but the options are the same for the **Rename Sequences in Lists** tool.

- **Add text to name** Select this option to add text at the beginning or the end of the existing name.

You can add text directly to these fields, and you can also include placeholders to indicate certain types of information should be added. Multiple placeholders can be used, in combination with other text if desired (figure 37.46). The available placeholders are:

- {name} The current name of the element. Usually used when defining a new naming pattern for replacing the full name of elements.
- {shortname} Truncates the original name to 10 characters. Usually used when replacing the full names of elements.
- {Parent folder} The name of the folder containing the element.
- {today} Today's date in the form YYYY-MM-DD
- {enumeration} Adds a number to the name. This is intended for use when multiple elements are selected as input. Each is assigned a number, starting with the number 1, (added as 0000001) for the first element that was selected, 2 (added as 0000002) for the second element selected, and so on.

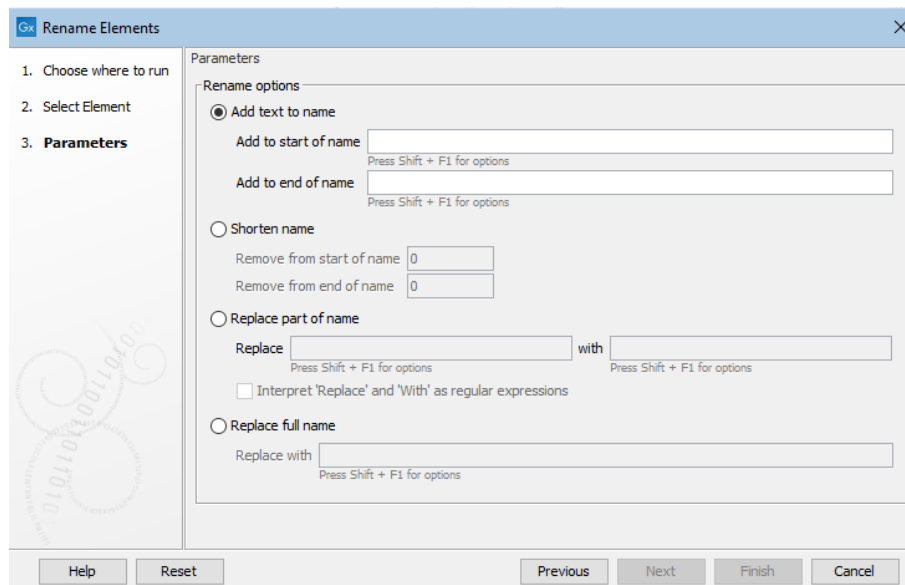


Figure 37.45: Text can be added, removed or replaced in the existing names.

Click in a field and use Shift + F1 (Shift + Fn + F1 on Mac) to show the list of available placeholders, as shown in figure 37.46. Click on a placeholder in that list to have it entered into the field.

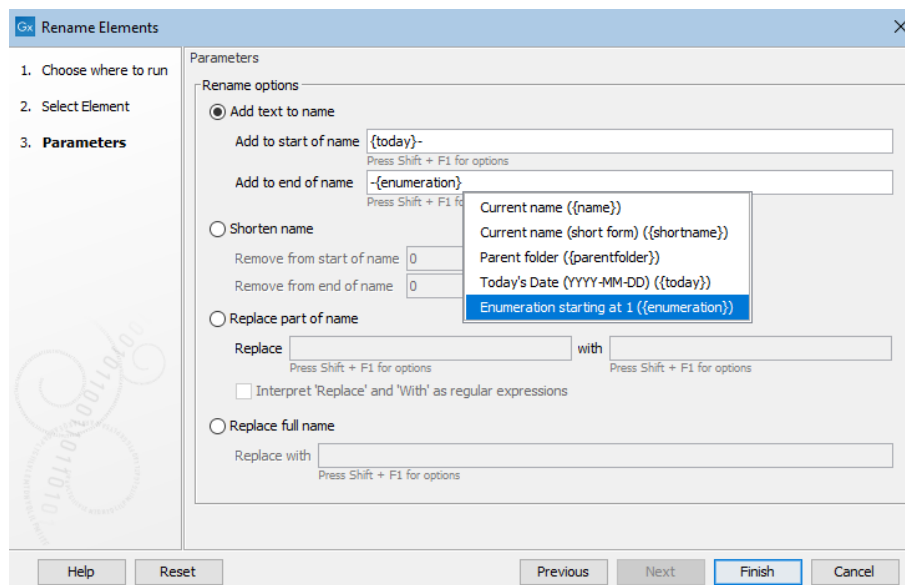


Figure 37.46: Click on Shift+F1 (Shift + Fn + F1 on Mac) to reveal a drop-down list of placeholders that can be used. Here, today's date and a hyphen would be prepended, and a hyphen and ascending numeric value appended, to the existing names.

- **Shorten name** Select this option to shorten a name by removing a specified number of characters from the start and/or end of the name.
- **Replace part of name** Select this option to specify text or regular expressions to define parts of the element names to be replaced. By default, the text entered in the fields is interpreted literally. Check the "Interpret 'Replace' as regular expression" option to indicate that the terms provided in the "Replace" field should be treated as regular

expressions. Information on regular expressions can be found at <https://docs.oracle.com/javase/tutorial/essential/regex/>.

By clicking in either the "Replace" or "with" field and pressing Shift + F1 (Shift + Fn + F1 on Mac), a drop down list of renaming possibilities is presented. The options listed for the Replace field are some commonly used regular expressions. Other standard regular expressions are also admissible in this field. The placeholders described above for adding text to names are available for use in the "with" field. **Note:** We recommend caution when using these placeholders in combination with regular expressions in the Replace field. Please run a small test to ensure it works as you intend.

- **Replace full name** Select this option to replace the full element name. Text and placeholders can be used in this field. The placeholders described above for adding text to names are available for use. Use Shift + F1 (Shift + Fn + F1 on Mac) to see a list.

Examples using regular expressions to replace parts of element names

The following are examples to illustrate using regular expressions for renaming. The same principles apply for renaming using the **Rename Elements** and **Rename Sequences in Lists** tools.

- Replacing part of an element's name with today's date and an underscore. Details are shown in figure 37.47.

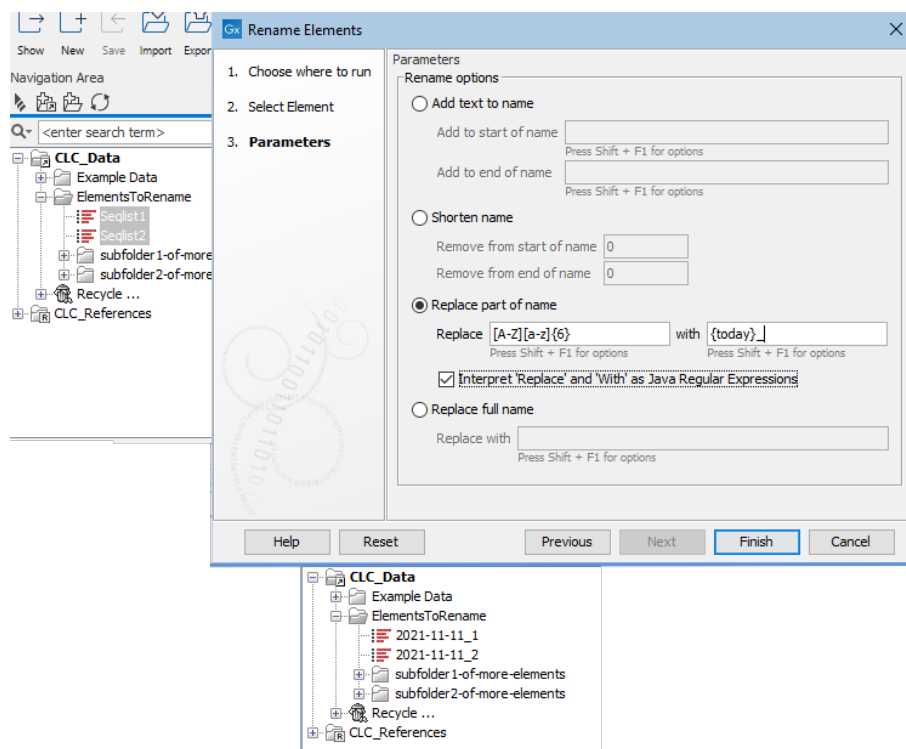


Figure 37.47: Elements with names *Seqlist1* and *Seqlist2* each start with a capital letter, followed by 6 small letters. Using the settings shown, their names are updated to be the date the renaming was done, followed by a hyphen, and the remaining parts of the original name, here, the integer at the end of each name.

- Rename using the first 4 non-whitespace characters from names that start with 2 characters, then have a space, then have multiple characters following, such as `1N R1_0001`.


- Check the box beside "Interpret 'Replace' and 'with' as Java regular expressions".
 - Enter `([\w]{2})\s([\w]{2})\.*` into the "Replaces" field.
 - Enter `$1$2` into the "with" field.
- Keep only the last 4 characters of the name.
 - Check the box beside "Interpret 'Replace' and 'with' as Java regular expressions".
 - Enter `(.*)\.{4}$` into the "Replaces" field.
 - Enter `$2` into the "with" field.
- Replace a set pattern of text with the name of the parent folder. Here, we start with the name `p140101034_1R_AMR` and replace the first letter and 9 numbers with the parent folder name.
 - Check the box beside "Interpret 'Replace' and 'with' as Java regular expressions".
 - Enter `([a-z]\d{9})(.*)` into the "Replaces" field.
 - Enter `{parentfolder}$2` into the "with" field.
- Rename using just the text between the first and second underscores in `1234_sample-code_5678`.
 - Check the box beside "Interpret 'Replace' and 'with' as Java regular expressions".
 - Enter `(^[^_]+)_[^_]+(.*)` into the "Replaces" field.
 - Enter `$2` into the "with" field.

Chapter 38

Legacy tools

Contents


38.1 Correct Long Reads	1230
38.1.1 Correct Long Reads output	1231

The documentation in this section is for tools that have been deprecated and that will be retired in a future version. In most cases, deprecated tools can be found in the **Legacy Tools**  folder under the Workbench Tools menu, with "(legacy)" appended to their names to highlight their status.

We recommend redesigning workflows containing any of these tools to remove them. Where a new tool has been introduced to take the deprecated tool's place, please try including the new tool.

If you have concerns about the retirement of particular tools in this section, please contact QIAGEN Bioinformatics Support team at ts-bioinformatics@qiagen.com.

38.1 Correct Long Reads

This tool has been deprecated and will be retired in a future version of the software. It has been moved to the **Legacy Tools**  folder under the Tools menu, and its name has "(legacy)" appended to it.

The Correct Long Reads tool was originally designed to enable the correction of error-prone long reads. Over the past years, the quality of long reads has improved dramatically, to the point where correction is no longer needed. In addition, the most common applications such as **Map Long Reads to Reference** and **De Novo Assemble Long Reads** can now handle the older error-prone reads without prior correction. Therefore, the tool will be retired.

If you have concerns about the future retirement of this tool, please contact QIAGEN Bioinformatics Support team at ts-bioinformatics@qiagen.com.

The Correct Long Reads tool enables the correction of a set of error-prone long reads by finding overlaps between the reads, and performing a consensus error correction using Racon [[Vaser et al., 2017](#)].

Note that if the aim is to create an assembly from a set of error-prone long reads it is not recommended to run **Correct Long Reads** prior to **De Novo Assemble Long Reads**. For other applications, however, it may be beneficial to correct the reads before further analysis.

To run the Correct Long Reads tool, go to:

Tools | Legacy Tools | Correct Long Reads (legacy) (RAP)

Select one or more sequence lists containing reads.

In the next dialog, set the following parameters (figure 38.1):

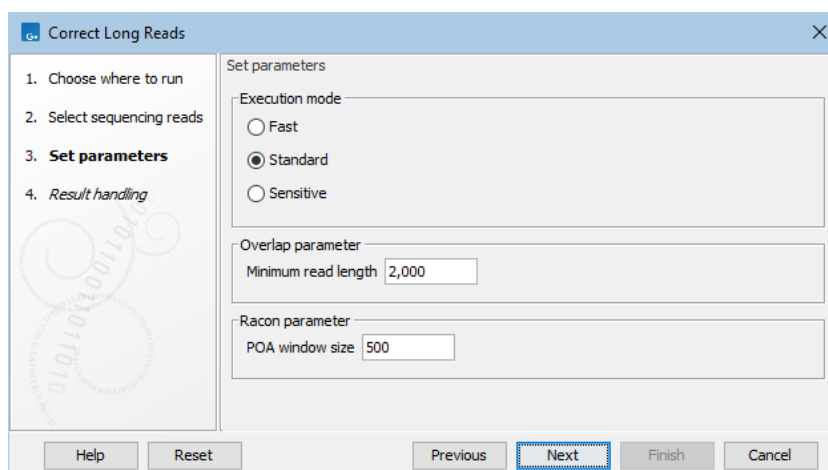


Figure 38.1: *Correct Long Reads parameters*

- **Execution mode.**

- **Fast.** An overlap mapper computes overlaps using a coarse all-to-all sequence alignment, prior to running Racon. This option will use the least amount of memory.
- **Standard.** Similar to the *Fast* , but with a higher sensitivity.
- **Sensitive.** Minimap2 is used to compute a more precise all-to-all sequence alignment prior to running Racon. This option should only be used for low-coverage datasets.

- **Minimum read length.** Reads shorter than this value will not be included in the correction process.

- **POA window size.** The window size for which Racon computes partial order alignments (POA). A larger window size enhances the ability to capture more global structure during the polishing process, but it also increases the memory requirement.

38.1.1 Correct Long Reads output

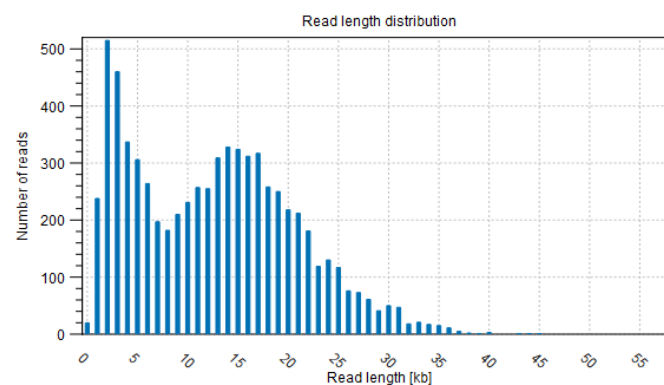
In addition to the sequence list of corrected reads, a summary report is available via the output option **Create report**.

Correct Long Reads report

An example of the top part of the Correct Long Reads summary report is shown in figure 38.2.

1 Input statistics

Nucleotide	Count	Frequency (%)
Adenine (A)	18,412,271	20.66
Cytosine (C)	25,880,634	29.04
Guanine (G)	26,068,337	29.25
Thymine (T)	18,749,472	21.04
Reads		7,039
Minimum		232
Maximum		57,649
Average		12,660
N50		17,593
N90		7,743
Total		89,110,714

**2 Output statistics**

Nucleotide	Count	Frequency (%)
Adenine (A)	7,734,576	21.86
Cytosine (C)	9,991,371	28.24
Guanine (G)	9,925,504	28.05
Thymine (T)	7,734,838	21.86
Reads		3,343
Minimum		1,973
Maximum		40,760
Average		10,585
N50		14,537
N90		5,383

Figure 38.2: *Correct Long Reads report.*

The report is divided into two sections, *Input statistics* and *Output statistics*, both containing the following information:

- **Reads.** Number of reads.
- **Minimum, Maximum, Average.** Minimum, maximum and average sequence length.
- **N50.** The length of the shortest sequence in sets of sequences of equal length or longer, where the summed length of sequences is at least 50% of the total sequence length.
- **N90.** The length of the shortest sequence in a set of sequences of equal length or longer, where the summed length of sequence is at least 90% of the total sequence length. N90 will be equal to or smaller than N50.
- **Total.** The number of bases in the sequences.

- **Sequence length distribution.** A graph depicting the number of sequences found at a specific length.

Part V

Appendix

Appendix A

Use of multi-core computers

The tools listed below can make use of multi-core CPUs. This does not necessarily mean that all available CPU cores are used throughout the analysis, but that these tools benefit from running on computers with multiple CPU cores.

- Amino Acid Changes
- Annotate from Known Variants
- Annotate with Conservation Scores
- Annotate with Exon Numbers
- Annotate with Flanking Sequences
- Basic Variant Detection
- BLAST (will not scale well on many cores)
- Call Methylation Levels
- Copy Number Variant Detection (Targeted)
- Create Alignment
- Create RRBS-fragment Track
- Demultiplex Reads
- De Novo Assembly
- De Novo Assemble Long Reads
- Detect and Refine Fusion Genes
- Differential Expression for RNA-Seq
- Differential Expression in Two Groups
- Filter against Known Variants
- Filter based on Overlap
- Fixed Ploidy Variant Detection
- GO Enrichment Analysis
- Identify Enriched Variants in Case vs Control Samples
- InDels and Structural Variants

- K-mer Based Tree Construction

- Link Variants to 3D Protein Structure
- Local Realignment
- Low Frequency Variant Detection

- Map Bisulfite Reads to Reference
- Map Long Reads to Reference
- Map Reads to Contigs
- Map Reads to Reference
- Maximum Likelihood Phylogeny
- Merge Annotation Tracks
- Merge Variant Tracks
- Model Testing

- Polish Contigs with Reads
- Predict Splice Site Effect

- QC for Read Mapping
- QC for Sequencing Reads
- QC for Targeted Sequencing

- Remove Variants Present in Control Reads
- Remove Marginal Variants
- RNA-Seq Analysis
- RNA-Seq Analysis for Long Reads

- Structural Variant Caller for Long Reads

- Target Region Coverage Analysis
- Trim Reads
- Trio Analysis

- Whole Genome Coverage Analysis

Appendix B

Graph preferences

This section explains the view settings of graphs. The **Graph preferences** at the top of the **Side Panel** includes the following settings:


- **Lock axes** This will always show the axes even though the plot is zoomed to a detailed level.
- **Frame** Shows a frame around the graph.
- **Show legends** Shows the data legends.
- **Tick type** Determine whether tick lines should be shown outside or inside the frame.
- **Tick lines at** Choosing Major ticks will show a grid behind the graph.
- **Horizontal axis range** Sets the range of the horizontal axis (x axis). Enter a value in **Min** and **Max**, and press Enter. This will update the view. If you wait a few seconds without pressing Enter, the view will also be updated.
- **Vertical axis range** Sets the range of the vertical axis (y axis). Enter a value in **Min** and **Max**, and press Enter. This will update the view. If you wait a few seconds without pressing Enter, the view will also be updated.

Certain types of graphs can additionally contain the following settings:

- **X-axis at zero.** This will draw the x axis at $y = 0$. Note that the axis range will not be changed.
- **Y-axis at zero.** This will draw the y axis at $x = 0$. Note that the axis range will not be changed.
- **Show as histogram.** For some data-series it is possible to see the graph as a histogram rather than a line plot.

The representation of the data is configured in the bottom area, e.g. line widths, dot types, colors, etc. For graphs of multiple data series, the series to apply the settings to can be selected from a drop down list.

- **Dot type** Can be None, Cross, Plus, Square, Diamond, Circle, Triangle, Reverse triangle, or Dot.
- **Dot color** Click the color box to select a color.
- **Line width** Thin, Medium or Wide
- **Line type** None, Line, Long dash or Short dash
- **Line color** Click the color box to select a color.

The graph and axes titles can be edited simply by clicking with the mouse. These changes will be saved when you **Save** () the graph - whereas the changes in the **Side Panel** need to be saved explicitly (see section 4.6).

Appendix C

BLAST databases

Several databases are available at NCBI, which can be selected to narrow down the possible BLAST hits.

C.1 Peptide sequence databases

- **nr.** Non-redundant GenBank CDS translations + PDB + SwissProt + PIR + PRF, excluding those in env_nr.
- **refseq.** Protein sequences from NCBI Reference Sequence project <https://www.ncbi.nlm.nih.gov/RefSeq/>.
- **swissprot.** Last major release of the SWISS-PROT protein sequence database (no incremental updates).
- **pat.** Proteins from the Patent division of GenBank.
- **pdb.** Sequences derived from the 3-dimensional structure records from the Protein Data Bank.
- **env_nr.** Non-redundant CDS translations from env_nt entries.
- **tsa_nr.** Transcriptome Shotgun Assembly db proteins. (NCBI BLAST only)
- **month.** All new or revised GenBank CDS translations + PDB + SwissProt + PIR + PRF released in the last 30 days. (Create Protein Report only)

C.2 Nucleotide sequence databases

- **nr.** All GenBank + EMBL + DDBJ + PDB sequences (but no EST, STS, GSS, or phase 0, 1 or 2 HTGS sequences). No longer "non-redundant" due to computational cost.
- **Human G+T.** Human genomic and transcript sequences
- **Mouse G+T.** Mouse genomic and transcript sequences
- **refseq_rna.** mRNA sequences from NCBI Reference Sequence Project.

- **refseq_genomic.** Genomic sequences from NCBI Reference Sequence Project.
- **refseq_representative_genomes.** Representative sequences from NCBI Reference Sequence Project.
- **est.** Database of GenBank + EMBL + DDBJ sequences from EST division.
- **est_human.** Human subset of est.
- **est_mouse.** Mouse subset of est.
- **est_others.** Subset of est other than human or mouse.
- **gss.** Genome Survey Sequence, includes single-pass genomic data, exon-trapped sequences, and Alu PCR sequences.
- **htgs.** Unfinished High Throughput Genomic Sequences: phases 0, 1 and 2. Finished, phase 3 HTG sequences are in nr.
- **pat.** Nucleotides from the Patent division of GenBank.
- **pdb.** Sequences derived from the 3-dimensional structure records from Protein Data Bank. They are NOT the coding sequences for the corresponding proteins found in the same PDB record.
- **alu.** Select Alu repeats from REPBASE, suitable for masking Alu repeats from query sequences. See "Alu alert" by Claverie and Makalowski, Nature 371: 752 (1994).
- **dbsts.** Database of Sequence Tag Site entries from the STS division of GenBank + EMBL + DDBJ.
- **chromosome.** Complete genomes and complete chromosomes from the NCBI Reference Sequence project. It overlaps with refseq_genomic.
- **env_nt.** Sequences from environmental samples, such as uncultured bacterial samples isolated from soil or marine samples. The largest single source is Sagarssso Sea project. This does overlap with nucleotide nr.
- **tsa_nt.** Transcriptome Shotgun Assembly database.
- **prokaryotic_16S_ribosomal_RNA.** 16S ribosomal RNA sequences.
- **Betacoronavirus.** NCBI database of betacoronavirus sequences.

C.3 Adding more databases

Besides the databases that are part of the default configuration, you can add more databases located at NCBI by configuring files in the Workbench installation directory.

The list of databases that can be added is here: https://web.archive.org/web/20120409025527/http://www.ncbi.nlm.nih.gov/staff/tao/URLAPI/remote_blastdblist.html.

In order to add a new database, find the `settings` folder in the Workbench installation directory (e.g. `C:\Program files\CLC Genomics Workbench 4`). Download unzip and place the following files in this directory to replace the built-in list of databases:

- Nucleotide databases: https://resources.qiagenbioinformatics.com/wbsettings/NCBI_BlastNucleotideDatabases.zip
- Protein databases: https://resources.qiagenbioinformatics.com/wbsettings/NCBI_BlastProteinDatabases.zip

Open the file you have downloaded into the settings folder, e.g. NCBI_BlastProteinDatabases.proper in a text editor and you will see the contents look like this:

```
nr[clcddefault] = Non-redundant protein sequences
refseq_protein = Reference proteins
swissprot = Swiss-Prot protein sequences
pat = Patented protein sequences
pdb = Protein Data Bank proteins
env_nr = Environmental samples
month = New or revised GenBank sequences
```

Simply add another database as a new line with the first item being the database name taken from https://web.archive.org/web/20120409025527/http://www.ncbi.nlm.nih.gov/staff/tao/URLAPI/remote_blastdblist.html and the second part is the name to display in the Workbench. Restart the Workbench, and the new database will be visible in the BLAST dialog.

Appendix D

Proteolytic cleavage enzymes

Most proteolytic enzymes cleave at distinct patterns. Below is a compiled list of proteolytic enzymes used in *CLC Genomics Workbench*.

Name	P4	P3	P2	P1	P1'	P2'
Cyanogen bromide (CNBr)	-	-	-	M	-	-
Asp-N endopeptidase	-	-	-	-	D	-
Arg-C	-	-	-	R	-	-
Lys-C	-	-	-	K	-	-
Trypsin	-	-	-	K, R	not P	-
Trypsin	-	-	W	K	P	-
Trypsin	-	-	M	R	P	-
Trypsin*	-	-	C, D	K	D	-
Trypsin*	-	-	C	K	H, Y	-
Trypsin*	-	-	C	R	K	-
Trypsin*	-	-	R	R	H,R	-
Chymotrypsin-high spec.	-	-	-	F, Y	not P	-
Chymotrypsin-high spec.	-	-	-	W	not M, P	-
Chymotrypsin-low spec.	-	-	-	F, L, Y	not P	-
Chymotrypsin-low spec.	-	-	-	W	not M, P	-
Chymotrypsin-low spec.	-	-	-	M	not P, Y	-
Chymotrypsin-low spec.	-	-	-	H	not D, M, P, W	-
o-Iodosobenzoate	-	-	-	W	-	-
Thermolysin	-	-	-	not D, E	A, F, I, L, M or V	-
Post-Pro	-	-	H, K, R	P	not P	-
Glu-C	-	-	-	E	-	-
Asp-N	-	-	-	-	D	-
Proteinase K	-	-	-	A, E, F, I, L, T, V, W, Y	-	-
Factor Xa	A, F, G, I, L, T, V, M	D,E	G	R	-	-
Granzyme B	I	E	P	D	-	-
Thrombin	-	-	G	R	G	-
Thrombin	A, F, G, I, L, T, V, M	A, F, G, I, L, T, V, W, A	P	R	not D, E	not D, E
TEV (Tobacco Etch Virus)	-	Y	-	Q	G, S	-

Appendix E

Restriction enzymes database configuration

CLC Genomics Workbench uses enzymes from the **REBASE** restriction enzyme database at <http://rebase.neb.com>. If you wish to add enzymes to this list, you can do this by manually using the procedure described here.

Note! Please be aware that this process needs to be handled carefully, otherwise you may have to re-install the Workbench to get it to work.

First, download the following file: https://resources.qiagenbioinformatics.com/wbsettings/link_emboss_e_custom. In the Workbench installation folder under `settings`, create a folder named `rebase` and place the extracted `link_emboss_e_custom` file here.

Note that in macOS, the extension file "link_emboss_e_custom" will have a ".txt" extension in its filename and metadata that needs to be removed. Right click the file name, choose "Get info" and remove ".txt" from the "Name & extension" field.

Open the file in a text editor. The top of the file contains information about the format, and at the bottom there are two example enzymes that you should replace with your own.

Please note that the CLC Workbenches only support the addition of 2-cutter enzymes. Further details about how to format your entries accordingly are given within the file mentioned above.

After adding the above file, or making changes to it, you must restart the Workbench for changes take effect.

Appendix F

Technical information about modifying Gateway cloning sites

The *CLC Genomics Workbench* comes with a pre-defined list of Gateway recombination sites. These sites and the recombination logics can be modified by downloading and editing a properties file. Note that this is a technical procedure only needed if the built-in functionality is not sufficient for your needs.

The properties file can be downloaded from <https://resources.qiagenbioinformatics.com/wbsettings/gatewaycloning.zip>. Extract the file included in the zip archive and save it in the `settings` folder of the Workbench installation folder. The file you download contains the standard configuration. You should thus update the file to match your specific needs. See the comments in the file for more information.

The name of the properties file you download is `gatewaycloning.1.properties`. You can add several files with different configurations by giving them a different number, e.g. `gatewaycloning.2.properties` and so forth. When using the Gateway tools in the Workbench, you will be asked which configuration you want to use (see figure F.1).

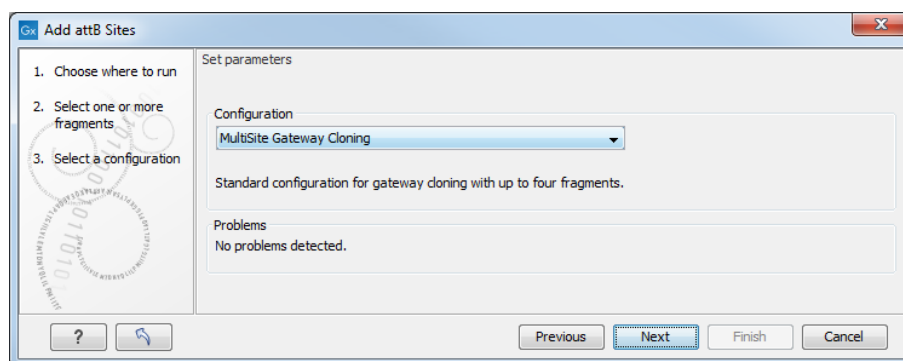


Figure F.1: Selecting between different gateway cloning configurations.

Appendix G

IUPAC codes for amino acids

(Single-letter codes based on International Union of Pure and Applied Chemistry)

The information is gathered from: https://www.insdc.org/documents/feature_table.html

One-letter abbreviation	Three-letter abbreviation	Description
A	Ala	Alanine
R	Arg	Arginine
N	Asn	Asparagine
D	Asp	Aspartic acid
C	Cys	Cysteine
Q	Gln	Glutamine
E	Glu	Glutamic acid
G	Gly	Glycine
H	His	Histidine
J	Xle	Leucine or Isoleucineucine
L	Leu	Leucine
I	ILe	Isoleucine
K	Lys	Lysine
M	Met	Methionine
F	Phe	Phenylalanine
P	Pro	Proline
O	Pyl	Pyrrolysine
U	Sec	Selenocysteine
S	Ser	Serine
T	Thr	Threonine
W	Trp	Tryptophan
Y	Tyr	Tyrosine
V	Val	Valine
B	Asx	Aspartic acid or Asparagine Asparagine
Z	Glx	Glutamic acid or Glutamine Glutamine
X	Xaa	Any amino acid

Appendix H

IUPAC codes for nucleotides

(Single-letter codes based on International Union of Pure and Applied Chemistry)

The information is gathered from: <https://www.iupac.org> and https://www.insdc.org/documents/feature_table.html.

Code	Description
A	Adenine
C	Cytosine
G	Guanine
T	Thymine
U	Uracil
R	Purine (A or G)
Y	Pyrimidine (C, T, or U)
M	C or A
K	T, U, or G
W	T, U, or A
S	C or G
B	C, T, U, or G (not A)
D	A, T, U, or G (not C)
H	A, T, U, or C (not G)
V	A, C, or G (not T, not U)
N	Any base (A, C, G, T, or U)

Appendix I

Formats for import and export

I.1 List of bioinformatic data formats

Below is a list of bioinformatic data formats, i.e. formats for importing and exporting molecule structures, sequences, alignments, trees, etc.

I.1.1 Sequence data formats

Sequence data formats that can be imported using **Standard Import** are listed in the table below.

Note: There are dedicated import tools for sequence data types not listed in the table below, such as high-throughput sequencing data from Illumina, IonTorrent and other vendors, as well as high-throughput fasta and trace files. See section [7.3](#).

File type	Suffix	Import	Export	Description
AB1	.ab1	X		Including chromatograms
ABI	.abi	X		Including chromatograms
CLC	.clc	X	X	Rich format including all information
Clone manager	.cm5	X		Clone manager sequence format
DNAstrider	.str/.strider	X	X	
DS Gene	.bsml	X		
EMBL	.emb/.embl	X	X	Rich information incl. annotations (nucs only)
FASTA	.fa/.fsa/.fasta	X	X	Simple format, name & description
FASTQ	.fastq	X	X	Simple format, name & description
GenBank	.gbk/.gb/.gp/.gbt	X	X	Rich information incl. annotations
Gene Construction Kit	.gcc	X		
Lasergene	.pro/.seq	X		
Nexus	.nxs/.nexus	X	X	
Phred	.phd	X		Including chromatograms
PIR (NBRF)	.pir	X	X	Simple format, name & description
Raw sequence	any	X		Only sequence (no name)
SCF2	.scf	X		Including chromatograms
SCF3	.scf	X	X	Including chromatograms
Sequence Comma separated values	.csv	X	X	Simple format. One seq per line: name, description(optional), sequence
Staden	.sdn	X		
Swiss-Prot	.swp	X	X	Rich information incl. annotations (only peptides)
Tab delimited text	.txt		X	Annotations in tab delimited text format

Additional notes about working with trace data

- When importing trace data, the called bases in the file are imported and the chromatogram information associated with the called bases is imported. If the base calls within the file have already been trimmed, the part of the chromatogram not associated with base calls will not be imported.
- The **Trim Sequences** tool, described in section 22.2, adds annotations to trimmed regions. When exporting to fasta format, there is an option to remove sequence ends covered by Trim annotations.

I.1.2 Read mapping formats

File type	Suffix	Import	Export	Description
ACE	.ace	X	X	No chromatogram or quality score
AGP	.agp/.fa		X	Exports scaffolded contigs (see below)
BAM	.bam	X	X	Binary Alignment/Map. A compressed representation of SAM format.
CLC	.clc	X	X	Rich format including all information
CLC Assembly File	.cas	X		Output from the CLC Assembly Cell
CRAM	.cram	X	X	Compressed Reference-oriented Alignment/Map. A compressed, reference-relative, space-saving representation of SAM format.
SAM	.sam	X	X	Sequence Alignment/Map. A tab delimited text format. See notes below.
Mapping coverage	.tsv		X	Detailed per-base info on coverage (see below)

Import and export of SAM/BAM /CRAM format files

Import of SAM, BAM and CRAM format files is described in section 7.4.3.

The format specification when exporting to SAM, BAM or CRAM format is described in section J. Index files can be created as part of BAM and CRAM exports.

AGP export

Sequence lists and read mappings generated by de novo assembly can be exported using the AGP exporter. On export, contigs are split up based on annotations of type **Scaffold**. These annotations are added when the "Perform scaffolding" option is enabled when assembling paired reads. Contig sequences are exported to a single FASTA format file, with the accompanying AGP format file containing information about how the contigs relate to one another.

AGP export is described further in section 35.1.1.

Export of coverage information from read mappings

Coverage information from read mappings can be exported in a tabular format using **Mapping Coverage** export. The output contains information on the number of nucleotides aligned to positions of reference sequences. Insertions are also reported, as described below while deletions are reported as reference regions without read coverage. Both stand-alone read mappings and reads tracks can be used as input.

The exported file contains the following columns:

Column	Description
1	Reference name
2	Reference position
3	Reference sub-position (insertion)
4	Reference symbol
5	Number of As
6	Number of Cs
7	Number of Gs
8	Number of Ts
9	Number of Ns
10	Number of Gaps
11	Total number of reads covering the position

The **Reference sub-position** column is empty (indicated by a - symbol) when the reference is defined at a given position. In case of an insertion this column contains an index into the insertion (a number between 1 and the length of the insertion) while the **Reference symbol** column is empty and the **Reference position** column contains the position of the last reference.

The "Export all columns" option is selected by default. When it is deselected, options for selecting the columns to export are presented in the next wizard step.

When selecting specific columns for export, the option "Export the table as currently shown" is particularly useful if you have filtered, sorted, or selected particular columns in a table that is open in a View. In this case, the effects of these manipulations are preserved in the exported file. This option is not available for all data types.

When the "Export the table as currently shown" is unchecked or disabled, checkboxes for each column to be exported are available to select or deselect. The buttons below that section can help speed up the process of column selection:

- **All** Select all possible columns.
- **None** Clear the existing selection.
- **Default** Select a standard set of columns, as defined by the software for this data type.
- **Last export** Select the columns that were selected during the most recent previous export.
- **Active View** Select the same set of columns as those selected in the Side Panel of the open data element. This button is only visible if the element being exported is in an open View.

In the final wizard step, select the location where the exported elements should be saved.

I.1.3 Alignment formats

File type	Suffix	Import	Export	Description
Aligned fasta	.fa/.fsa/.fasta	X	X	Simple fasta-based format with – for gaps
CLC	.clc	X	X	Rich format including all information
ClustalW	.aln	X	X	
GCG Alignment	.msf	X	X	
Nexus	.nxs/.nexus	X	X	
Phylip Alignment	.phy	X	X	

I.1.4 Tree formats

File type	Suffix	Import	Export	Description
CLC	.clc	X	X	Rich format including all information
Newick	.nwk	X	X	
Nexus	.nxs/.nexus	X	X	

I.1.5 Expression data formats

File type	Suffix	Import	Export	Description
Affymetrix CHP	.chp/.psi	X		Expression values and annotations, see section K.2 .
Affymetrix pivot/metric	.txt/.csv	X		Gene-level expression value, see section K.2 .
Affymetrix NetAffx	.csv	X		Annotations, see section K.2 .
BED	.bed		X	Expression values, see section 8.1.9
CLC	.clc	X	X	Rich format including all information
Excel	.xls/.xlsx		X	All tables and reports
Generic	.txt/.csv	X		Expression values
Generic	.txt/.csv	X		Annotations
GEO sample/series	.txt/.csv	X		Expression values, see section K.1
Illumina	.txt	X		Expression values and annotations, see section K.3
Table CSV	.csv		X	Samples and experiments, see section K.5
Tab delimited	.txt	X	X	Samples and experiments, see section K.5

I.1.6 Annotation and variant formats

Please note that all of the annotation and variant formats can be imported as tracks (see section 7.2). GFF, GVF and GTF formats can also be imported as annotations on a standard (i.e., non-track) sequence or sequence list using Annotate with GFF/GTF/GVF file, see section 18.1.

File type	Suffix	Import	Export	Description
Annotation CSV export	.csv		X	Annotations in csv format
Annotation Excel 2010	.xlsx		X	Annotations in Excel format
Annotation Excel 97 - 2007	.xls		X	Annotations in Excel format
BED	.bed	X	X	See section 7.2 and section 8.1.9
COSMIC database	.tsv	X		Special format for COSMIC data
CLC	.clc	X	X	Rich format including all information
GFF	.gff	X		To import as annotation track, see section 7.2.
GFF3	.gff3	X	X	See section 7.2.1 and section 8.1.8
GVF	.gvf	X	X	Special version of GFF for variant data, see section 7.2.
GTF	.gtf	X	X	Special version of GFF for gene annotation data, see section 7.2.
UCSC database dump	.txt	X		See section 7.2
VCF	.vcf	X	X	See section 7.2.2 and section 8.1.7
Wiggle	.wig	X	X	See section 7.2

Special notes on chromosome names synonyms used during import

When importing annotations as tracks, we try to make things simple for the user by having a set of chromosome names that are recognized as synonyms. The check on the chromosome name comparison is made by looking through the chromosomes in the order in which they are registered in the genome. The first match with any of the synonym names for a given chromosome is the chromosome to which the information will be added.

The synonyms applied are:

For any number N between (including) 1 and 22:

N, chrN, chromosome_N, and NC_00000N are seen as meaning the same thing. As concrete examples:

1 == chr1 == chromosome_1 == NC_000001

22 == chr22 == chromosome_22 == NC_000022

For any number N larger than 23:

N, chrN, chromosome_N are seen as meaning the same thing. As a concrete example:

26 == chr26 == chromosome_26

For chromosome names with letters, not numbers:

X, chrX, and chromosome_X and NC_000023 are synonyms.

Y, chrY, chromosome_Y and NC_000024 are synonyms.

M, MT, chrM, chrMT, chromosome_M, chromosome_MT and NC_001807 are synonyms.

The accession numbers in the listings above (NC_XXXXXX) allow for the matching against NCBI hg19 human reference names against the names used by USCS and vitally, the names used by Ensembl. Thus, in this case, if you have the correct number of chromosomes in a human reference (i.e. 25 references, including the hg19 mitochondria), that set of tracks can be used as the basis for downloading/importing annotations via Download Genomes, for example.

Note: These rules only apply for importing annotations as tracks, whether that is directly or via Download Genomes. Synonyms are not applied when doing BAM imports or when using **Annotate with GFF/GTF/GVF file**. There, your reference names in the *CLC Genomics Workbench* must exactly match the references names used in your BAM file or GFF/GTF/GVF file respectively.

I.1.7 miRBase data file format

The miRBase database is available for download and installation via the *CLC Genomics Workbench*, as described in section [33.3.1](#).

MiRBase .dat files can also be imported using Standard Import functionality, and selecting the *miRBase dat* in the **Force import as type** menu of the Standard Import dialog.

A *.dat file has the following format:

```
ID cel-let-7
XX
DE Caenorhabditis elegans let-7 stem-loop
XX
FH Key Location/Qualifiers
FH
FT miRNA 17..38
FT /product="cel-let-7-5p"
FT miRNA 60..81
FT /product="cel-let-7-3p"
XX
SQ Sequence 99 BP; 26 A; 19 C; 24 G; 0 T; 30 other;
uacacugugg auccggugag guaguagguu guauaguuug gaauauuacc accggugaac 60
uauagcauuu ucuaccuac cggagacaga acucuucga 99
//
ID cel-lin-4
XX
DE Caenorhabditis elegans lin-4 stem-loop
XX
FH Key Location/Qualifiers
FH
FT miRNA 16..36
```

```

FT /product="cel-lin-4-5p"
FT miRNA 55..76
FT /product="cel-lin-4-3p"
XX
SQ Sequence 94 BP; 17 A; 25 C; 26 G; 0 T; 26 other;
augcuuccgg ccuguucccu gagaccucaa gugugagugu acuauugaug cuucacaccu 60
gggcucuccg gguaccagga cgguuugagc agau 94
//

```

If the above formatting is followed, the dat file can be imported as a miRBase file for annotation purposes. In particular, the following needs to be in place:

- The sequences needs "miRNA" annotation on the precursor sequences. In the *CLC Genomics Workbench*, you can add a miRNA annotation by selecting a region and right clicking on **Add Annotation**. You should have a maximum of 2 miRNA annotations per precursor sequence. Matches to first miRNA annotation are counting in 5' column. Matches to second miRNA annotation are counted as 3' matches.
- If you have sequence list containing sequences from multiple species, the **Latin name** of the sequences should be set. This is used in the annotation dialog where you can select the species. If the Latin name is not set, the dialog will show "N/A".

I.1.8 Other formats

File type	Suffix	Import	Export	Description
CLC	.clc	X	X	Rich format including all information
PDB	.pdb	X		3D structure
RNA structures	.ct, .col, .rnaml/.xml	x		Secondary structure for RNA

I.1.9 Table and text formats

File type	Suffix	Import	Export	Description
Excel	.xls/.xlsx	X	X	All tables and reports
Table CSV	.csv	X	X	All tables
Tab delimited	.tsv	X		All tables
Tab delimited	.txt		X	All tables
Text	.txt	X	X	All data in a textual format
CLC	.clc	X	X	Rich format including all information
HTML	.html		X	All tables
PDF	.pdf		X	Export reports in Portable Document Format

Please see section [I.1.5](#) for special cases of table imports.

I.1.10 File compression formats

File type	Suffix	Import	Export	Description
Zip export	.zip		X	Selected files in CLC format
Zip import	.zip/.gz/.tar	X		Contained files/folder structure (.tar and .zip not supported for NGS data)

Note: It is possible to import 'external' files into the *CLC Genomics Workbench* and view these in the **Navigation Area**, but it is only the above mentioned formats whose *contents* can be shown in the *CLC Genomics Workbench*.

I.2 List of graphics data formats

Below is a list of formats for exporting graphics. All data displayed in a graphical format can be exported using these formats. Data represented in lists and tables can only be exported in .pdf format (see section 8.2 for further details).

Format	Suffix	Type
Portable Network Graphics	.png	bitmap
JPEG	.jpg	bitmap
Tagged Image File	.tif	bitmap
PostScript	.ps	vector graphics
Encapsulated PostScript	.eps	vector graphics
Portable Document Format	.pdf	vector graphics
Scalable Vector Graphics	.svg	vector graphics

Appendix J

SAM/BAM/CRAM export format specification

Specifications

The *CLC Genomics Workbench* aims to import and export SAM and BAM files according to the v1.4-r962 version of the SAM specification (see <https://samtools.github.io/hts-specs/SAMv1.pdf>), and CRAM files according to the v3.1 version of the CRAM specification (see <https://samtools.github.io/hts-specs/CRAMv3.pdf>). This appendix describes how the *CLC Genomics Workbench* exports SAM, BAM and CRAM files, along with known limitations.

General notes about the exporters

The exporters write unsorted SAM/BAM/CRAM files.

Reference names are updated to match the SAM specification:

- All leading and trailing whitespaces are removed.
- Occurrences of disallowed characters according to the specification (whitespaces \ , " ' ' @ () [] <>) are replaced by _ (underscore). Additionally, = and * are only disallowed at the beginning of the reference names. E.g., a reference name `*my=reference@sequence` is exported to `_my=reference_sequence`.

The following read group tags are supported: ID, SM, PI and PL. All other read group tags are ignored.

The exporters can also output additional annotations added by tools provided by plugins. Where that is the case, further details are provided in the plugin manual.

Alignment Section

Here are a few remarks on the alignment sections of the exported files:

- Unmapped reads are not exported.

- If pairs are not on the same contig, the mates will be exported as single reads.
- Multi segment mappings will be imported as a paired data set.
- If a read name contains spaces, the spaces are replaced by an underscore '_'.
- The exported CIGAR string uses 'M' to indicate match or mismatch and does not use '=' (equals sign) or 'X'.
- The *CLC Genomics Workbench* does not support or record mapping quality for read mappings. To fulfill the requirement in the format specifications that a read mapping quality is recorded for all mapped reads, the values 0 and 60 are used when mappings are exported. The value 60 is given to reads that mapped uniquely. The value 0 is given to reads that could map equally well to other locations besides the one being reported in the file.

Optional fields in the alignment section

The following is true for the export of optional fields:

- The NH tag is exported.
- The NM tag is not exported.
- For bisulfite mapped reads, an XR tag is exported with value "CT" or "GA". It describes the read conversion.
- For bisulfite mapped reads, an XG tag is exported with value "CT" or "GA". It describes the reference conversion.

J.1 Flags

The use of alignment flags by the *CLC Genomics Workbench* is shown in the following table and subsequent examples.

Bit	SAM description	Usage
0x1	Template having multiple segments in sequencing	Set if the segment is part of a pair
0x2	Each segment properly aligned according to the aligner	Set if the pair is not broken
0x4	Segment unmapped	Never set since the exporter does not export unmapped reads
0x8	Next segment in the template unmapped	Never set by the exporter. If a segment has an unmapped mate, the flag 0x1 is not set for the segment, i.e. it is not output as part of a pair
0x10	SEQ being reverse complemented	Set if and only if the segment was reverse complemented during mapping
0x20	SEQ of the next segment in the template being reversed	Set if and only if the mate was reverse complemented during mapping
0x40	The first segment in the template	This mate is the first segment of the pair
0x80	The last segment in the template	This mate is the second segment of the pair
0x100	Secondary alignment	Never set by the exporter. No reads with this flag set are imported.
0x200	Not passing quality controls	Never set by the exporter and ignored by the importer
0x400	PCR or optical duplicate	Never set by the exporter and ignored by the importer

Flag Examples

The following table illustrates some of the possible flags in the *CLC Genomics Workbench*.

Description of the example	Bits	Flag	Illustration
The first mate of a non-broken paired read	0x1, 0x2, 0x20, 0x40	99	Figure J.1
The second mate of a non-broken paired read	0x1, 0x2, 0x10, 0x80	147	Figure J.2
A single, forward read (or paired read, where only one mate of the pair is mapped)	No set bits	0	Figure J.3
A single, reversed read (or paired read, where only one mate of the pair is mapped)	0x10	16	Figure J.4
The first, forward segment from a broken pair with forward mate	0x1, 0x40	65	Figure J.5
The second, forward segment from broken pair with reversed mate	0x1, 0x20, 0x80	161	Figure J.6
The first, reversed segment from broken pair with forward mate	0x1, 0x10, 0x40	81	Figure J.7
The second, reversed segment from broken pair with reversed mate	0x1, 0x10, 0x20, 0x80	177	Figure J.8

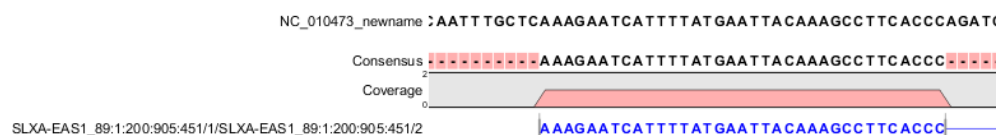


Figure J.1: The read is paired, both reads are mapped and the mate of this read is reversed

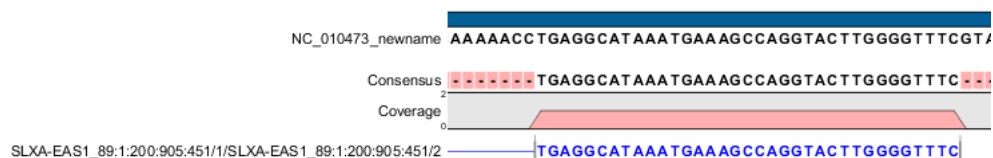


Figure J.2: The read is paired, both mates are mapped, and this segment is reversed

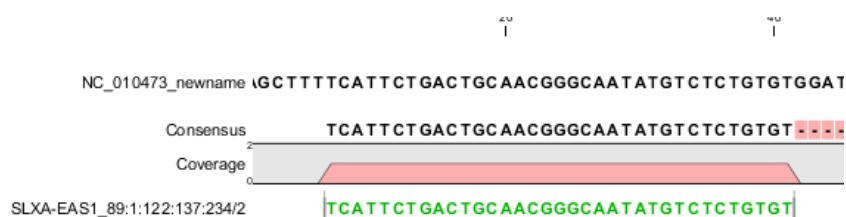


Figure J.3: A single, forward read, or a paired read where the mate is not mapped

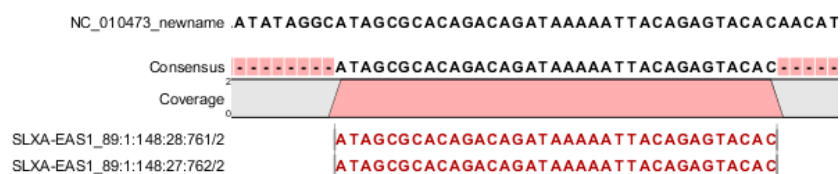


Figure J.4: The read is a single, reversed read, or a paired read where the mate is not mapped

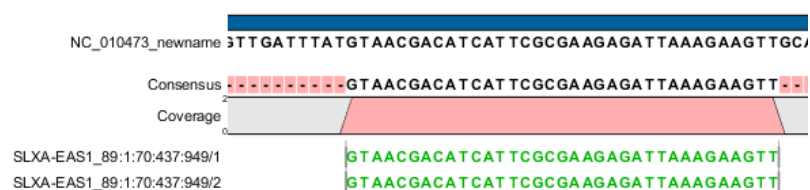


Figure J.5: These forward reads are paired. They map to the same place, so the pair is broken

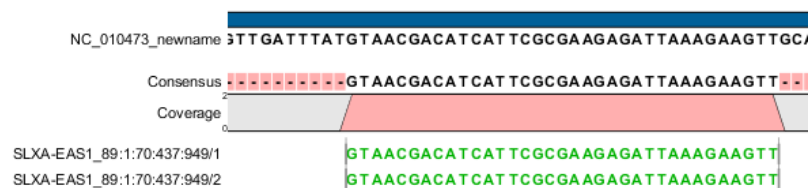


Figure J.6: Forward read that is part of a broken read where the mate is reversed

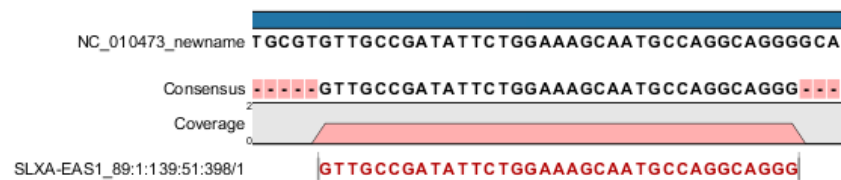


Figure J.7: Reversed read that is part of a broken pair, where the mate is forward

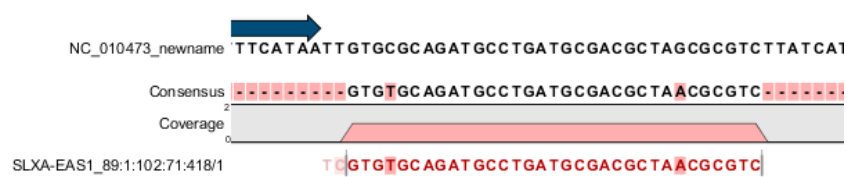


Figure J.8: Reversed read that is part of a broken pair, where the mate is also reversed.

Appendix K

Gene expression annotation files and microarray data formats

The *CLC Genomics Workbench* supports analysis of one-color expression arrays. These may be imported from GEO soft sample- or series- file formats, or for Affymetrix arrays, tab-delimited pivot or metrics files, or from Illumina expression files. Expression array data from other platforms may be imported from tab, semi-colon or comma separated files containing the expression feature IDs and levels in a tabular format (see [internalrefsec:customexpressiondataformatssectionGeneric](#) expression and annotation data file formats).

The *CLC Genomics Workbench* assumes that expression values are given at the gene level, thus probe-level analysis of Affymetrix GeneChips and import of Affymetrix CEL and CDF files is currently not supported. However, the *CLC Genomics Workbench* allows import of txt files exported from R containing processed Affymetrix CEL-file data (see [internalrefsec:AffymetrixGeneChipFormatssectionAffymetrix](#) GeneChip).

Affymetrix NetAffx annotation files for expression GeneChips in csv format and Illumina annotation files can also be imported.

Also, you may import your own annotation data in tabular format (see [internalrefsec:customexpressiondataforma](#) expression and annotation data file formats).

Below you find descriptions of the microarray data formats that are supported by *CLC Genomics Workbench*. Note that we for some platforms support both expression data and annotation data.

K.1 GEO (Gene Expression Omnibus)

The GEO (Gene Expression Omnibus) sample and series formats are supported. Figure [K.1](#) shows how to download the data from GEO in the right format. GEO is located at <https://www.ncbi.nlm.nih.gov/geo/>.

The GEO sample files are tab-delimited .txt files. They have three required lines:

```
^SAMPLE = GSM21610
!sample_table_begin
...
!sample_table_end
```

NCBI GEO > **Accession Display** [?] Not logged in | [Login](#) [?]

Scope: Format: Amount: GEO accession:

Series GSE6943 [Query DataSets for GSE6943](#)

Status	Public on Jan 24, 2008
Title	Normal Heart vs Normal Diaphragm
Organism(s)	Rattus norvegicus
Experiment type	Expression profiling by array
Summary	Comparison of gene expression of heart (left vent) and diaphragm of normal Sprague Dawley rats, young adult

Figure K.1: Selecting Samples, SOFT and Data before clicking go will give you the format supported by the **CLC Genomics Workbench**.

The first line should start with `^SAMPLE` = followed by the sample name, the line `!sample_table_begin` and the line `!sample_table_end`. Between the `!sample_table_begin` and `!sample_table_end`, lines are the column contents of the sample.

Note that GEO sample importer will also work for concatenated GEO sample files – allowing multiple samples to be imported in one go. Download a sample file containing concatenated sample files here:

<https://resources.qiagenbioinformatics.com/madata/GEOSampleFilesConcatenated.txt>

Below you can find examples of the formatting of the GEO formats.

GEO sample file, simple

This format is very simple and includes two columns: one for feature id (e.g. gene name) and one for the expression value.

```
^SAMPLE = GSM21610
!sample_table_begin
ID_REF    VALUE
id1       105.8
id2       32
id3       50.4
id4       57.8
id5       2914.1
!sample_table_end
```

Download the sample file here:

<https://resources.qiagenbioinformatics.com/madata/GEOSampleFileSimple.txt>

GEO sample file, including present/absent calls

This format includes an extra column for absent/present calls that can also be imported.

```
^SAMPLE = GSM21610
```

```
!sample_table_begin
ID_REF    VALUE    ABS_CALL
id1       105.8    M
id2       32      A
id3       50.4    A
id4       57.8    A
id5       2914.1  P
!sample_table_end
```

Download the sample file here:

<https://resources.qiagenbioinformatics.com/madata/GEOSampleFileAbsentPresent.txt>

GEO sample file, including present/absent calls and p-values

This format includes two extra columns: one for absent/present calls and one for absent/present call p-values, that can also be imported.

```
^SAMPLE = GSM21610
!sample_table_begin
ID_REF    VALUE    ABS_CALL    DETECTION P-VALUE
id1       105.8    M           0.00227496
id2       32      A           0.354441
id3       50.4    A           0.904352
id4       57.8    A           0.937071
id5       2914.1  P           6.02111e-05
!sample_table_end
```

Download the sample file here:

<https://resources.qiagenbioinformatics.com/madata/GEOSampleFileAbsentPresentCall.txt>

GEO sample file: using absent/present call and p-value columns for sequence information

The *CLC Genomics Workbench* assumes that if there is a third column in the GEO sample file then it contains present/absent calls and that if there is a fourth column then it contains p-values for these calls. This means that the contents of the third column is assumed to be text and that of the fourth column a number. As long as these two basic requirements are met, the sample should be recognized and interpreted correctly.

You can thus use these two columns to carry additional information on your probes. The absent/present column can be used to carry additional information like e.g. sequence tags as shown below:

```
^SAMPLE = GSM21610
!sample_table_begin
ID_REF    VALUE    ABS_CALL
id1       105.8    AAA
id2       32      AAC
```

```
id3          50.4      ATA
id4          57.8      ATT
id5          2914.1    TTA
!sample_table_end
```

Download the sample file here:

<https://resources.qiagenbioinformatics.com/madata/GEOSampleFileSimpleSequenceTag.txt>

Or, if you have multiple probes per sequence you could use the present/absent column to hold the sequence name and the p-value column to hold the interrogation position of your probes:

```
^SAMPLE = GSM21610
!sample_table_begin
ID_REF      VALUE      ABS_CALL      DETECTION P-VALUE
probe1      755.07      seq1          1452
probe2      587.88      seq1          497
probe3      716.29      seq1          1447
probe4      1287.18      seq2          1899
!sample_table_end
```

Download the sample file here: <https://resources.qiagenbioinformatics.com/madata/GEOSampleFileSimpleSequenceTagAndProbe.txt>

GEO series file, simple

The series file includes expression values for multiple samples. Each of the samples in the file will be represented by its own element with the sample name. The first row lists the sample names.

```
!Series_title "Myb specificity determinants"
!series_matrix_table_begin
"ID_REF" "GSM21610" "GSM21611" "GSM21612"
"id1"    2541      1781.8      1804.8
"id2"    11.3      621.5       50.2
"id3"    61.2      149.1       22
"id4"    55.3      328.8       97.2
"id5"    183.8     378.3       423.2
!series_matrix_table_end
```

Download the sample file here: <https://resources.qiagenbioinformatics.com/madata/GEOSeriesFile.txt>

K.2 Affymetrix GeneChip

For Affymetrix, three types of files are currently supported: Affymetrix .CHP files, Affymetrix NetAffx annotation files and tab-delimited pivot or metrics files. Affymetrix .CEL files are currently not supported. However, the Bioconductor R package 'affy' allows you to preprocess the .CEL files

and export a txt file containing a table of estimated probe-level log-transformed expression values in three lines of code:

```
library(affy) # loading Bioconductor library 'affy'
data=ReadAffy() # probe-level data import
eset=rma(data) # probe-level data pre-processing using 'rma'
write.exprs(eset,file="evals.txt") # writing log2 expression levels to 'evals.txt'
```

The exported txt file (evals.txt) can be imported into the *CLC Genomics Workbench* using the Generic expression data table format importer (see [internalrefsec:customexpressionformatssectionGeneric](#) expression and annotation data file formats; you can just 'drag-and-drop' it in). In R, you should have all the CEL files you wish to process in your working directory and the file 'evals.txt' will be written to that directory.

If multiple probes are present for the same gene, further processing may be required to merge them into a single gene-level expression.

Affymetrix CHP expression files

The Affymetrix scanner software produces a number of files when a GeneChip is scanned. Two of these are the .CHP and the .CEL files. These are binary files with native Affymetrix formats. The Affymetrix GeneChips contain a number of probes for each gene (typically between 22 and 40). The .CEL file contains the probe-level intensities, and the .CHP file contains the gene-level information. The gene-level information has been obtained by the scanner software through postprocessing and summarization of the probe-level intensities.

In order to interpret the probe-level information in the .CEL file, the .CDF file for the type of GeneChip that was used is required. Similarly for the .CHP file: in order to interpret the gene-level information in the .CHP file, the .PSI file for the type of GeneChip that was used is required.

In order to import a .CHP file it is required that the corresponding .PSI file is present in the same folder as the .CHP file you want to import, and furthermore, this must be the only .PSI file that is present there. There are no requirements for the name of the .PSI file. Note that the .PSI file itself will not be imported - it is only used to guide the import of the .CHP file which contains the expression values.

Download example .CHP and .PSI files here (note that these are binary files):

<https://resources.qiagenbioinformatics.com/madata/AffymetrixCHPandPSI.zip>

Affymetrix metrics files

The Affymetrix metrics or pivot files are tab-delimited files that may be exported from the Affymetrix scanner software. The metrics files have a lot of technical information that is only partly used in the *CLC Genomics Workbench*. The feature ids (Probe Set Name), expression values (Used Signal), absent/present call (Detection) and absent/present p-value (Detection p-value) are imported into the *CLC Genomics Workbench*.

Download a small example sample file here:

<https://resources.qiagenbioinformatics.com/madata/AffymetrixMetrics.txt>

Affymetrix NetAffx annotation files

The NetAffx annotation files for Whole-Transcript Expression Gene arrays and 3' IVT Expression Analysis Arrays can be imported and used to annotate experiments as shown in section 34.1.3.

Download a small example annotation file here which includes header information:

<https://resources.qiagenbioinformatics.com/madata/AffymetrixNetAffxAnnotationFile.csv>

K.3 Illumina BeadChip

Both BeadChip expression data files from Illumina's BeadStudio software and the corresponding BeadChip annotation files are supported by *CLC Genomics Workbench*. The formats of the BeadStudio and annotation files have changed somewhat over time and various formats are supported.

Illumina expression data, compact format

An example of this format is shown below:

TargetID	AVG_Signal	BEAD_STDEV	Detection
GI_10047089-S	112.5	4.2	0.16903226
GI_10047091-S	127.6	4.8	0.76774194

All this information is imported into the *CLC Genomics Workbench*. The AVG_Signal is used as the expression measure.

Download a small sample file here:

<https://resources.qiagenbioinformatics.com/madata/IlluminaBeadChipCompact.txt>

Illumina expression data, extended format

An example of this format is shown below:

TargetID	MIN_Signal	AVG_Signal	MAX_Signal	NARRAYS	ARRAY_STDEV	BEAD_STDEV	Avg_NBEADS	Detection
GI_10047089-S	73.7	73.7	73.7	1	NaN	3.4	53	0.05669084
GI_10047091-S	312.7	312.7	312.7	1	NaN	11.1	50	0.99604483

All this information is imported into the *CLC Genomics Workbench*. The AVG_Signal is used as the expression measure.

Download a small sample file here:

<https://resources.qiagenbioinformatics.com/madata/IlluminaBeadChipExtended.txt>

Illumina expression data, with annotations

An example of this format is shown below:

TargetID	Accession	Symbol	Definition	Synonym	Signal-BG02	DCp32	Detection-BG02	DCp32
GI_10047089-S	NM_014332.1	SMPX	"Homo sapiens small muscle protein, X-linked (SMPX), mRNA."		-17.6	0.03559657		
GI_10047091-S	NM_013259.1	NP25	"Homo sapiens neuronal protein (NP25), mRNA."		NP22	32.6	0.99604483	
GI_10047093-S	NM_016299.1	HSP70-4	"Homo sapiens likely ortholog of mouse heat shock protein, 70 kDa 4 (HSP70-4), mRNA."		228.1	1		

Only the TargetID, Signal and Detection columns will be imported, the remaining columns will be ignored. This means that the annotations are not imported. The Signal is used as the expression measure.

Download a small example sample file here:

<https://resources.qiagenbioinformatics.com/madata/IlluminaBeadStudioWithAnnotation.txt>

Illumina expression data, multiple samples in one file

This file format has too much information to show it inline in the text. You can download a small example sample file here:

<https://resources.qiagenbioinformatics.com/madata/IlluminaBeadStudioMultipleSamples.txt>

This file contains data for 18 samples. Each sample has an expression value (the value in the AVG_Signal column), a detection p-value, a bead standard deviation and an average bead number column. The *CLC Genomics Workbench* recognizes the 18 samples and their columns.

Illumina annotation files

The *CLC Genomics Workbench* supports import of two types of Illumina BeadChip annotation files. These are either comma-separated or tab-delimited .txt files. They can be used to annotate experiments as shown in section 34.1.3.

This file format has too much information to show it inline in the text.

Download a small example annotation file of the first type here:

<https://resources.qiagenbioinformatics.com/madata/IlluminaBeadChipAnnotation.txt>

K.4 Gene ontology annotation files

The Gene ontology web site provides annotation files for a variety of species which can all be downloaded and imported into the *CLC Genomics Workbench*. They can be used to annotate experiments as shown in section 34.1.3. They can also be used with the Gene Set Test and Create Expression Browser tools.

Import GO annotation file using the Standard Import tool. For GO annotation files in GAF format, use the option "Force import as type: Gene Ontology Annotation file" from the drop down menu at the bottom of the Standard Import dialog.

See the list of available files at <https://current.geneontology.org/products/pages/downloads.html>.

K.5 Generic expression and annotation data file formats

If you have your expression or annotation data in Excel and can export the data as a txt file, or if you are able to do some scripting or other manipulations to format your data files, you will be

able to import them into the *CLC Genomics Workbench* as a 'generic' expression or annotation data file. There are a few simple requirements that need to be fulfilled to do this as described below.

Generic expression data table format

The *CLC Genomics Workbench* will import a tab, semicolon or comma-separated .txt or .csv file as expression array samples if the following requirements are met:

1. the first non-empty line of the file contains text. All entries, except the first, will be used as sample names
2. the following (non-empty) lines contain the same number of entries as the first non-empty line. The requirements to these are that the first entry should be a string (this will be used as the feature ID) and the remaining entries should contain numbers (which will be used as expression values – one per sample). Empty entries are not allowed, but NaN values are allowed.
3. the file contains at least two samples.

An example of this format is shown below:

```
FeatureID;sample1;sample2;sample3
gene1;200;300;23
gene2;210;30;238
gene3;230;50;23
gene4;50;100;235
gene5;200;300;23
gene6;210;30;238
gene7;230;50;23
gene8;50;100;235
```

This will be imported as three samples with eight genes in each sample.

Download this example as a file here:

<https://resources.qiagenbioinformatics.com/madata/CustomExpressionData.txt>

Generic annotation file for expression data format

The *CLC Genomics Workbench* will import a tab, semicolon or comma-separated .txt or .csv file as an annotation file if the following is met:

1. It has a line which can serve as a valid header line. In order to do this, the line should have a number of headers where at least two are among the valid column headers in the **Column header** column below.
2. It contains one of the PROBE_ID headers (that is: 'Probe Set ID', 'Feature ID', 'ProbeID' or 'Probe_Id').

The importer will import an annotation table with a column for each of the valid column headers (those in the **Column header** column below). Columns with invalid headers will be ignored.

Note that some column headers are alternatives so that only one of the alternative columns headers should be used.

When adding annotations to an experiment, you can specify the column in your annotation file containing the relevant identifiers. These identifiers are matched to the feature ids already present in your experiment. When a match is found, the annotation is added to that entry in the experiment. In other words, at least one column in your annotation file must contain identifiers matching the feature identifiers in the experiment, for those annotations to be applied.

A simple example of an annotation file is shown here:

```
"Probe Set ID","Gene Symbol","Gene Ontology Biological Process"
"1367452_at","Sumo2","0006464 // protein modification process // not recorded"
"1367453_at","Cdc37","0051726 // regulation of cell cycle // not recorded"
"1367454_at","Copb2","0006810 // transport // /// 0016044 // membrane organization // "
```

Download this example plus a more elaborate one here:

<https://resources.qiagenbioinformatics.com/madata/SimpleCustomAnnotation.csv>

<https://resources.qiagenbioinformatics.com/madata/FullCustomAnnotation.csv>

To meet requirements imposed by special functionalities in the *CLC Genomics Workbench*, there are a number of further restrictions on the contents in the entries of the columns:

Download sequence functionality In the experiment table, you can click a button to download sequence. This uses the contents of the `PUBLIC_ID` column, so this column must be present for the action to work and should contain the NCBI accession number.

Annotation tests The annotation tests can make use of several entries in a column as long as a certain format is used. The tests assume that entries are separated by `///` and it interprets all that appears before `//` as the actual entry and all that appears after `//` within an entry as comments. Example:

```
/// 0000001 // comment1 /// 0000008 // comment2 /// 0003746 // comment3
```

The annotation tests will interpret this as three entries (0000001, 0000008, and 0003746) with the according comments.

The most common column headers are summarized below:

Column header in imported file (alternatives separated by commas)	Label in experiment table	Description (tool tip)
Probe Set ID, Feature ID, ProbelD, Probe_Id, transcript_cluster_id	Feature ID	Probe identifier tag
Representative Public ID, Public identifier tag, GenbankAccession	Public identifier tag	Representative public ID
Gene Symbol, GeneSymbol	Gene symbol	Gene symbol
Gene Ontology Biological Process, Ontology_Process, GO_biological_process	GO biological process	Gene Ontology biological process
Gene Ontology Cellular Component, Ontology_Component, GO_cellular_component	GO cellular component	Gene Ontology cellular component
Gene Ontology Molecular Function, Ontology_Function, GO_molecular_function	GO molecular function	Gene Ontology molecular function
Pathway	Pathway	Pathway

The full list of possible column headers:

Column header in imported file (alternatives separated by commas)	Label in experiment table	Description (tool tip)
Species Scientific Name, Species Name, Species	Species name	Scientific species name
GeneChip Array	Gene chip array	Gene Chip Array name
Annotation Date	Annotation date	Date of annotation
Sequence Type	Sequence type	Type of sequence
Sequence Source	Sequence source	Source from which sequence was obtained
Transcript ID(Array Design), Transcript	Transcript ID	Transcript identifier tag
Target Description	Target description	Target description
Archival UniGene Cluster	Archival UniGene cluster	Archival UniGene cluster
UniGene ID, UniGeneID, Unigene_ID, unigene	UniGene ID	UniGene identifier tag
Genome Version	Genome version	Version of genome on which annotation is based
Alignments	Alignments	Alignments
Gene Title	Gene title	Gene title
geng_assignments	Gene assignments	Gene assignments
Chromosomal Location	Chromosomal location	Chromosomal location
Unigene Cluster Type	UniGene cluster type	UniGene cluster type
Ensembl Ensembl	Ensembl	
Entrez Gene, EntrezGeneID, Entrez_Gene_ID	Entrez gene	Entrez gene
SwissProt	SwissProt	SwissProt
EC	EC	EC
OMIM	OMIM	Online Mendelian Inheritance in Man
RefSeq Protein ID	RefSeq protein ID	RefSeq protein identifier tag
RefSeq Transcript ID	RefSeq transcript ID	RefSeq transcript identifier tag
FlyBase	FlyBase	FlyBase
AGI	AGI	AGI
WormBase	WormBase	WormBase
MGI Name	MGI name	MGI name
RGD Name	RGD name	RGD name
SGD accession number	SGD accession number	SGD accession number
InterPro	InterPro	InterPro
Trans Membrane	Trans membrane	Trans membrane
QTL	QTL	QTL
Annotation Description	Annotation description	Annotation description
Annotation Transcript Cluster	Annotation transcript cluster	Annotation transcript cluster
Transcript Assignments	Transcript assignments	Transcript assignments
mma_assignments	mRNA assignments	mRNA assignments
Annotation Notes	Annotation notes	Annotation notes
GO, Ontology	Go annotations	Go annotations
Cytoband	Cytoband	Cytoband
PrimaryAccession	Primary accession	Primary accession
RefSeqAccession	RefSeq accession	RefSeq accession
GeneName	Gene name	Gene name
TIGRID	TIGR id	TIGR id
Description	Description	Description
GenomicCoordinates	Genomic coordinates	Genomic coordinates
Search_key	Search key	Search key
Target	Target	Target
Gid, GI	Genbank identifier	Genbank identifier
Accession	GenBank accession	GenBank accession
Symbol	Gene symbol	Gene symbol
Probe_Type	Probe type	Probe type
crosshyb_type	Crosshyb type	Crosshyb type
category	category	category
Start, Probe_Start	Start	Start
Stop	Stop	Stop
Definition	Definition	Definition
Synonym, Synonyms	Synonym	Synonym
Source	Source	Source
Source_Reference_ID	Source reference id	Source reference id
RefSeq_ID	Reference sequence id	Reference sequence id
ILMN_Gene	Illumina Gene	Illumina Gene
Protein_Product	Protein product	Protein product
protein_domains	Protein domains	Protein domains
Array_Address_Id	Array adress id	Array adress id
Probe_Sequence	Sequence	Sequence
seqname	Seqname	Seqname
Chromosome	Chromosome	Chromosome
strand	Strand	Strand
Probe_Chr_Orientation	Probe chr orientation	Probe chr orientation
Probe_Coordinates	Probe coordinates	Probe coordinates
Obsolete_Probe_Id	Obsolete probe id	Obsolete probe id

Appendix L

Custom codon frequency tables

You can edit the list of codon frequency tables used by *CLC Genomics Workbench*.

Note! Please be aware that this process needs to be handled carefully, otherwise you may have to re-install the Workbench to get it to work.

In the Workbench installation folder under `res`, there is a folder named `codonfreq`. This folder contains all the codon frequency tables organized into subfolders in a hierarchy. In order to change the tables, you simply add, delete or rename folders and the files in the folders. If you wish to add new tables, please use the existing ones as template. In existing tables, the "_number" at the end of the ".cftbl" file name is the number of CDSs that were used for calculation, according to the <https://www.kazusa.or.jp/codon/> site.

When creating a custom table, it is not necessary to fill in all fields as only the codon information (e.g. 'GCG' in the example below) and the counts (e.g. 47869.00) are used when doing reverse translation:

Name: Rattus norvegicus GeneticCode: 1 Ala GCG 47869.00 6.86 0.10 Ala GCA 109203.00 15.64 0.23

In particular, the amino acid type is not used: in order to use an alternative genetic code, it must be specified in the 'GeneticCode' line instead.

Restart the Workbench to have the changes take effect.

Appendix M

Comparison of track comparison tools

This section of the manual provides an overview about comparison, filtering and annotation tools that work with tracks.

Tool name	Description of the tool	Example of possible applications	Comments
Identify Shared Variants (see section 32.3.1)	Identifies common variants in a group of samples	Identification of common variants in inherited deafness	Can also be used to get all variants in a group of samples by setting the frequency threshold to 0%
TRIO analysis (see section 32.3.3)	Identifies de novo and accumulated variants in a child by comparing with the variants present in the mother and father	Identification of causative variants in rare mendelian diseases	-
Identify Enriched Variants in Case vs Control Samples (see section 32.3.2)	Identifies enriched variants in a group of samples with a certain phenotype (the case group) in comparison to a group of samples not showing this phenotype (the control group). Control and case samples are from different individuals	Identification of causative common variants in non-rare diseases	To retrieve significant results, a large number of samples is required for each group

Tool name	Description of the tool	Example applications	Comments
Remove Variants Present in Control Reads (see section 32.1.4)	Removes germline variants from a set of called variants in a case sample (e.g. a cancer sample) by using mapped sequencing reads from a normal sample from the same individual	Comparison of cancer versus normal from the same individual	-
Filter against known variants (see section 32.1.1)	Removes variants that are present (or absent) in an external variant database (available as track) from a set of called variants	Removal of common (assumed germline) variants from a set of called variants in a case sample to identify somatic variants	-
Annotate against known variants (see section 32.2.1)	Adds information from one or several external database(s), that are available as track(s), to called variants in a sample to see how many of them are known in this(these) database(s)	Adds information from COSMIC to see how many of the variants found in the sample are known to be associated with cancer	The tool has special rules for when information from the database are transferred and when it is not. This means that only information is transferred in cases with exact matches (when the same variant is found in the sample AND the database) and in cases where variants in the database are completely contained in the set of variants that have been called in the sample
Filter based on overlap (see section 27.6.2)	Removes elements from a set of genetic annotations such as genes, regulatory elements, or variants. Only genetic annotations are kept that overlap or do not overlap regions in the other track	Removing deletions or amplifications that do not overlap genes to identify those that are potential causative. Filtering away variants outside targeted regions in an DNA amplification sequencing experiment	-

Bibliography

- [Allison et al., 2006] Allison, D., Cui, X., Page, G., and Sabripour, M. (2006). Microarray data analysis: from disarray to consolidation and consensus. *NATURE REVIEWS GENETICS*, 7(1):55.
- [Altschul et al., 1990] Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *J Mol Biol*, 215(3):403–410.
- [Andrade et al., 1998] Andrade, M. A., O'Donoghue, S. I., and Rost, B. (1998). Adaptation of protein surfaces to subcellular location. *J Mol Biol*, 276(2):517–525.
- [Ashburner et al., 2000] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2000). Gene ontology: tool for the unification of biology. *Nat Genet*, 25(1):25–29.
- [Auer and Doerge, 2010] Auer, P. L. and Doerge, R. (2010). Statistical design and analysis of rna sequencing data. *Genetics*, 185(2):405–416.
- [Bachmair et al., 1986] Bachmair, A., Finley, D., and Varshavsky, A. (1986). In vivo half-life of a protein is a function of its amino-terminal residue. *Science*, 234(4773):179–186.
- [Baggerly et al., 2003] Baggerly, K., Deng, L., Morris, J., and Aldaz, C. (2003). Differential expression in SAGE: accounting for normal between-library variation. *Bioinformatics*, 19(12):1477–1483.
- [Bateman et al., 2004] Bateman, A., Coin, L., Durbin, R., Finn, R. D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E. L., Studholme, D. J., Yeats, C., and Eddy, S. R. (2004). The Pfam protein families database. *Nucleic Acids Res.*, 32(Database issue):D138–D141.
- [Benjamini and Hochberg, 1995] Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *JOURNAL-ROYAL STATISTICAL SOCIETY SERIES B*, 57:289–289.
- [Berman et al., 2003] Berman, H., Henrick, K., and Nakamura, H. (2003). Announcing the worldwide protein data bank. *Nat Struct Biol*, 10(12):980.
- [Bishop and Friday, 1985] Bishop, M. J. and Friday, A. E. (1985). Evolutionary trees from nucleic acid and protein sequences. *Proceeding of the Royal Society of London*, B 226:271–302.
- [Blaisdell, 1989] Blaisdell, B. E. (1989). Average values of a dissimilarity measure not requiring sequence alignment are twice the averages of conventional mismatch counts requiring sequence alignment for a computer-generated model system. *J Mol Evol*, 29(6):538–47.

- [Bolstad et al., 2003] Bolstad, B., Irizarry, R., Astrand, M., and Speed, T. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193.
- [Bommarito et al., 2000] Bommarito, S., Peyret, N., and SantaLucia, J. (2000). Thermodynamic parameters for DNA sequences with dangling ends. *Nucleic Acids Res*, 28(9):1929–1934.
- [Chen et al., 2004] Chen, G., Znosko, B. M., Jiao, X., and Turner, D. H. (2004). Factors affecting thermodynamic stabilities of RNA 3 x 3 internal loops. *Biochemistry*, 43(40):12865–12876.
- [Cheng et al., 2021] Cheng, H., Concepcion, G., Feng, X., Zhang, H., and H., L. (2021). Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm. *Nat Methods*, 18(2):170–175.
- [Clote et al., 2005] Clote, P., Ferré, F., Kranakis, E., and Krizanc, D. (2005). Structural RNA has lower folding energy than random RNA of the same dinucleotide frequency. *RNA*, 11(5):578–591.
- [Cornette et al., 1987] Cornette, J. L., Cease, K. B., Margalit, H., Spouge, J. L., Berzofsky, J. A., and DeLisi, C. (1987). Hydrophobicity scales and computational techniques for detecting amphipathic structures in proteins. *J Mol Biol*, 195(3):659–685.
- [Costa, 2007] Costa, F. F. (2007). Non-coding RNAs: lost in translation? *Gene*, 386(1-2):1–10.
- [Crooks et al., 2004] Crooks, G. E., Hon, G., Chandonia, J.-M., and Brenner, S. E. (2004). WebLogo: a sequence logo generator. *Genome Res*, 14(6):1188–1190.
- [Dayhoff and Schwartz, 1978] Dayhoff, M. O. and Schwartz, R. M. (1978). *Atlas of Protein Sequence and Structure*, volume 3 of 5 suppl., pages 353–358. Nat. Biomed. Res. Found., Washington D.C.
- [Dayhoff et al., 1978] Dayhoff, M. O., Schwartz, R. M., and Orcutt, B. C. (1978). A model of evolutionary change in protein. *Atlas of Protein Sequence and Structure*, 5(3):345–352.
- [Dempster et al., 1977] Dempster, A., Laird, N., Rubin, D., et al. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38.
- [Dudoit et al., 2003] Dudoit, S., Shaffer, J., and Boldrick, J. (2003). Multiple Hypothesis Testing in Microarray Experiments. *STATISTICAL SCIENCE*, 18(1):71–103.
- [Eddy, 2004] Eddy, S. R. (2004). Where did the BLOSUM62 alignment score matrix come from? *Nat Biotechnol*, 22(8):1035–1036.
- [Edgar, 2004] Edgar, R. C. (2004). Muscle: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, 5:113.
- [Efron, 1982] Efron, B. (1982). *The jackknife, the bootstrap and other resampling plans*, volume 38. SIAM.
- [Eisen et al., 1998] Eisen, M., Spellman, P., Brown, P., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868.

- [Eisenberg et al., 1984] Eisenberg, D., Schwarz, E., Komaromy, M., and Wall, R. (1984). Analysis of membrane and surface protein sequences with the hydrophobic moment plot. *J Mol Biol*, 179(1):125–142.
- [Emini et al., 1985] Emini, E. A., Hughes, J. V., Perlow, D. S., and Boger, J. (1985). Induction of hepatitis a virus-neutralizing antibody by a virus-specific synthetic peptide. *J Virol*, 55(3):836–839.
- [Engelman et al., 1986] Engelman, D. M., Steitz, T. A., and Goldman, A. (1986). Identifying nonpolar transbilayer helices in amino acid sequences of membrane proteins. *Annu Rev Biophys Chem*, 15:321–353.
- [Falcon and Gentleman, 2007] Falcon, S. and Gentleman, R. (2007). Using GOstats to test gene lists for GO term association. *Bioinformatics*, 23(2):257.
- [Felsenstein, 1981] Felsenstein, J. (1981). Evolutionary trees from DNA sequences: a maximum likelihood approach. *J Mol Evol*, 17(6):368–376.
- [Felsenstein, 1985] Felsenstein, J. (1985). Confidence limits on phylogenies: An approach using the bootstrap. *Journal of Molecular Evolution*, 39:783–791.
- [Feng and Doolittle, 1987] Feng, D. F. and Doolittle, R. F. (1987). Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J Mol Evol*, 25(4):351–360.
- [Forsberg et al., 2001] Forsberg, R., Oleksiewicz, M. B., Petersen, A. M., Hein, J., Bøtner, A., and Storgaard, T. (2001). A molecular clock dates the common ancestor of European-type porcine reproductive and respiratory syndrome virus at more than 10 years before the emergence of disease. *Virology*, 289(2):174–179.
- [Galperin and Koonin, 1998] Galperin, M. Y. and Koonin, E. V. (1998). Sources of systematic error in functional annotation of genomes: domain rearrangement, non-orthologous gene displacement and operon disruption. *In Silico Biol*, 1(1):55–67.
- [Gentleman and Mullin, 1989] Gentleman, J. F. and Mullin, R. (1989). The distribution of the frequency of occurrence of nucleotide subsequences, based on their overlap capability. *Biometrics*, 45(1):35–52.
- [Gill and von Hippel, 1989] Gill, S. C. and von Hippel, P. H. (1989). Calculation of protein extinction coefficients from amino acid sequence data. *Anal Biochem*, 182(2):319–326.
- [Gnerre et al., 2011] Gnerre, S., Maccallum, I., Przybylski, D., Ribeiro, F. J., Burton, J. N., Walker, B. J., Sharpe, T., Hall, G., Shea, T. P., Sykes, S., Berlin, A. M., Aird, D., Costello, M., Daza, R., Williams, L., Nicol, R., Gnirke, A., Nusbaum, C., Lander, E. S., and Jaffe, D. B. (2011). High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proceedings of the National Academy of Sciences of the United States of America*, 108(4):1513–8.
- [Gonda et al., 1989] Gonda, D. K., Bachmair, A., Wüning, I., Tobias, J. W., Lane, W. S., and Varshavsky, A. (1989). Universality and structure of the N-end rule. *J Biol Chem*, 264(28):16700–16712.
- [Guindon and Gascuel, 2003] Guindon, S. and Gascuel, O. (2003). A Simple, Fast, and Accurate Algorithm to Estimate Large Phylogenies by Maximum Likelihood. *Systematic Biology*, 52(5):696–704.

- [Guo et al., 2006] Guo, L., Lobenhofer, E. K., Wang, C., Shippy, R., Harris, S. C., Zhang, L., Mei, N., Chen, T., Herman, D., Goodsaid, F. M., Hurban, P., Phillips, K. L., Xu, J., Deng, X., Sun, Y. A., Tong, W., Dragan, Y. P., and Shi, L. (2006). Rat toxicogenomic study reveals analytical consistency across microarray platforms. *Nat Biotechnol*, 24(9):1162–1169.
- [Han et al., 1999] Han, K., Kim, D., and Kim, H. (1999). A vector-based method for drawing RNA secondary structure. *Bioinformatics*, 15(4):286–297.
- [Hasegawa et al., 1985] Hasegawa, M., Kishino, H., and Yano, T. (1985). Dating of the human–ape splitting by a molecular clock of mitochondrial DNA. *Journal of Molecular Evolution*, 22(2):160–174.
- [Heinz et al., 2010] Heinz, S., Benner, C., Spann, N., Bertolino, E., Lin, Y. C., Laslo, P., Cheng, J. X., Murre, C., Singh, H., and Glass, C. K. (2010). Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. *Mol cell*, 38(4):576–589.
- [Henikoff and Henikoff, 1992] Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci U S A*, 89(22):10915–10919.
- [Heydarian et al., 2014] Heydarian, M., Romeo Luperchio, T., Cutler, J., Mitchell, C., Kim, M.-S., Pandey, A., Soliner-Webb, B., and Reddy, K. (2014). Prediction of gene activity in early B cell development based on an integrative multi-omics analysis. *J Proteomics Bioinform*, 7(2):050–063.
- [Höhl et al., 2007] Höhl, M., Rigoutsos, I., and Ragan, M. A. (2007). Pattern-based phylogenetic distance estimation and tree reconstruction. *Evolutionary Bioinformatics*, 2:0–0.
- [Homer N, 2010] Homer N, N. S. (2010). Improved variant discovery through local re-alignment of short-read next-generation sequencing data using srma. *Genome Biol.*, 11(10):R99.
- [Hopp and Woods, 1983] Hopp, T. P. and Woods, K. R. (1983). A computer program for predicting protein antigenic determinants. *Mol Immunol*, 20(4):483–489.
- [Ikai, 1980] Ikai, A. (1980). Thermostability and aliphatic index of globular proteins. *J Biochem (Tokyo)*, 88(6):1895–1898.
- [Janin, 1979] Janin, J. (1979). Surface and inside volumes in globular proteins. *Nature*, 277(5696):491–492.
- [Jones et al., 1992] Jones, D., Taylor, W., and Thornton, J. (1992). The rapid generation of mutation data matrices from protein sequences. *Computer Applications in the Biosciences (CABIOS)*, 8:275–282.
- [Jukes and Cantor, 1969] Jukes, T. and Cantor, C. (1969). *Mammalian Protein Metabolism*, chapter Evolution of protein molecules, pages 21–32. New York: Academic Press.
- [Kal et al., 1999] Kal, A. J., van Zonneveld, A. J., Benes, V., van den Berg, M., Koerkamp, M. G., Albermann, K., Strack, N., Ruijter, J. M., Richter, A., Dujon, B., Ansorge, W., and Tabak, H. F. (1999). Dynamics of gene expression revealed by comparison of serial analysis of gene expression transcript profiles from yeast grown on two different carbon sources. *Mol Biol Cell*, 10(6):1859–1872.

- [Karplus and Schulz, 1985] Karplus, P. A. and Schulz, G. E. (1985). Prediction of chain flexibility in proteins. *Naturwissenschaften*, 72:212–213.
- [Kaufman and Rousseeuw, 1990] Kaufman, L. and Rousseeuw, P. (1990). Finding groups in data. an introduction to cluster analysis. *Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics*, New York: Wiley, 1990.
- [Kelly et al., 2012] Kelly, T. K., Liu, Y., Lay, F. D., Liang, G., Berman, B. P., and Jones, P. A. (2012). Genome-wide mapping of nucleosome positioning and DNA methylation within individual DNA molecules. *Genome Res.*, 22(12):2497–2506.
- [Kierzek et al., 1999] Kierzek, R., Burkard, M. E., and Turner, D. H. (1999). Thermodynamics of single mismatches in RNA duplexes. *Biochemistry*, 38(43):14214–14223.
- [Kimura, 1980] Kimura, M. (1980). A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J Mol Evol*, 16(2):111–120.
- [Knudsen and Miyamoto, 2001] Knudsen, B. and Miyamoto, M. M. (2001). A likelihood ratio test for evolutionary rate shifts and functional divergence among proteins. *Proc Natl Acad Sci U S A*, 98(25):14512–14517.
- [Knudsen and Miyamoto, 2003] Knudsen, B. and Miyamoto, M. M. (2003). Sequence alignments and pair hidden markov models using evolutionary history. *Journal of Molecular Biology*, 333(2):453 – 460.
- [Kolaskar and Tongaonkar, 1990] Kolaskar, A. S. and Tongaonkar, P. C. (1990). A semi-empirical method for prediction of antigenic determinants on protein antigens. *FEBS Lett*, 276(1-2):172–174.
- [Kumar et al., 2013] Kumar, V., Muratani, M., Rayan, N. A., Kraus, P., Lufkin, T., Ng, H. H., and Prabhakar, S. (2013). Uniform, optimal signal processing of mapped deep-sequencing data. *Nat Biotechnol*, 31(7):615–22.
- [Kyte and Doolittle, 1982] Kyte, J. and Doolittle, R. F. (1982). A simple method for displaying the hydropathic character of a protein. *J Mol Biol*, 157(1):105–132.
- [Landt et al., 2012] Landt, S. G., Marinov, G. K., Kundaje, A., Kheradpour, P., Pauli, F., Batzoglou, S., Bernstein, B. E., Bickel, P., Brown, J. B., Cayting, P., Chen, Y., DeSalvo, G., Epstein, C., Fisher-Aylor, K. I., Euskirchen, G., Gerstein, M., Gertz, J., Hartemink, A. J., Hoffman, M. M., Iyer, V. R., Jung, Y. L., Karmakar, S., Kellis, M., Kharchenko, P. V., Li, Q., Liu, T., Liu, X. S., Ma, L., Milosavljevic, A., Myers, R. M., Park, P. J., Pazin, M. J., Perry, M. D., Raha, D., Reddy, T. E., Rozowsky, J., Shores, N., Sidow, A., Slaterry, M., Stamatoyannopoulos, J. A., Tolstorukov, M. Y., White, K. P., Xi, S., Farnham, P. J., Lieb, J. D., Wold, B. J., and Snyder, M. (2012). ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia. *Genome Res*, 22(9):1813–31.
- [Law et al., 2014] Law, V., Knox, C., Djoumbou, Y., Jewison, T., Guo, A., Liu, Y., Maciejewski, A., Arndt, D., Wilson, M., Neveu, V., Tang, A., Gabriel, G., Ly, C., Adamjee, S., Dame, Z., Han, B., Zhou, Y., and Wishart, D. (2014). Drugbank 4.0: shedding new light on drug metabolism. *Nucleic Acids Res.*, 42:D1091–7.

- [Leitner and Albert, 1999] Leitner, T. and Albert, J. (1999). The molecular clock of HIV-1 unveiled through analysis of a known transmission history. *Proc Natl Acad Sci U S A*, 96(19):10752–10757.
- [Li et al., 2007] Li, B., Carey, M., and Workman, J. L. (2007). The role of chromatin during transcription. *Cell*, 128(4):707–719.
- [Li, 2018] Li, H. (2018). Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100.
- [Li et al., 2012] Li, J., Lupat, R., Amarasinghe, K. C., Thompson, E. R., Doyle, M. A., Ryland, G. L., Tothill, R. W., Halgamuge, S. K., Campbell, I. G., and Gorringe, K. L. (2012). Contra: copy number analysis for targeted resequencing. *Bioinformatics*, 28(10):1307–1313.
- [Li et al., 2010] Li, R., Zhu, H., Ruan, J., Qian, W., Fang, X., Shi, Z., Li, Y., Li, S., Shan, G., Kristiansen, K., Li, S., Yang, H., Wang, J., and Wang, J. (2010). De novo assembly of human genomes with massively parallel short read sequencing. *Genome research*, 20(2):265–72.
- [Lloyd, 1982] Lloyd, S. (1982). Least squares quantization in PCM. *Information Theory, IEEE Transactions on*, 28(2):129–137.
- [Longfellow et al., 1990] Longfellow, C. E., Kierzek, R., and Turner, D. H. (1990). Thermodynamic and spectroscopic study of bulge loops in oligoribonucleotides. *Biochemistry*, 29(1):278–285.
- [Love et al., 2014] Love, M. I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome Biology*, 15:550–.
- [Lu et al., 2008] Lu, M., Dousis, A. D., and Ma, J. (2008). Opus-rotax: A fast and accurate method for side-chain modeling. *Protein Science*, 17(9):1576–1585.
- [Maizel and Lenk, 1981] Maizel, J. V. and Lenk, R. P. (1981). Enhanced graphic matrix analysis of nucleic acid and protein sequences. *Proc Natl Acad Sci U S A*, 78(12):7665–7669.
- [Marinov et al., 2014] Marinov, G. K., Kundaje, A., Park, P. J., and Wold, B. J. (2014). Large-scale quality analysis of published ChIP-seq data. *G3 (Bethesda)*, 4(2):209–23.
- [Mathews et al., 2004] Mathews, D. H., Disney, M. D., Childs, J. L., Schroeder, S. J., Zuker, M., and Turner, D. H. (2004). Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of rna secondary structure. *Proc Natl Acad Sci U S A*, 101(19):7287–7292.
- [Mathews et al., 1999] Mathews, D. H., Sabina, J., Zuker, M., and Turner, D. H. (1999). Expanded sequence dependence of thermodynamic parameters improves prediction of rna secondary structure. *J Mol Biol*, 288(5):911–940.
- [Mathews and Turner, 2002] Mathews, D. H. and Turner, D. H. (2002). Experimentally derived nearest-neighbor parameters for the stability of RNA three- and four-way multibranch loops. *Biochemistry*, 41(3):869–880.
- [Mathews and Turner, 2006] Mathews, D. H. and Turner, D. H. (2006). Prediction of RNA secondary structure by free energy minimization. *Curr Opin Struct Biol*, 16(3):270–278.
- [McCarthy et al., 2012] McCarthy, D. J., Chen, Y., and Smyth, G. K. (2012). Differential expression analysis of multifactor rna-seq experiments with respect to biological variation. *Nucleic Acids Research*, 40:4288–4297.

- [McCaskill, 1990] McCaskill, J. S. (1990). The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29(6-7):1105–1119.
- [McGinnis and Madden, 2004] McGinnis, S. and Madden, T. L. (2004). BLAST: at the core of a powerful and diverse set of sequence analysis tools. *Nucleic Acids Res*, 32(Web Server issue):W20–W25.
- [Meyer et al., 2007] Meyer, M., Stenzel, U., Myles, S., Pruefer, K., and Hofreiter, M. (2007). Targeted high-throughput sequencing of tagged nucleic acid samples. *Nucleic Acids Res*, 35(15):e97.
- [Miao et al., 2011] Miao, Z., Cao, Y., and Jiang, T. (2011). Rasp: rapid modeling of protein side chain conformations. *Bioinformatics*, 27(22):3117–3122.
- [Michener and Sokal, 1957] Michener, C. and Sokal, R. (1957). A quantitative approach to a problem in classification. *Evolution*, 11:130–162.
- [Mortazavi et al., 2008] Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L., and Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by rna-seq. *Nat Methods*, 5(7):621–628.
- [Mukherjee and Zhang, 2009] Mukherjee, S. and Zhang, Y. (2009). MM-align: A quick algorithm for aligning multiple-chain protein complex structures using iterative dynamic programming. *Nucleic Acids Res.*, 37.
- [Niu and Zhang, 2012] Niu, Y. S. and Zhang, H. (2012). The screening and ranking algorithm to detect dna copy number variations. *Ann Appl Stat*, 6(3):1306–1326.
- [Pace et al., 1995] Pace, C. N., Vajdos, F., Fee, L., Grimsley, G., and Gray, T. (1995). How to measure and predict the molar absorption coefficient of a protein. *Protein science*, 4(11):2411–2423.
- [Parkhomchuk et al., 2009] Parkhomchuk, D., Borodina, T., Amstislavskiy, V., Banaru, M., Hallen, L., Krobisch, S., Lehrach, H., and Soldatov, A. (2009). Transcriptome analysis by strand-specific sequencing of complementary dna. *Nucleic Acids Res*, 37(18):e123.
- [Purvis, 1995] Purvis, A. (1995). A composite estimate of primate phylogeny. *Philos Trans R Soc Lond B Biol Sci*, 348(1326):405–421.
- [Rivas and Eddy, 2000] Rivas, E. and Eddy, S. R. (2000). Secondary structure alone is generally not statistically significant for the detection of noncoding RNAs. *Bioinformatics*, 16(7):583–605.
- [Robinson et al., 2010] Robinson, M. D., McCarthy, D. J., and Smyth, G. K. (2010). edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140.
- [Robinson and Oshlack, 2010] Robinson, M. D. and Oshlack, A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol.*, 11(3):R25.
- [Rose et al., 1985] Rose, G. D., Geselowitz, A. R., Lesser, G. J., Lee, R. H., and Zehfus, M. H. (1985). Hydrophobicity of amino acid residues in globular proteins. *Science*, 229(4716):834–838.

- [Rost, 2001] Rost, B. (2001). Review: protein secondary structure prediction continues to rise. *J Struct Biol*, 134(2-3):204–218.
- [Rye et al., 2011] Rye, M. B., Saetrom, P., and Drablos, F. (2011). A manually curated ChIP-seq benchmark demonstrates room for improvement in current peak-finder programs. *Nucleic Acids Res*, 39(4):e25.
- [Saitou and Nei, 1987] Saitou, N. and Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol*, 4(4):406–425.
- [Sankoff et al., 1983] Sankoff, D., Kruskal, J., Mainville, S., and Cedergren, R. (1983). *Time Warps, String Edits, and Macromolecules: the Theory and Practice of Sequence Comparison*, chapter Fast algorithms to determine RNA secondary structures containing multiple loops, pages 93–120. Addison-Wesley, Reading, Ma.
- [SantaLucia, 1998] SantaLucia, J. (1998). A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proc Natl Acad Sci U S A*, 95(4):1460–1465.
- [Schechter and Berger, 1967] Schechter, I. and Berger, A. (1967). On the size of the active site in proteases. I. Papain. *Biochem Biophys Res Commun*, 27(2):157–162.
- [Schechter and Berger, 1968] Schechter, I. and Berger, A. (1968). On the active site of proteases. 3. Mapping the active site of papain; specific peptide inhibitors of papain. *Biochem Biophys Res Commun*, 32(5):898–902.
- [Schneider and Stephens, 1990] Schneider, T. D. and Stephens, R. M. (1990). Sequence logos: a new way to display consensus sequences. *Nucleic Acids Res*, 18(20):6097–6100.
- [Schroeder et al., 1999] Schroeder, S. J., Burkard, M. E., and Turner, D. H. (1999). The energetics of small internal loops in RNA. *Biopolymers*, 52(4):157–167.
- [Shapiro et al., 2007] Shapiro, B. A., Yingling, Y. G., Kasprzak, W., and Bindewald, E. (2007). Bridging the gap in RNA structure prediction. *Curr Opin Struct Biol*, 17(2):157–165.
- [Siepel and Haussler, 2004] Siepel, A. and Haussler, D. (2004). Combining phylogenetic and hidden Markov models in biosequence analysis. *J Comput Biol*, 11(2-3):413–428.
- [Smith and Waterman, 1981] Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *J Mol Biol*, 147(1):195–197.
- [Stangl et al., 2020] Stangl, C., de Blank, S., Renkens, I., Westera, L., Verbeek, T., Valle-Inclan, J. E., González, R. C., Henssen, A. G., van Roosmalen, M. J., Stam, R. W., et al. (2020). Partner independent fusion gene detection by multiplexed crispr-cas9 enrichment and long read nanopore sequencing. *Nature Communications*, 11(1):2861.
- [Stanton et al., 2013] Stanton, K. P., Parisi, F., Strino, F., Rabin, N., Asp, P., and Kluger, Y. (2013). Arpeggio: harmonic compression of ChIP-seq data reveals protein-chromatin interaction signatures. *Nucleic Acids Res*, 41(16):e161.
- [Sturges, 1926] Sturges, H. A. (1926). The choice of a class interval. *Journal of the American Statistical Association*, 21:65–66.
- [The Gene Ontology Consortium, 2019] The Gene Ontology Consortium (2019). Gene ontology resource: 20 years and still going strong. *Nucleic Acids Research*, 47(D1):D330–D338.

- [Tian et al., 2005] Tian, L., Greenberg, S., Kong, S., Altschuler, J., Kohane, I., and Park, P. (2005). Discovering statistically significant pathways in expression profiling studies. *Proceedings of the National Academy of Sciences*, 102(38):13544–13549.
- [Tobias et al., 1991] Tobias, J. W., Shrader, T. E., Rocap, G., and Varshavsky, A. (1991). The N-end rule in bacteria. *Science*, 254(5036):1374–1377.
- [Tusher et al., 2001] Tusher, V. G., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proc Natl Acad Sci U S A*, 98(9):5116–5121.
- [Vandesompele et al., 2002] Vandesompele, J., Preter, K. D., Pattyn, F., Poppe, B., Roy, N. V., Paepe, A. D., and Speleman, F. (2002). Accurate normalization of real-time quantitative rt-pcr data by geometric averaging of multiple internal control genes. *Genome Biol.*
- [Vaser and Šiki, 2021] Vaser, R. and Šiki, M. (2021). Time-and memory-efficient genome assembly with raven. *Nature Computational Science*, 1(5):332–336.
- [Vaser et al., 2017] Vaser, R., Sovi, I., Nagarajan, N., and Šiki, M. (2017). Fast and accurate de novo genome assembly from long uncorrected reads. *Genome Research*, 27(5):737–746.
- [von Ahsen et al., 2001] von Ahsen, N., Wittwer, C. T., and Schütz, E. (2001). Oligonucleotide melting temperatures under PCR conditions: nearest-neighbor corrections for Mg(2+), deoxynucleotide triphosphate, and dimethyl sulfoxide concentrations with comparison to alternative empirical formulas. *Clin Chem*, 47(11):1956–1961.
- [Welling et al., 1985] Welling, G. W., Weijer, W. J., van der Zee, R., and Welling-Wester, S. (1985). Prediction of sequential antigenic regions in proteins. *FEBS Lett*, 188(2):215–218.
- [Whelan and Goldman, 2001] Whelan, S. and Goldman, N. (2001). A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. *Molecular Biology and Evolution*, 18:691–699.
- [Wick and Holt, 2019] Wick, R. R. and Holt, K. E. (2019). Benchmarking of long-read assemblers for prokaryote whole genome sequencing. *F1000Research*, 8:2138.
- [Wishart et al., 2006] Wishart, D., Knox, C., Guo, A., Shrivastava, S., Hassanali, M., Stothard, P., Chang, Z., and Woolsey, J. (2006). Drugbank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Res.*, 34:D668–72.
- [Wootton and Federhen, 1993] Wootton, J. C. and Federhen, S. (1993). Statistics of local complexity in amino acid sequences and sequence databases. *Computers in Chemistry*, 17:149–163.
- [Workman and Krogh, 1999] Workman, C. and Krogh, A. (1999). No evidence that mRNAs have lower folding free energies than random sequences with the same dinucleotide distribution. *Nucleic Acids Res*, 27(24):4816–4822.
- [Xu and Zhang, 2010] Xu, J. and Zhang, Y. (2010). How significant is a protein structure similarity with TM-score = 0.5? *Bioinformatics*, 26(7):889–95.
- [Yang, 1994a] Yang, Z. (1994a). Estimating the pattern of nucleotide substitution. *Journal of Molecular Evolution*, 39(1):105–111.

- [Yang, 1994b] Yang, Z. (1994b). Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: Approximate methods. *Journal of Molecular Evolution*, 39(3):306–314.
- [Zerbino and Birney, 2008] Zerbino, D. R. and Birney, E. (2008). Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res*, 18(5):821–829.
- [Zerbino et al., 2009] Zerbino, D. R., McEwen, G. K., Margulies, E. H., and Birney, E. (2009). Pebble and rock band: heuristic resolution of repeats and scaffolding in the velvet short-read de novo assembler. *PloS one*, 4(12):e8407.
- [Zhang and Skolnick, 2004] Zhang, Y. and Skolnick, J. (2004). Scoring function for automated assessment of protein structure template quality. *Proteins*, 57(4):702–10.
- [Zhou et al., 2014] Zhou, X., Lindsay, H., and Robinson, M. D. (2014). Robustly detecting differential expression in RNA sequencing data using observation weights. *Nucleic acids research*, 42(11):e91–e91.
- [Zuker, 1989a] Zuker, M. (1989a). On finding all suboptimal foldings of an rna molecule. *Science*, 244(4900):48–52.
- [Zuker, 1989b] Zuker, M. (1989b). The use of dynamic programming algorithms in rna secondary structure prediction. *Mathematical Methods for DNA Sequences*, pages 159–184.
- [Zuker and Sankoff, 1984] Zuker, M. and Sankoff, D. (1984). Rna secondary structures and their prediction. *Bulletin of Mathematical Biology*, 46:591–621.
- [Zuker and Stiegler, 1981] Zuker, M. and Stiegler, P. (1981). Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res*, 9(1):133–148.