

CLC **Genomics** Workbench

USER MANUAL

Manual for
CLC Genomics Workbench 21.0.1
Windows, macOS and Linux

December 11, 2020

This software is for research purposes only.

QIAGEN Aarhus
Silkeborgvej 2
Prismet
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 19
1.4 Workbench Licenses 21
1.5 Plugins 36
1.6 Network configuration 39
1.7 CLC Server connection 39
1.8 Getting started and latest improvements 41

II Core Functionalities **43**

2 User interface **44**

2.1 View Area 46
2.2 Zoom and selection in View Area 54
2.3 Toolbox and Favorites tabs 56
2.4 Processes tab and Status bar 57
2.5 Workspace 59
2.6 List of shortcuts 60

3 Data management and search **63**

3.1 Navigation Area 63
3.2 Working with tables 72
3.3 Customized attributes on data locations 76

3.4 Local search 81

4 User preferences and settings 88

4.1 General preferences 89
4.2 View preferences 90
4.3 Data preferences 93
4.4 Advanced preferences 93
4.5 Export/import of preferences 95
4.6 View settings for the Side Panel 96

5 Printing 98

5.1 Selecting which part of the view to print 99
5.2 Page setup 100
5.3 Print preview 102

6 Import/export of data and graphics 103

6.1 Standard import 104
6.2 Import tracks 106
6.3 Import high-throughput sequencing data 114
6.4 Import RNA spike-in controls 129
6.5 Import Primers 130
6.6 Data export 131
6.7 Export graphics to files 152
6.8 Export graph data points to a file 156
6.9 CLC Server data import and export 157
6.10 Copy/paste view output 158

7 Data download 159

7.1 Search for Sequences at NCBI 159
7.2 Search for PDB Structures at NCBI 162
7.3 Search for Sequences in UniProt (Swiss-Prot/TrEMBL) 165
7.4 SRA search 168
7.5 Sequence web info 173

8 References management **175**

8.1 Download Genomes 178
8.2 QIAGEN Sets 179
8.3 Custom Sets 181
8.4 Imported Data 186

9 Running tools, handling results and batching **188**

9.1 Running tools 188
9.2 Handling results 192
9.3 Batch processing 193

10 Metadata **197**

10.1 Creating metadata tables 200
10.2 Associating data elements with metadata 205
10.3 Working with data and metadata 209
10.4 Moving, copying and exporting metadata 212
10.5 Editing Metadata tables 213

11 Workflows **215**

11.1 Creating a workflow 216
11.2 Editing existing workflows 221
11.3 Workflow elements 229
11.4 Launching workflows individually and in batches 237
11.5 Batching part of a workflow 246
11.6 Advanced workflow batching 250
11.7 Managing workflows 255

III Basic sequence analysis **264**

12 Viewing and editing sequences **265**

12.1 View sequence 265
12.2 Circular DNA 275
12.3 Working with annotations 278
12.4 Element information 287

12.5 View as text . . . 288
12.6 Sequence Lists . . . 288

13 BLAST search **292**
13.1 Running BLAST searches . . . 293
13.2 Output from BLAST searches . . . 300
13.3 Local BLAST databases . . . 306
13.4 Manage BLAST databases . . . 309
13.5 Bioinformatics explained: BLAST . . . 310

14 3D Molecule Viewer **317**
14.1 Importing molecule structure files . . . 318
14.2 Viewing molecular structures in 3D . . . 322
14.3 Customizing the visualization . . . 323
14.4 Tools for linking sequence and structure . . . 332
14.5 Align Protein Structure . . . 336
14.6 Generate Biomolecule . . . 340

15 General sequence analyses **342**
15.1 Extract sequences . . . 342
15.2 Shuffle sequence . . . 344
15.3 Dot plots . . . 345
15.4 Local complexity plot . . . 353
15.5 Sequence statistics . . . 354
15.6 Join Sequences . . . 358
15.7 Pattern discovery . . . 359
15.8 Motif Search . . . 361
15.9 Create motif list . . . 366

16 Nucleotide analyses **368**
16.1 Convert DNA to RNA . . . 368
16.2 Convert RNA to DNA . . . 369
16.3 Reverse complements of sequences . . . 369
16.4 Translation of DNA or RNA to protein . . . 370

16.5 Find open reading frames 372

17 Protein analyses **375**

17.1 Protein charge 375
17.2 Antigenicity 377
17.3 Hydrophobicity 378
17.4 Download Pfam Database 383
17.5 Pfam domain search 383
17.6 Find and Model Structure 385
17.7 Secondary structure prediction 393
17.8 Protein report 395
17.9 Reverse translation from protein into DNA 397
17.10 Proteolytic cleavage detection 400

18 Primers **405**

18.1 Primer design - an introduction 406
18.2 Setting parameters for primers and probes 408
18.3 Graphical display of primer information 410
18.4 Output from primer design 412
18.5 Standard PCR 413
18.6 Nested PCR 417
18.7 TaqMan 418
18.8 Sequencing primers 420
18.9 Alignment-based primer and probe design 421
18.10 Analyze primer properties 426
18.11 Find binding sites and create fragments 427
18.12 Order primers 431

19 Sequencing data analyses **433**

19.1 Importing and viewing trace data 434
19.2 Trim sequences 435
19.3 Assemble sequences 438
19.4 Assemble sequences to reference 440

19.5 Sort sequences by name . . . 443
19.6 Add sequences to an existing contig . . . 446
19.7 View and edit contigs and read mappings . . . 447
19.8 Reassemble contig . . . 456
19.9 Secondary peak calling . . . 457
19.10 Extract Consensus Sequence . . . 458
19.11 Combine Reports . . . 462

20 Cutting and cloning **465**
20.1 Restriction site analyses . . . 465
20.2 Restriction enzyme lists . . . 474
20.3 Molecular cloning . . . 475
20.4 Gateway cloning . . . 484
20.5 Gel electrophoresis . . . 491

21 Sequence alignment **494**
21.1 Create an alignment . . . 494
21.2 View alignments . . . 499
21.3 Edit alignments . . . 503
21.4 Join alignments . . . 507
21.5 Pairwise comparison . . . 509

22 Phylogenetic trees **514**
22.1 K-mer Based Tree Construction . . . 516
22.2 Create tree . . . 517
22.3 Model Testing . . . 518
22.4 Maximum Likelihood Phylogeny . . . 520
22.5 Tree Settings . . . 528
22.6 Metadata and phylogenetic trees . . . 537

23 RNA structure **543**
23.1 RNA secondary structure prediction . . . 544
23.2 View and edit secondary structures . . . 550
23.3 Evaluate structure hypothesis . . . 558

23.4 Structure scanning plot . . . 560
23.5 Bioinformatics explained: RNA structure prediction by minimum free energy minimization . . . 562

IV High-throughput sequencing 568

24 Tracks 569

24.1 Track types . . . 570
24.2 Working with tracks . . . 572
24.3 Track lists . . . 580
24.4 Retrieving reference data tracks . . . 583
24.5 Merge Annotation Tracks . . . 583
24.6 Merge Variant Tracks . . . 584
24.7 Track Conversion . . . 585
24.8 Annotate and Filter . . . 588
24.9 Graphs . . . 591

25 Prepare sequencing data 598

25.1 QC for Sequencing Reads . . . 598
25.2 Trim Reads . . . 602
25.3 Demultiplex Reads . . . 617

26 Quality control for resequencing analysis 627

26.1 QC for Targeted Sequencing . . . 627
26.2 QC for Read Mapping . . . 637
26.3 Whole Genome Coverage Analysis . . . 643
26.4 Combine Reports . . . 645
26.5 Create Sample Report . . . 649

27 Read mapping 652

27.1 Map Reads to Reference . . . 653
27.2 Reads tracks and stand-alone read mappings . . . 664
27.3 Local Realignment . . . 680
27.4 Merge Read Mappings . . . 688

27.5 Remove Duplicate Mapped Reads . . . 688
27.6 Extract Consensus Sequence . . . 692

28 Variant detection **697**
28.1 Variant Detection tools . . . 698
28.2 Fixed Ploidy Variant Detection . . . 703
28.3 Low Frequency Variant Detection . . . 705
28.4 Basic Variant Detection . . . 705
28.5 Variant Detection - filters . . . 706
28.6 Variant Detection - the outputs . . . 714
28.7 Fixed Ploidy and Low Frequency Detection tools: detailed descriptions . . . 723
28.8 Copy Number Variant Detection . . . 731
28.9 Identify Known Mutations from Sample Mappings . . . 746
28.10 InDels and Structural Variants . . . 751

29 Resequencing **767**
29.1 Variant filtering . . . 768
29.2 Variant annotation . . . 772
29.3 Variants comparison . . . 777
29.4 Variant quality control . . . 782
29.5 Functional consequences . . . 783

30 RNA-seq and Small RNA analysis **800**
30.1 RNA-seq normalization . . . 802
30.2 RNA-Seq Analysis . . . 802
30.3 PCA for RNA-Seq . . . 825
30.4 Differential Expression . . . 829
30.5 Create Heat Map for RNA-Seq . . . 841
30.6 Create Expression Browser . . . 845
30.7 Create Venn Diagram for RNA-Seq . . . 849
30.8 Gene Set Test . . . 851
30.9 miRNA analysis . . . 855

31 Microarray analysis **866**

31.1 Experimental design . . . 867
31.2 Transformation and normalization . . . 876
31.3 Quality control . . . 880
31.4 Feature clustering . . . 889
31.5 Statistical analysis - identifying differential expression . . . 895
31.6 Annotation tests . . . 903
31.7 General plots . . . 907

32 De Novo sequencing **912**
32.1 The CLC de novo assembly algorithm . . . 912
32.2 De Novo Assembly . . . 921
32.3 Map Reads to Contigs . . . 928

33 Epigenomics analysis **931**
33.1 Histone Chip-Seq . . . 931
33.2 ChIP-Seq Analysis . . . 933
33.3 Annotate with nearby gene information . . . 938
33.4 Bisulfite Sequencing . . . 939
33.5 Advanced Peak Shape Tools . . . 957

34 Utility tools **963**
34.1 Batch Rename . . . 963
34.2 Extract Annotations . . . 969
34.3 Sample reads . . . 970
34.4 Extract Reads . . . 972
34.5 Merge Overlapping Pairs . . . 974

35 Legacy tools **978**
35.1 Compare Sample Variant Tracks . . . 978
35.2 Remove Reference Variants . . . 980
35.3 Import Roche 454 . . . 980
35.4 Create Combined RNA-Seq Report . . . 982
35.5 Create Track from Experiment . . . 984
35.6 Small RNA Analysis . . . 989

35.7 Batch launching workflows with multiple inputs 1007

V Appendix 1012

A Use of multi-core computers 1013

B Graph preferences 1015

C BLAST databases 1017

C.1 Peptide sequence databases 1017

C.2 Nucleotide sequence databases 1017

C.3 Adding more databases 1018

D Proteolytic cleavage enzymes 1020

E Restriction enzymes database configuration 1022

F Technical information about modifying Gateway cloning sites 1023

G IUPAC codes for amino acids 1024

H IUPAC codes for nucleotides 1025

I Formats for import and export 1026

I.1 List of bioinformatic data formats 1026

I.2 List of graphics data formats 1033

J SAM/BAM export format specification 1034

J.1 Flags 1035

K Gene expression annotation files and microarray data formats 1038

K.1 GEO (Gene Expression Omnibus) 1038

K.2 Affymetrix GeneChip 1041

K.3 Illumina BeadChip 1042

K.4 Gene ontology annotation files 1044

K.5 Generic expression and annotation data file formats 1044

L Custom codon frequency tables **1048**

M Comparison of track comparison tools **1049**

Bibliography **1051**

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 Program download . . . 17
1.2.2 Installation on Microsoft Windows . . . 17
1.2.3 Installation on macOS . . . 18
1.2.4 Installation on Linux with an installer . . . 18
1.3 System requirements . . . 19
1.3.1 Limitations on maximum number of cores . . . 20
1.4 Workbench Licenses . . . 21
1.4.1 Request an evaluation license . . . 22
1.4.2 Download a license using a license order ID . . . 24
1.4.3 Import a license from a file . . . 26
1.4.4 Upgrade license . . . 26
1.4.5 Configure license server connection . . . 29
1.4.6 Download a static license on a non-networked machine . . . 33
1.4.7 Viewing mode . . . 34
1.4.8 Start in safe mode . . . 35
1.5 Plugins . . . 36
1.5.1 Install . . . 36
1.5.2 Uninstall . . . 37
1.5.3 Updating plugins . . . 37
1.6 Network configuration . . . 39
1.7 CLC Server connection . . . 39
1.8 Getting started and latest improvements . . . 41

Welcome to *CLC Genomics Workbench 21.0.1* — a software package supporting your daily bioinformatics work.

We strongly encourage you to read this user manual in order to get the best possible basis for working with the software package.

This software is for research purposes only.

1.1 Contact information and citation

CLC Genomics Workbench is developed by:

QIAGEN Aarhus
Silkeborgvej 2
Prismet
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/), you can contact our customer support via the workbench by going to the menu option:

Help | **Contact Support**

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 how to to find your license information is included in the licenses section of our Frequently Asked Questions (FAQ) area: https://secure.clcbio.com/helpspot/index.php?pg=kb. Information about MUS cover on particular licenses can be found by 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 and add (QIAGEN) to it. For example CLC Main Workbench 20.0 (QIAGEN) or CLC Genomics Workbench 20.0 (QIAGEN). If a location is required by the publisher of the publication,

use (QIAGEN, Aarhus, Denmark).

In the References, cite our web site https://digitalinsights.qiagen.com/according to the preferences of the particular journal you will publish in.

1.2 Download and installation

The *CLC Genomics Workbench* is developed for Windows, macOS and Linux. The software for either platform can be downloaded from https://digitalinsights.qiagen.com/downloads/product-downloads/. To check for available updates of the workbench and plugins, click on **Help** | **Check for Updates...** ().

1.2.1 Program download

Before you download the program you are asked to fill in the **Download** dialog.

In the dialog you must choose:

- Which operating system you use
- Whether you would like to receive information about future releases

When the download of the installer (an application which facilitates the installation of the program) is complete, follow the platform specific instructions below to complete the installation procedure.

1.2.2 Installation on Microsoft Windows

When you have downloaded an installer, locate the downloaded installer and double-click the icon. The default location for downloaded files is your desktop.

Installing the program is done in the following steps:

- On the welcome screen, click **Next**.
- Read and accept the License agreement and click **Next**.
- Choose where you would like to install the application and click **Next**.
- Choose a name for the Start Menu folder used to launch *CLC Genomics Workbench* and click **Next**.
- Choose if *CLC Genomics Workbench* should be used to open CLC files and click **Next**.
- Choose where you would like to create shortcuts for launching *CLC Genomics Workbench* and click **Next**.
- Choose if you would like to associate .clc files to *CLC Genomics Workbench*. If you check this option, double-clicking a file with a "clc" extension will open the *CLC Genomics Workbench*.
- Wait for the installation process to complete, choose whether you would like to launch *CLC Genomics Workbench* right away, and click **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

Starting the installation process is done in the following way: When you have downloaded an installer, locate the downloaded installer and double-click the icon. The default location for downloaded files is your desktop.

Launch the installer by double-clicking on the "*CLC Genomics Workbench*" icon.

Installing the program is done in the following steps:

- On the welcome screen, click **Next**.
- Read and accept the License agreement and click **Next**.
- Choose where you would like to install the application and click **Next**.
- Choose if *CLC Genomics Workbench* should be used to open CLC files and click **Next**.
- Choose whether you would like to create desktop icon for launching *CLC Genomics Workbench* and click **Next**.
- Choose if you would like to associate .clc files to *CLC Genomics Workbench*. If you check this option, double-clicking a file with a "clc" extension will open the *CLC Genomics Workbench*.
- Wait for the installation process to complete, choose whether you would like to launch *CLC Genomics Workbench* right away, and click **Finish**.

When the installation is complete the program can be launched from your Applications folder, or from the desktop shortcut you chose to create. If you like, you can drag the application icon to the dock for easy access.

1.2.4 Installation on Linux with an installer

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

```
# sh CLCGenomicsWorkbench_21_0_1.sh
```

Installing the program is done in the following steps:

- On the welcome screen, click **Next**.
- Read and accept the License agreement and click **Next**.
- Choose where you would like to install the application and click **Next**.
  For a system-wide installation you can choose for example /opt or /usr/local. If you do not have root privileges you can choose to install in your home directory.

- Choose where you would like to create symbolic links to the program
  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 **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:

```
# clcgenomicswb21
```

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

```
# ./clcgenomicswb21
```

1.3 System requirements

- Windows 7, Windows 8, Windows 10, Windows Server 2012, Windows Server 2016 and Windows Server 2019
- OS X 10.11 and macOS 10.12, 10.13 and 10.14
- Linux: RHEL 7 and later, SUSE Linux Enterprise Server 12 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
- 2 GB RAM required
- 4 GB RAM recommended
- 1024 x 768 display required
- 1600 x 1200 display recommended
- Intel or AMD CPU required

Special system requirements for the 3D Molecule Viewer

- **Requirements**
  - A graphics card capable of supporting OpenGL 2.0.
  - Updated graphics drivers. Please make sure the latest driver for the graphics card is installed.

- **Recommendations**
  - A discrete graphics card from either Nvidia or AMD/ATI. Modern integrated graphics cards (such as the Intel HD Graphics series) may also be used, but these are usually slower than the discrete cards.

  Indirect rendering (such as x11 forwarding through ssh), remote desktop connection/VNC, and running in virtual machines is not supported.

Special system requirements for read mapping . The numbers below give minimum and recommended memory for systems running mapping and analysis tasks. The requirements suggested are based on the genome size.

- **E. coli K12 (4.6 megabases)**
  - Minimum: 2 GB RAM
  - Recommended: 4 GB RAM
- **C. elegans (100 megabases)** and **Arabidopsis thaliana (120 megabases)**
  - Minimum: 2 GB RAM
  - Recommended: 4 GB RAM
- **Zebrafish (1.5 gigabases)**
  - Minimum: 2 GB RAM
  - Recommended: 4 GB RAM
- **Human (3.2 gigabases)** and **Mouse (2.7 gigabases)**
  - Minimum: 6 GB RAM
  - Recommended: 8 GB RAM

Special 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. See http://resources.qiagenbioinformatics.com/white-papers/White_paper_on_de_novo_assembly_4.pdf for examples of the memory usage of various data sets.

1.3.1 Limitations on maximum number of cores

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

For static licenses, there is a limitation on the number of logical cores on the computer. If there are more than 64 logical cores, the *CLC Genomics Workbench* cannot be started. In this case, a network license is needed (read more at https://digitalinsights.qiagen.com/licensing/).

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 1.1, will be presented to you. The **License Assistant** can be 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 ( )

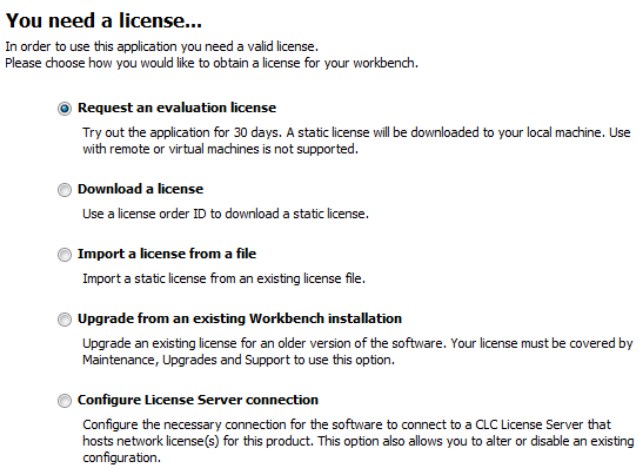


Figure 1.1: *The License Assistant provides access to licensing options.*

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 license.
- **Download a license** Use the license order ID provided when you purchase the software to download and install a license file.
- **Import a license from a file** Import an existing 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 license file.
- **Configure License Server connection** If your organization has a CLC License Server, 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 1.4.6.

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 1.4.7) for further information about this.

1.4.1 Request an evaluation license

We offer a fully functional version of the *CLC Genomics Workbench* for evaluation purposes, free of charge. Each person is entitled to a 14-day trial of *CLC Genomics Workbench*. If you are unable to complete your assessment in the available time, please send an email to bioinformaticssales@qiagen.com to request an additional evaluation period.

When you choose the option **Request an evaluation license**, you will see the dialog shown in figure 1.2.

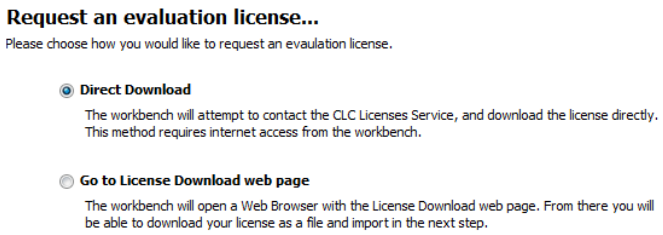


Figure 1.2: *Choose between downloading a license directly, or opening the license download form in a web browser.*

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 1.3 will appear if the license is successfully downloaded and installed.

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.

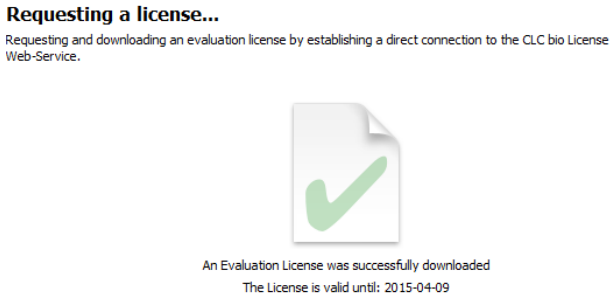


Figure 1.3: *A license has been successfully downloaded and installed for use.*

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 1.4.

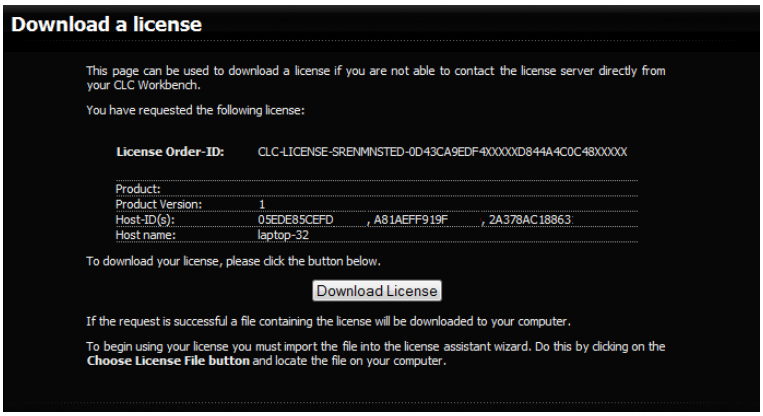


Figure 1.4: *The license download form opened in a web browser.*

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 1.5.

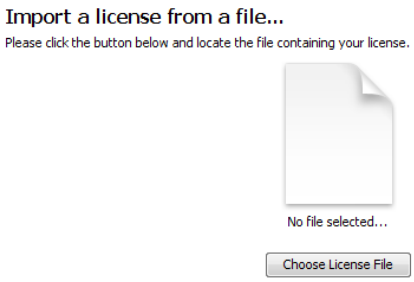


Figure 1.5: *Importing the license file downloaded from the web page.*

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.

1.4.2 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 1.6. 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).

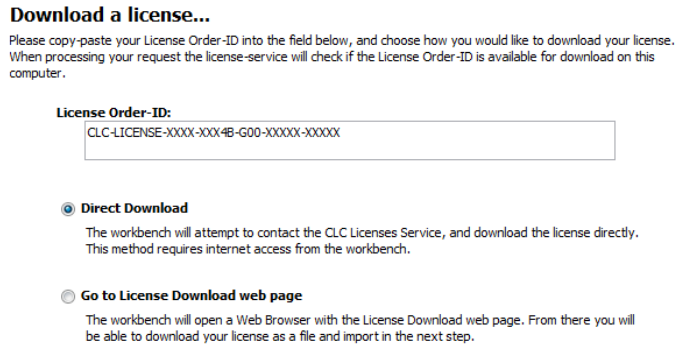


Figure 1.6: *Enter a license order ID into the text field and then click on the Next button.*

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 1.7 will appear if the license is successfully downloaded and installed.

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 1.8.

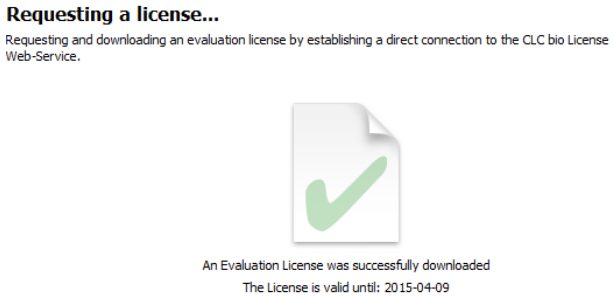


Figure 1.7: *A license has been successfully downloaded and installed for use.*



Figure 1.8: *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 1.9.

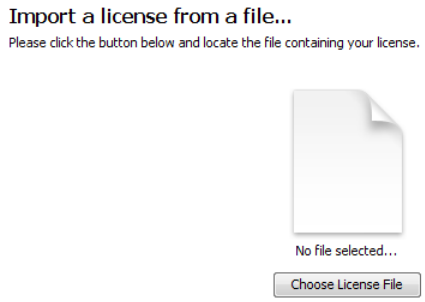


Figure 1.9: *Importing the license file downloaded from the web page.*

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.

1.4.3 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 1.10.

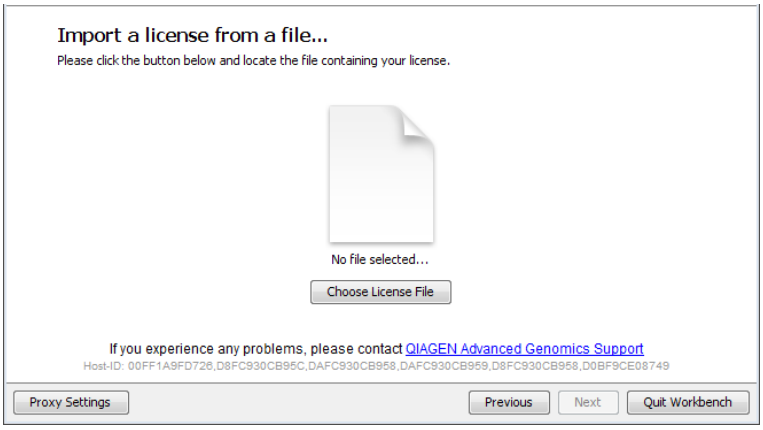


Figure 1.10: *Selecting a license file.*

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 1.4.6.

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 1.11.

When you click on the **Next** button, the Workbench checks if you are entitled to upgrade your

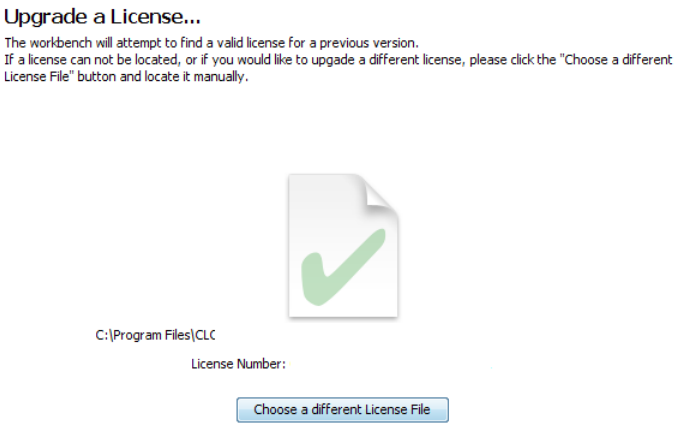


Figure 1.11: *An license from an older installation was found.*

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 it[1], 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 1.12 will be displayed. If there is a problem updating the selected license, a dialog will appear indicating this.

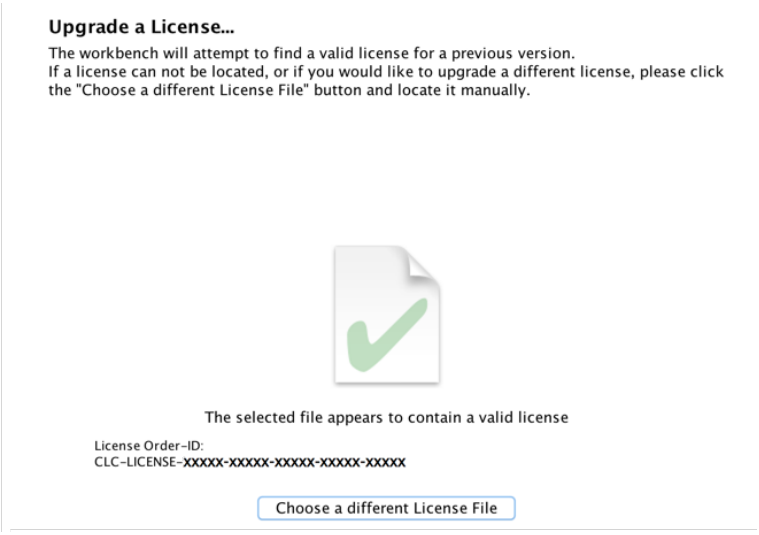


Figure 1.12: *An license from an older installation was found.*

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

[1] In 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.

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 1.13 will appear if the license is successfully downloaded and installed.

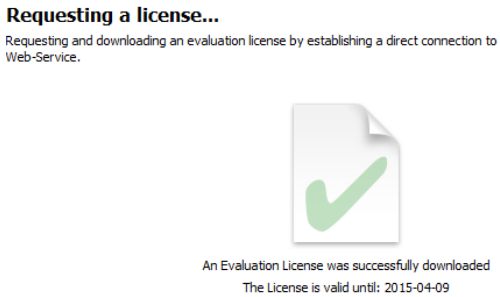


Figure 1.13: *A license has been successfully downloaded and installed for use.*

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 1.14.

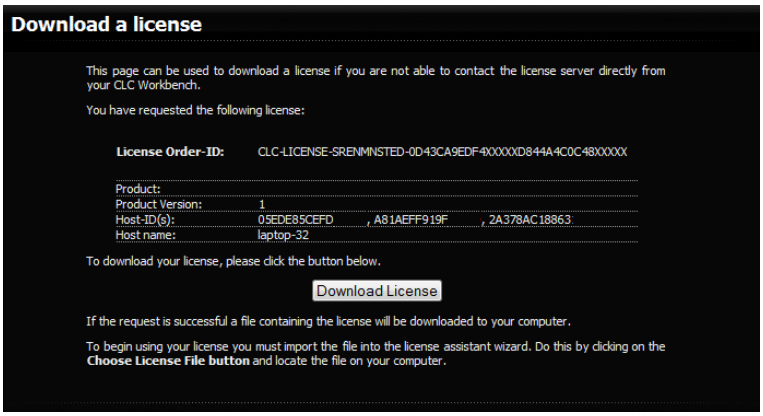


Figure 1.14: *The license download form opened in a web browser.*

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

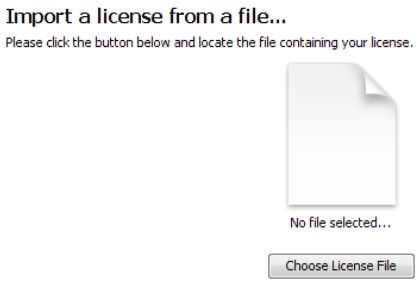


Figure 1.15: *Importing the license file downloaded from the web page.*

Back in the Workbench window, you will now see the dialog shown in 1.15.

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 server connection

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

To configure the Workbench to connect to a CLC License Server, select the **Configure License Server connection** option and click on the **Next** button. A dialog for the license server connection configuration is then presented, as shown in figure 1.16.

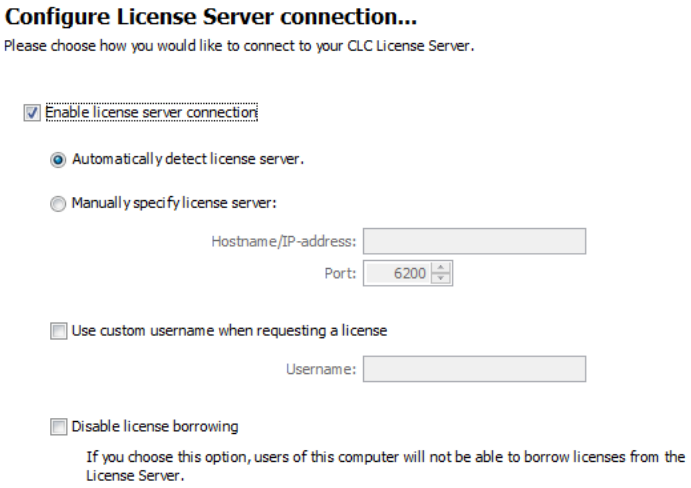


Figure 1.16: *Connecting to a CLC License Server.*

The options in that dialog are:

- **Enable license server connection**. This box must be checked for the Workbench is to contact the CLC License Server to get a license for *CLC Genomics Workbench*.
- **Automatically detect license server**. By checking this option the Workbench will look for a CLC License Server 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 License Server using the **Manually specify license server** option instead.
- **Manually specify license server**. Select this option to enter the details of the machine the CLC License Server software is running on, specifically:
  - **Host name**. The address of the machine the CLC Licenser Server software is running on.
  - **Port**. The port used by the CLC License Server to receive requests.
- **Use custom username when requesting a license**. Optional. If this is checked, a username can be entered. That will be passed to the CLC License Server instead of the username of the account being used to run the Workbench.
- **Disable license borrowing on this computer**. Check this box if you do not want users of the computer to borrow a license. See section 1.4.5 for further details.

Special note on modules needing a license

This note concerns CLC Genomics Workbench 11.0 and higher, Biomedical Genomics Workbench 5.0 and higher and CLC Main Workbench 8.0 and higher.

A valid module license is needed to start a module tool, or a workflow including a module tool. Module licenses obtained through a License Server connection will be valid for four hours after starting the tool or the workflow. A process started (whether a module tool or a workflow including a module tool) will always be completed, even if its completion exceeds the four hours period where the license is valid.

If the tool or the workflow completes before the four hour validity period, it is possible to start a new tool or a workflow, and this will always refresh the validity of the license to a full four hours period. However, if the tool or the workflow completes after the four hour validity period, a new license will need to be requested after that to start the next tool or workflow.

These measures ensure that more licenses are available to active users, rather than blocked on an inactive computer, i.e., where the workbench would be open but not in use.

Borrowing a license

A *CLC Genomics Workbench* using a network license normally needs to maintain a connection to the CLC License Server. However, if allowed by the network license administrator, network licenses can be *borrowed* for offline use. During the period a license has been borrowed, there will be one less network license available for other users.

If CLC License Server administrator has chosen not to allow the borrowing of network licenses, then the information in this section is not relevant.

The Workbench must be connected to the CLC License Server at the point when the license is borrowed. 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 dialog shown in figure 1.17.

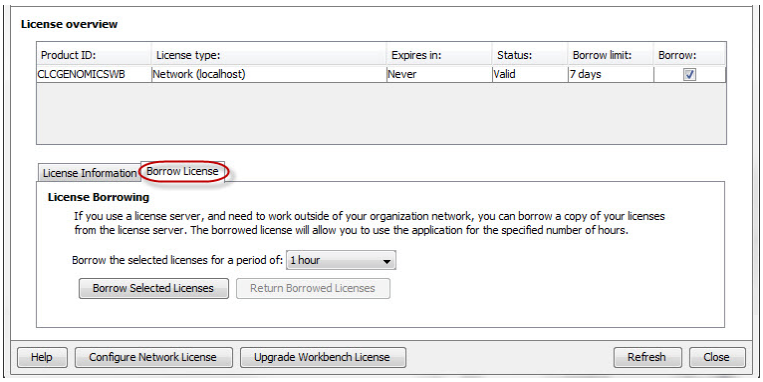


Figure 1.17: *Borrow a license.*

3. Select the license(s) that you wish to borrow by clicking in the checkboxes in the Borrow column in the License overview panel.

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.

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 contact the CLC Licence Server.

You can return borrowed licenses early if you wish by started up the **License Manager**, opening the "Borrow License" tab, and clicking on the **Return Borrowed Licenses** button.

Common issues when using a network license

- No license available at the moment If all licenses are in use, you will see a dialog like that shown in figure 1.18 when you start up the Workbench.

  You will need to wait for at least one license to be returned before you can continue to work with a fully functional copy of the software. If running out of licenses is a frequent issue, you may wish to discuss this with your CLC License Server administrator.

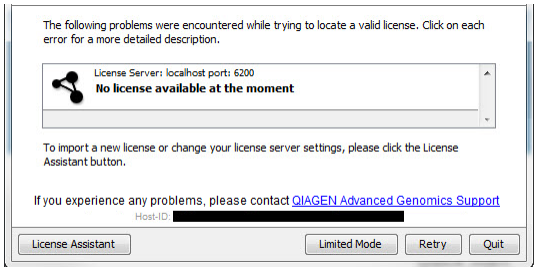


Figure 1.18: *This window appears when there are no available network licenses for the software you are running.*

Clicking on the **Viewing Mode** button in the dialog allows you to run the *CLC Genomics Workbench* for viewing data, and for basic analyses, import and export. Please see section 1.4.7 for further details.

- Lost connection to the CLC License Server If the Workbench connection to the CLC License Server is lost, you will see a dialog like that shown in figure 1.19.

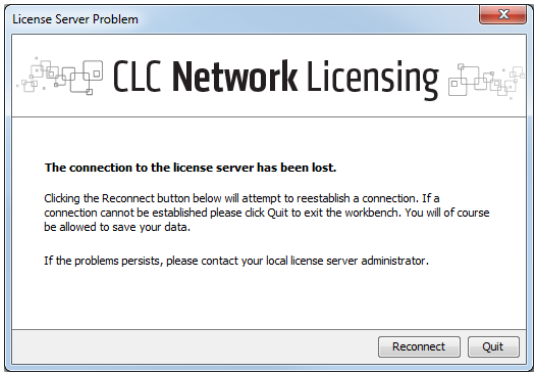


Figure 1.19: *Here, the Workbench is unable to establish a connection to a CLC License Server.*

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

If you have successfully contacted the CLC License Server from your Workbench previously, please consider discussing this issue with your CLC License Server administrator or your local IT support, for example, making sure that the CLC License Server is running and that your Workbench is able to connect to it.

There may be situations where you wish to use a different license or view information about the license(s) the Workbench is currently using. To do this, open the License Manager using the menu option:

Help | License Manager ( )

The license manager is shown in figure 1.20.

This License Manager can be used to:

- See information about the license (e.g. the license type, when it expires, etc.)

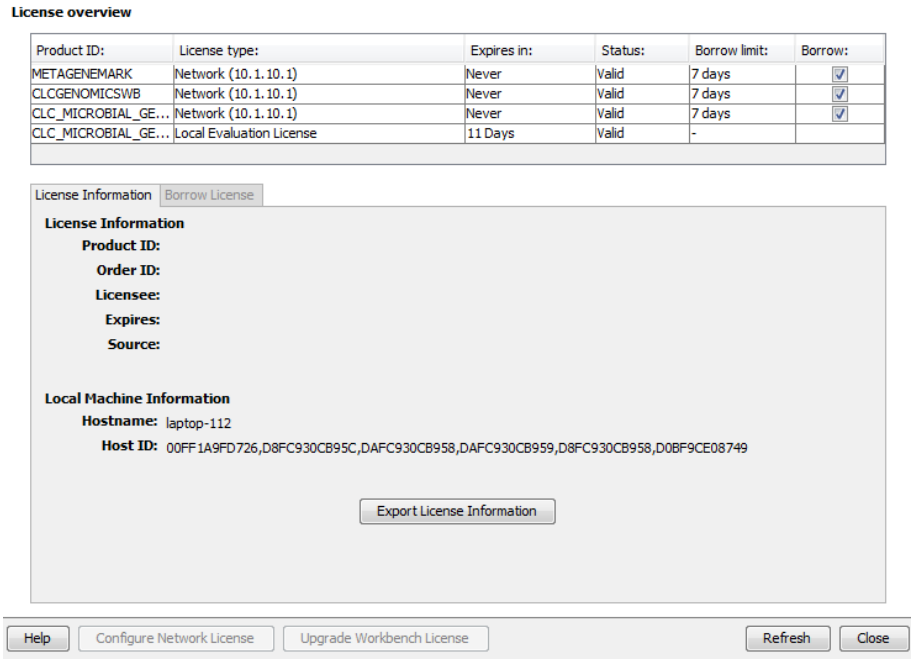


Figure 1.20: *The License Manager provides information about licenses being used and access to other license-related functionality.*

- Configure the connect to a CLC License Server. Click on the **Configure Network License** button at the lower left corner to open the dialog seen in figure 1.16.
- Upgrade from an evaluation license. Click on the **Upgrade Workbench License** button to open the dialog shown in figure 1.1.
- Export license information to a text file.
- Borrow a license, relevant when using a network license.

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 server connection** in the dialog. When you restart the Workbench, you can set up the new license as described in section 1.4.

1.4.6 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[2]:

  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 selct 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.

1.4.7 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 Toolbox, as illustrated in figure 1.21.

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 1.22. 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.

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.

[2]For CLC Genomics Workbench 5.x and earlier or CLC Main Workbench 6.7.x and earlier, the license download page URL is http://licensing.clcbio.com/LmxWSv1/GetLicenseFile

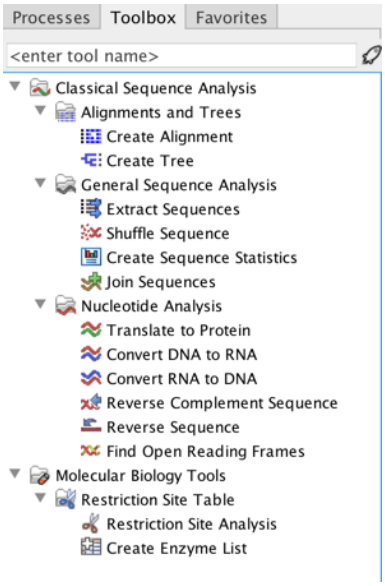


Figure 1.21: *Bioinformatics tools available when using Viewing Mode are found in the Toolbox.*

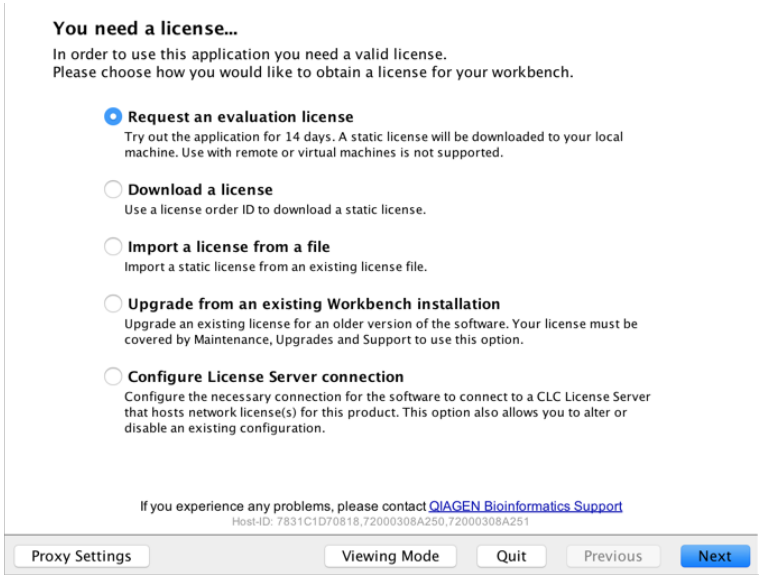


Figure 1.22: *Click on the Viewing Mode button at the bottom of the License Manager window to launch the Workbench in Viewing Mode.*

1.4.8 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

When you install *CLC Genomics Workbench*, it has a standard set of features. However, you can upgrade and customize the program using a variety of plugins.

As the range of plugins is continuously updated and expanded, they will not be listed here. Instead we refer to https://digitalinsights.qiagen.com/products-overview/plugins/ for a full list of plugins with descriptions of their functionalities.

Note: In order to install plugins and modules, the Workbench must be run in administrator mode. On Linux and Mac, it means you must be logged in as an administrator. On Windows, you can do this by right-clicking the program shortcut and choosing "Run as Administrator".

Plugins are installed and uninstalled using the plugin manager.

Help in the Menu Bar | Plugins... ( ) or Plugins ( ) in the Toolbar

The plugin manager has two tabs at the top:

- **Manage Plugins.** This is an overview of plugins that are installed.
- **Download Plugins.** This is an overview of available plugins on QIAGEN Aarhus server.

1.5.1 Install

To install a plugin, click the **Download Plugins** tab. This will display an overview of the plugins that are available for download and installation (see figure 1.23).

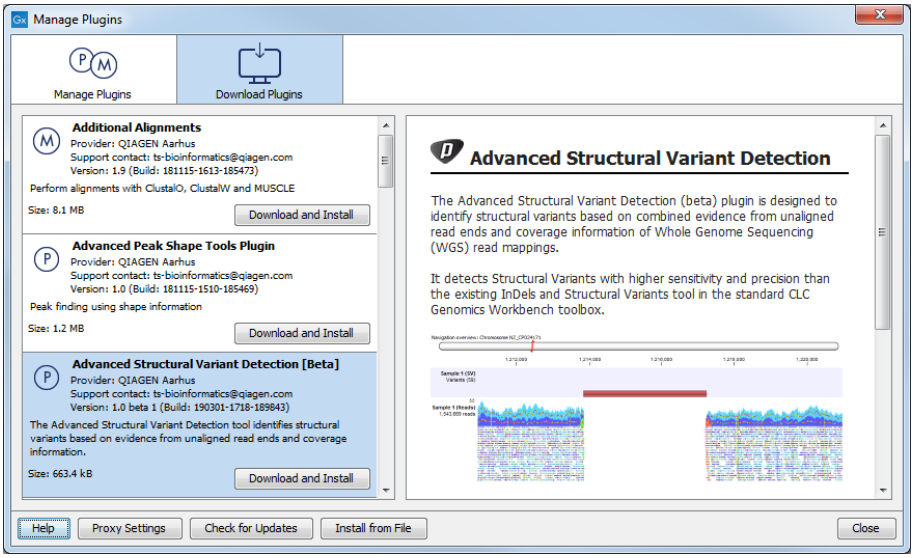


Figure 1.23: *The plugins that are available for download.*

Select the plugin of interest to display additional information about the plugin on the right side of the dialog. Click **Download and Install** to add the plugin functionalities to your workbench.

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 the plugin is not shown on the server but you have the installer file on your computer (for example if you have downloaded it from our website), you can install the plugin by clicking the **Install from File** button at the bottom of the dialog and specifying the plugin *.cpa file saved on your computer.

When you close the dialog, you will be asked whether you wish to restart the workbench. The plugin will not be ready for use until you have restarted.

1.5.2 Uninstall

Plugins are uninstalled using the plugin manager:

Help in the Menu Bar | Plugins... ( ) or Plugins ( ) in the Toolbar

This will open the dialog shown in figure 1.24.

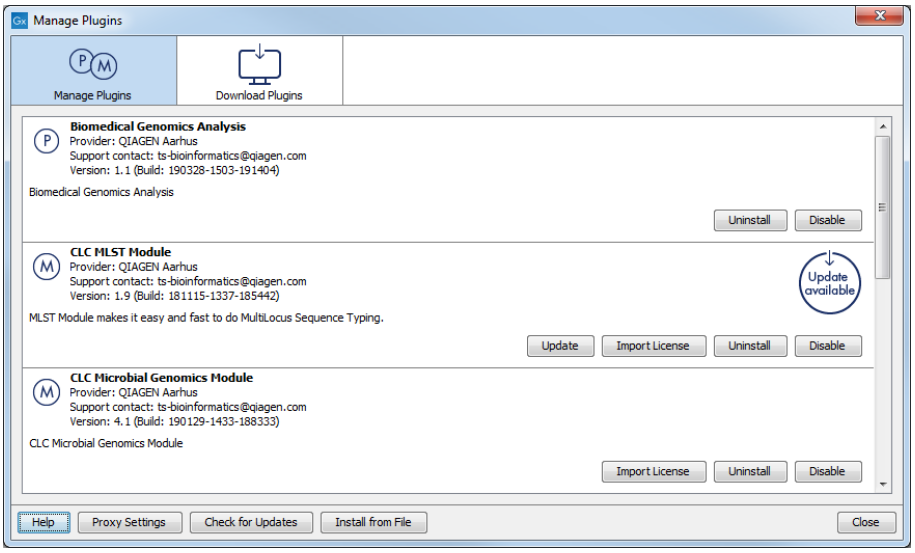


Figure 1.24: *The plugin manager with plugins installed.*

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 as shown in figure 1.25.

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

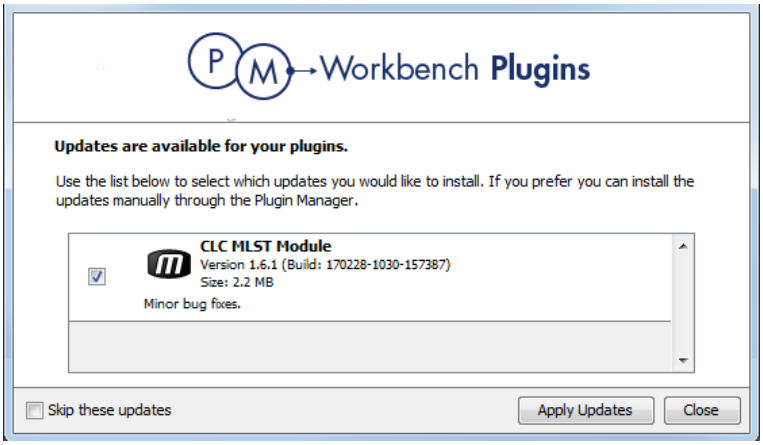


Figure 1.25: *Plugin updates.*

manager (see figure 1.24).

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, open the workbench, go to **Edit | Preferences** and choose the **Advanced** tab (figure 1.26).

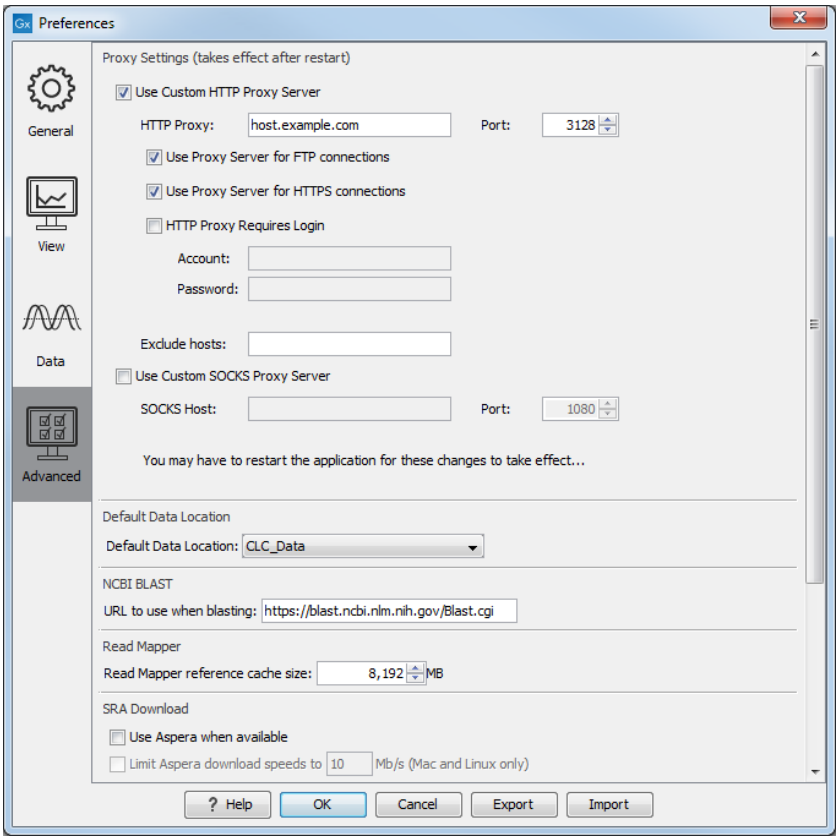


Figure 1.26: *Adjusting proxy preferences.*

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

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

Exclude hosts can be used if there are some hosts that should be contacted directly and not through the proxy server. The value can be a list of hosts, each separated by a `|`, and in addition a wildcard character `*` can be used for matching. For example: `*.foo.com|localhost`.

If you have any problems with these settings you should contact your systems administrator.

1.7 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 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 from the Toolbox.
- External applications configured and enabled on the *CLC Server* will be available to launch from the Toolbox, and to include in workflows.

To log into a *CLC Server* or to check on the status of an existing connection, go to:

File | **CLC Server Connection (** **)**

This will bring up a login dialog as shown in figure 1.27.

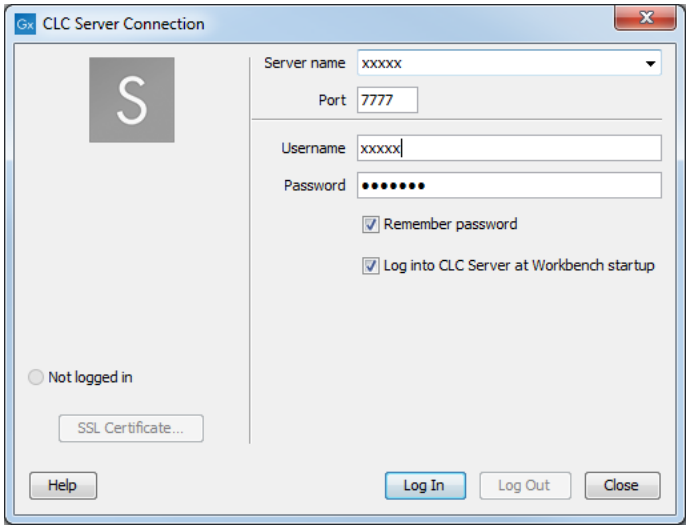


Figure 1.27: *The CLC Server Connection dialog.*

Your server administrator should be able to provide you with the necessary details to fill in the fields. When you click on the **Log In** button, the Workbench will connect to the *CLC Server* if your credentials are accepted.

Your username and the server details will be saved between Workbench sessions. If you wish your password to be saved also, click in the box beside the **Remember password** box.

If you wish the Workbench to connect to the server automatically on startup, then check the box beside the option **Log into CLC Server at Workbench startup**. This option is only enabled when the **Remember password** option has been selected.

Further information about working with a *CLC Server* from a CLC Workbench is available in this manual:

- Launching tasks on a *CLC Server* is described in section 9.1.1.
- Monitoring processes sent to the *CLC Server* from a CLC Workbench is described in section 2.4.
- Viewing and working with data held on a *CLC Server* is described in section 3.1.1, and deleting data held on a *CLC Server* is described in section 3.1.7.

- Importing data to a *CLC Server* and exporting data held on a *CLC Server* is described in section 6.9.

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 in detail at http://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=User_authentication_using_Workbench.html.

1.8 Getting started and latest improvements

CLC Genomics Workbench includes an extensive **Help** function, which can be found in the **Help** menu of the program's **Menu bar** (or by pressing F1).

Tutorials describing hands-on examples of how to use the individual tools and features of the *CLC Genomics Workbench* can be found at https://digitalinsights.qiagen.com/support/tutorials/.We also recommend our **Online presentations** where a product specialist demonstrates our software. This is a very easy way to get started using the program. Read more about video tutorials and other online presentations here: http://tv.qiagenbioinformatics.com/.

Finally, *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/.

History of the CLC Workbenches In November 2005, CLC bio releases two Workbenches: *CLC Free Workbench* and *CLC Protein Workbench*. *CLC Protein Workbench* is developed from the free version, giving it the well-tested user friendliness and look & feel with a range of more advanced analyses.

In March 2006, *CLC DNA Workbench* (formerly *CLC Gene Workbench*) and *CLC Main Workbench* are added to the product portfolio of CLC bio. Like *CLC Protein Workbench*, *CLC DNA Workbench* builds on *CLC Free Workbench*. It shares some of the advanced product features of *CLC Protein Workbench*, and has additional advanced features. *CLC Main Workbench* holds all basic and advanced features of the *CLC Workbenches*.

In June 2007, *CLC RNA Workbench* is released as a sister product of *CLC Protein Workbench* and *CLC DNA Workbench*. *CLC Main Workbench* now also includes all the features of *CLC RNA Workbench*.

In March 2008, *CLC Free Workbench* changes name to *CLC Sequence Viewer*.

In June 2008, the first version of the *CLC Genomics Workbench* is released due to an extraordinary demand for software capable of handling sequencing data from all new high-throughput sequencing platforms such as Roche-454, Illumina and SOLiD in addition to Sanger reads and hybrid data.

In December 2006, CLC bio releases a *Software Developer Kit* which makes it possible for anybody with a knowledge of programming in Java to develop plugins. The plugins are fully integrated with the CLC Workbenches and the Viewer and provide an easy way to customize and extend their functionalities.

In April 2012, *CLC Protein Workbench*, *CLC DNA Workbench* and *CLC RNA Workbench* are discontinued. All customers with a valid license for any of these products are offered an upgrade to the *CLC Main Workbench*.

In February 2014, CLC bio expands the product repertoire with the release of *CLC Drug Discovery Workbench*, a product that enables studies of protein-ligand interactions for drug discovery.

In April 2014, CLC bio releases *CLC Cancer Research Workbench*, a product that contains streamlined data analysis workflows with integrated trimming and quality control tailored to meet the requirements of clinicians and researchers working within the cancer field.

In April 2015, *CLC Cancer Research Workbench* is renamed to *Biomedical Genomics Workbench* to reflect the inclusion of tools addressing the requirements of clinicians and researchers working within the hereditary disease field in addition to the tools designed for those working within the cancer field.

In June 2017, Viewing Mode is introduced in all commercial CLC Workbenches. This mode is available when a Workbench is launched without a valid license. In this mode, data can be viewed and some basic analyses equivalent to those available in the free CLC Sequence Viewer, can be run.

In January 2018, **CLC Drug Discovery Workbench** is discontinued.

In November 2018, the *Biomedical Genomics Analysis* plugin is released for use with *CLC Genomics Workbench*. With the Biomedical Genomics Analysis plugin installed, CLC Genomics Workbench becomes the delivery mechanism for all biomedical analyses previously delivered by *Biomedical Genomics Workbench* and some associated plugins. *Biomedical Genomics Workbench* is correspondingly discontinued, and all customers with valid licenses for *Biomedical Genomics Workbench* can use them for *CLC Genomics Workbench*.

Part II

Core Functionalities

Chapter 2

User interface

Contents

2.1 View Area 46
2.1.1 Open view 46
2.1.2 History and Element Info views 47
2.1.3 Close views 48
2.1.4 Save changes in a view 49
2.1.5 Undo/Redo 50
2.1.6 Arrange views in View Area 50
2.1.7 Moving a view to a different screen 51
2.1.8 Side Panel 52
2.2 Zoom and selection in View Area 54
2.2.1 Zoom in 54
2.2.2 Zoom out 55
2.2.3 Selecting, panning and zooming 55
2.3 Toolbox and Favorites tabs 56
2.3.1 Toolbox tab 56
2.3.2 Favorites tab 56
2.4 Processes tab and Status bar 57
2.5 Workspace 59
2.6 List of shortcuts 60

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 3 tabs:
  - **Processes** Running and finished processes are listed under this tab. (Section 2.4)

  - **Toolbox** Analysis tools and installed workflows are listed and can be launched from under this tab. (Section 2.3.1)
  - **Favorites** Tools you use most often are listed here, and you can add tools you want, for quick access. (Section 2.3.2)

- **View Area** Data and workflow designs can be opened in this area for viewing and editing. (Section 2.1) When elements are open in the View Area, a **Side Panel** with configuration options will be present on the right hand side. (Section 4.6)
- **Menu bar** and **Tool bar** 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)

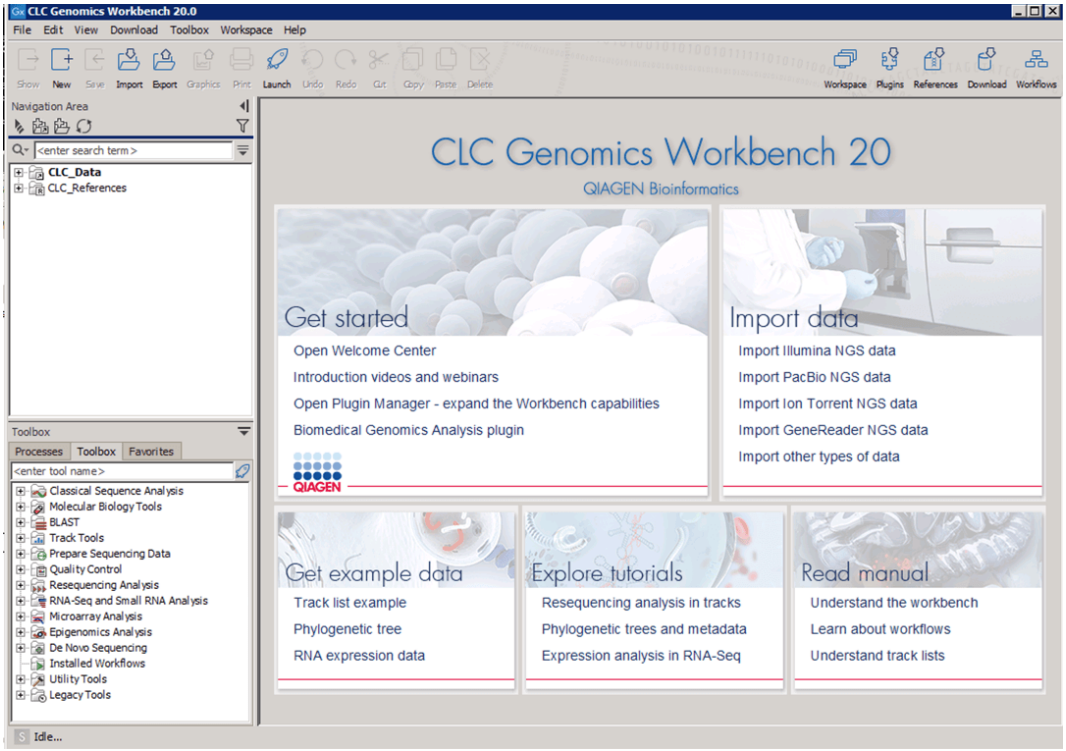


Figure 2.1: *The* **CLC Workbench** *interface includes the Navigation Area in the top left, several tabs in the Toolbox area at the bottom left, a large viewing area on the right, menus and toolbars at the top, and a status bar at the bottom.*

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, what is shown in the Toolbox area can be configured using the menu options found 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 the central part of the screen, displaying your current work. The View Area may consist of one or more **Views**, represented by **tabs** at the top of the View Area. In figure 2.2, four views are displayed: three as tabs in the upper view, and one in an horizontal split view. The tab currently selected, i.e., active, is indicated by a blue bar underneath the tab (here the bottom tab open in the bottom view).

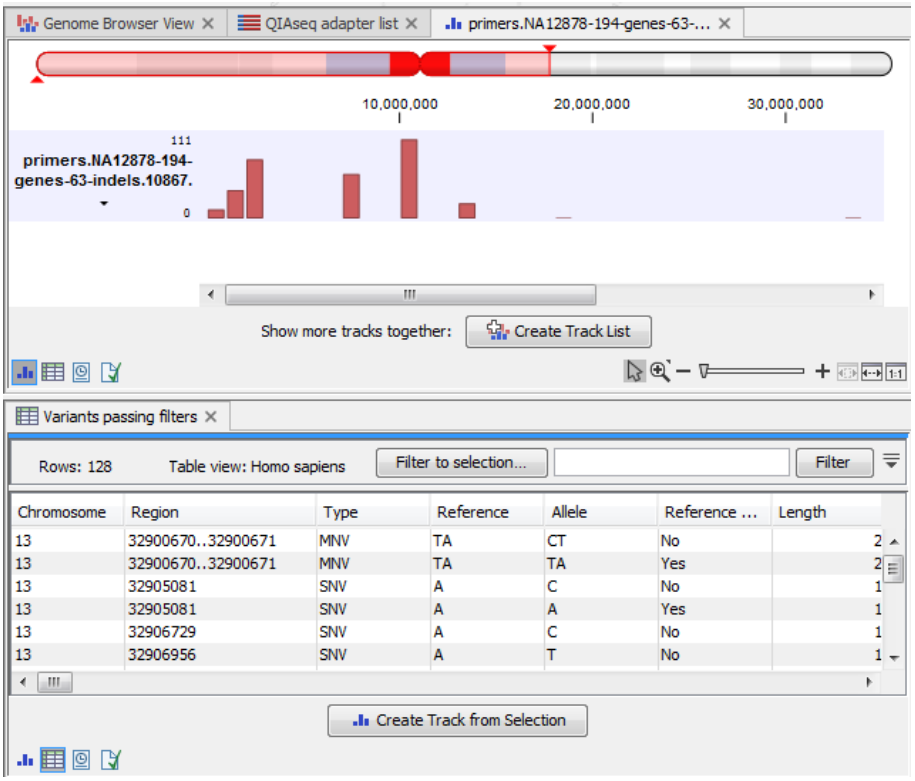


Figure 2.2: *A View Area can enclose several views, each view indicated with a tab.*

Switch tabs in View Area using the following shortcuts Ctrl + PageUp or PageDown (or ⌘ + PageUp or PageDown on Mac).

Several operations can be performed by right-click menus that can be activated from the tab, or by using the icon list at the bottom of each view.

2.1.1 Open view

Elements

Opening an element can be done in a number of ways:

double-click an element in the Navigation Area

or **select an element in the Navigation Area** | **Show** or **Ctrl + O (⌘ + B on Mac)**

Opening an element while another element is already open in the View Area will show the new element in front of the other. The element that was already open can be brought to front by clicking its tab.

Views

Each element can be shown in different ways. A sequence, for example, can be shown as linear, circular, text, etc.

For example, to see a linear sequence in a circular view, open the sequence as linear in the View Area and

Click Show As Circular ( ) at the lower left part of the view

The buttons used for switching views are shown in figure 2.3. They are element-dependent, meaning that different elements may have different buttons available. You can switch from one to the other sequentially by clicking Ctrl + Shift + PageUp or Ctrl + Shift + PageDown.



Figure 2.3: *The buttons shown at the bottom of a view of a nucleotide sequence. You can click the buttons to change the view to a circular view or a history view.*

Split views

If the sequence is already open in a linear view ( ), and you wish to see both a circular and a linear view, you can split the views very easily:

Press Ctrl (⌘ on Mac) while you | Click Show As Circular ( ) at the lower left part of the view

This will open a split view with a linear view at the bottom and a circular view at the top (see 12.5).

You can also show a circular view of a sequence without opening the sequence first:

Select the sequence in the Navigation Area | Show ( ) | As Circular ( )

2.1.2 History and Element Info views

History view

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

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

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 2.4).

The summary information shown in the table for each operation is:

- **Description** The operation performed
- **User** The username of the person who performed the operation. If you import data created by another person in a CLC Workbench, 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.

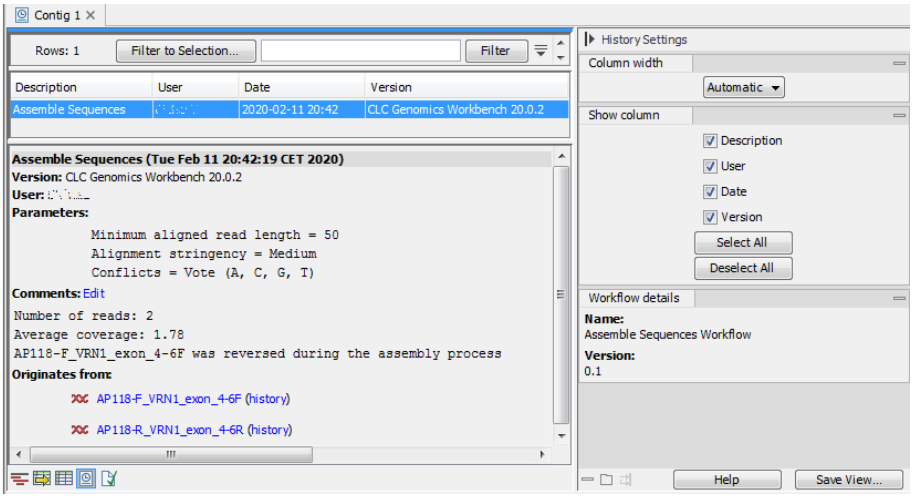


Figure 2.4: *An element's history.*

In addition to the fields above, the following fields are displayed 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** The elements that the current element originates from. Clicking the name of an element here selects it in the Navigation Area. Clicking the "history" link opens that element with its History view shown.

If the element was created as a result of running a workflow, the name and version of the workflow is sown in the side panel under the Workflow details tab.

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 12.4. For further information about metadata associations, see section 10.3.2.

2.1.3 Close views

When a view is closed, the **View Area** remains open as long as there is at least one open view.

A view is closed by:

Right-click the tab | **Close** or **Select the view** | **Ctrl + W**

By right-clicking a tab, the following close options exist (figure 2.5).

- **Close.** See above.

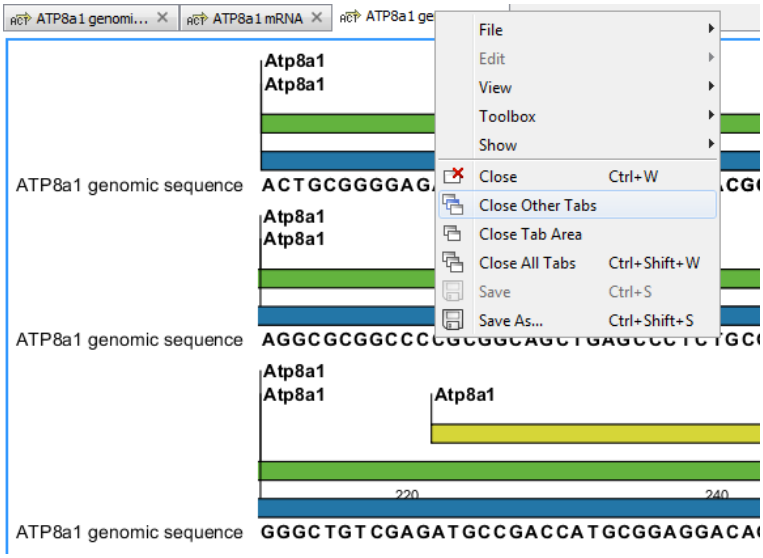


Figure 2.5: *By right-clicking a tab, several close options are available.*

- **Close Other Tabs.** Closes all other tabs, in all tab areas, except the one that is selected.
- **Close Tab Area.** Closes all tabs in the tab area, but not the tabs that are in split view.
- **Close All Tabs.** Closes all tabs, in all tab areas. Leaves an empty workspace.

2.1.4 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).

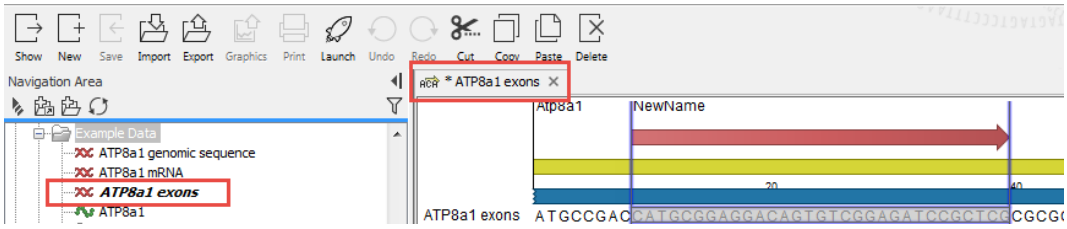


Figure 2.6: *An * on a tab name always indicates that the view is unsaved. In this case, an existing element was edited but not saved yet, so the element's name is also highlighted in bold and italic in the Navigation Area.*

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.

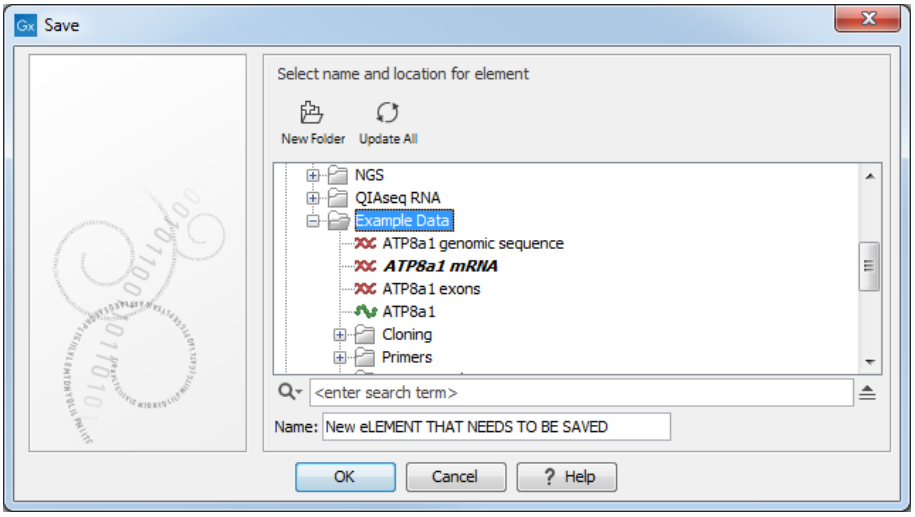


Figure 2.7: *Save dialog. The new element has been name "New element that needs to be saved" and will be saved in the "Example Data" folder.*

2.1.5 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 3.1.7).

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

2.1.6 Arrange views in View Area

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.

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.*

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**

or **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**

2.1.7 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.

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.

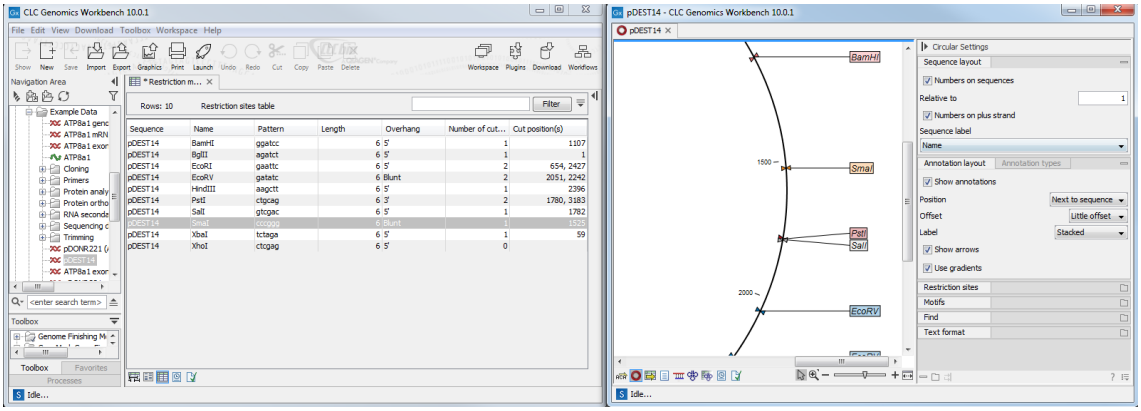


Figure 2.9: *Showing the table on one screen while the sequence is displayed on another screen. Clicking the table of open reading frames causes the view on the other screen to follow the selection.*

2.1.8 Side Panel

The **Side Panel** allows you to change the way the content of a view is displayed. The options in the Side Panel depend on the kind of data in the view, and they are described in the relevant sections about sequences, alignments, trees etc.

Figure 2.10 shows the default Side Panel for a protein sequence. It is organized into **palettes**.

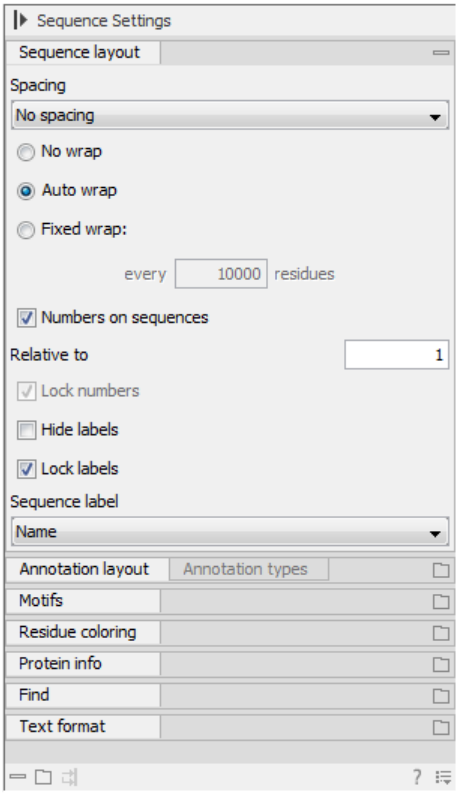


Figure 2.10: *The default view of the Side Panel when opening a protein sequence.*

In this example, there is one palette for Sequence layout, one for Annotation Layout etc. These palettes can be re-organized by dragging the palette name with the mouse and dropping it where you want it to be. They can either be situated next to each other, so that you can switch between

them, or they can be listed on top of each other, so that expanding one of the palettes will push the palettes below further down.

In addition, they can be moved away from the Side Panel and placed anywhere on the screen as shown in figure 2.11.

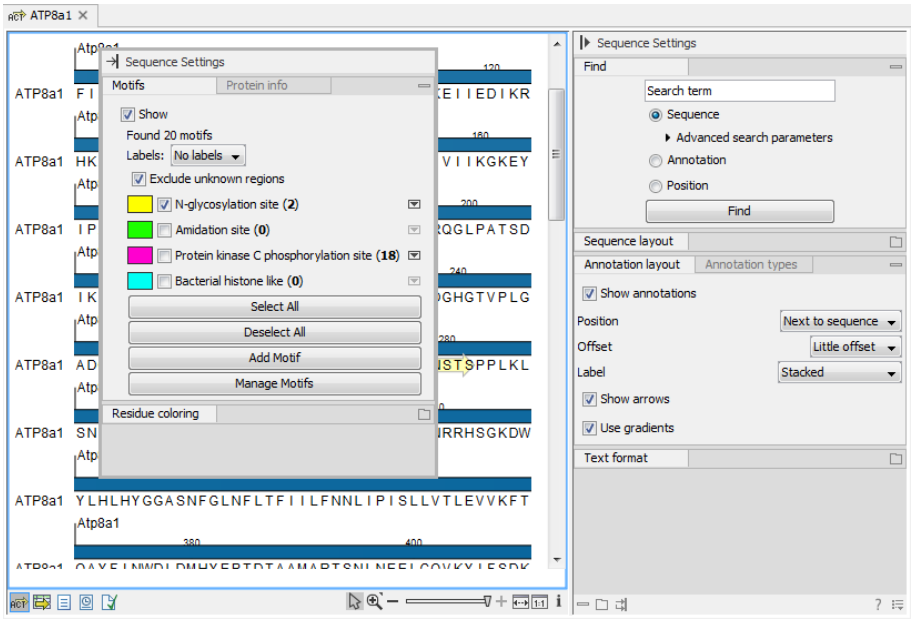


Figure 2.11: *Palettes can be organized in the Side Panel as you like or placed anywhere on the screen.*

In this example, the Motifs palette has been placed on top of the sequence view together with the Residue coloring palette. In the Side Panel to the right, the Find palette has been put on top.

In order to make all palettes dock in the Side Panel again, click the **Dock Side Panel** icon ( ).

You can completely hide the Side Panel by clicking the **Hide Side Panel** icon ( ).

At the bottom of the Side Panel (see figure 2.12) there are a number of icons used to:

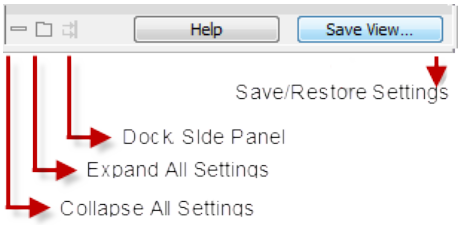


Figure 2.12: *Functionalities found at the bottom of the Side Panel.*

- Collapse all settings ( ).
- Expand all settings ( ).
- Dock all palettes ( )
- Get **Help** for the particular view and settings
- Save the settings of the Side Panel or apply already saved settings. Changes made to the Side Panel, including the organization of palettes, will not be saved when you save the view. Learn how to save Side Panel settings in section 4.6.

2.2 Zoom and selection in View Area

All views except tabular and text views support zooming. Figure 2.13 shows the zoom tools, located at the bottom right corner of the view.

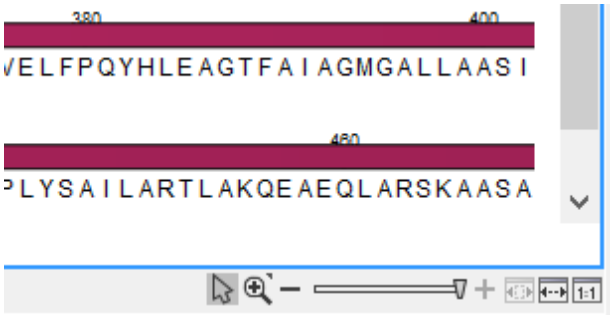


Figure 2.13: *The zoom tools are located at the bottom right corner of the view.*

The zoom tools consist of some shortcuts for zooming to fit the width of the view (), zoom to 100 % to see details (), zoom to a selection (), a zoom slider, and two mouse mode buttons () ().

The slider reflects the current zoom level and can be used to quickly adjust this. For more fine-grained control of the zoom level, move the mouse upwards while sliding.

The sections below describes how to use these tools as well as other ways of zooming and navigating data.

Please note that when working with protein 3D structures, there are specific ways of controlling zooming and navigation as explained in section 14.2.

2.2.1 Zoom in

There are six ways of **zooming in**:

Click Zoom in mode () in the zoom tools (or press Ctrl+2) | click the location in. the view that you want to zoom in on

or **Click Zoom in mode () in the zoom tools | click-and-drag a box around a part of the view | the view now zooms in on the part you selected**

or **Press '+' on your keyboard**

or **Move the zoom slider located in the zoom tools**

or **Click the plus icon in the zoom tools**

The last option for zooming in is only available if you have a mouse with a scroll wheel:

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

Note! You might have to click in the view before you can use the keyboard or the scroll wheel to zoom.

If you press the **Shift** button on your keyboard while in zoom mode, the zoom function is reversed.

If you want to zoom in to 100 % to see the data at base level, click the **Zoom to base level** () icon.

2.2.2 Zoom out

It is possible to zoom out in different ways:

Click Zoom out mode ( ) in the zoom tools (or press Ctrl+3) | click in the view

or **Press '-' on your keyboard**

or **Move the zoom slider located in the zoom tools**

or **Click the minus icon in the zoom tools**

The last option for zooming out is only available if you have a mouse with a scroll wheel:

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

Note! You might have to click in the view before you can use the keyboard or the scroll wheel to zoom.

If you want to zoom out to see all the data, click the **Zoom to Fit** ( ) icon.

If you press **Shift** while clicking in a **View**, the zoom function is reversed. Hence, clicking on a sequence in this way while the **Zoom out** mode toolbar item is selected, zooms in instead of zooming out.

2.2.3 Selecting, panning and zooming

In the zoom tools, you can control which mouse mode to use. The default is **Selection mode** ( ) which is used for selecting data in a view. Next to the selection mode, you can select the **Zoom in mode** as described in section 2.2.1. If you press and hold this button, two other modes become available as shown in figure 2.14:

- **Panning** ( ) is used for dragging the view with the mouse as a way of scrolling.
- **Zoom out** ( ) is used to change the mouse mode so that whenever you click the view, it zooms out.



Figure 2.14: *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.

The mouse modes only apply when the mouse is within the view where they are selected.

The **Selection mode** can also be invoked with the keyboard shortcut Ctrl+1, while the **Panning mode** can be invoked with Ctrl+4.

For some views, if you have made a selection, there is a **Zoom to Selection** ( ) button, which allows you to zoom and scroll directly to fit the view to the selection.

2.3 Toolbox and Favorites tabs

2.3.1 Toolbox tab

The Toolbox tab contains tools and installed workflows, including those distributed via plugins, as well as external applications configured and enabled on a *CLC Genomics Server* that the Workbench is connected to. See figures 2.15 and 2.16.

Launching analyses from the Toolbox can be done by:

- Double-clicking on the tool or workflow name.
- Right-clicking on the tool or workflow name and choosing the option "Run" from the menu that appears.
- Dragging elements from the Navigation Area onto the name of a tool or workflow.

Other methods of launching tools, including using the Quick Launch tool (), are described in section 9.1.

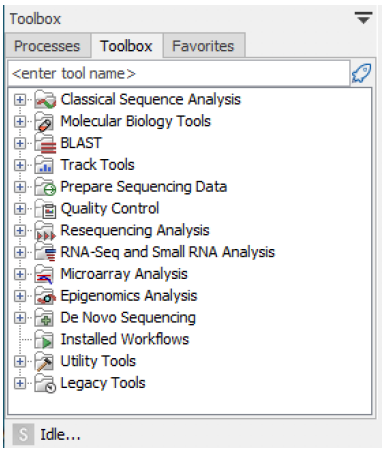


Figure 2.15: *The Toolbox tab in the bottom right contains folders of available tools, and when available, installed workflows. This Workbench is not connected to a CLC Server, as indicated by the grey server icon in the status bar.*

2.3.2 Favorites tab

Tools you want quick access to can be placed under the Favorites tab. Tools you use the most are added to this tab automatically, in a separate section. See figure 2.17.

To manually add tools to the Favorites tab, you can

- Right-click on the tool in the Toolbox and choose the option "Add to Favorites" from the menu that appears, or
- Right-click on the Favorites folder under the Favorites tab, choose the option "Add tools" or "Add group of tools", and then select the tool or tool group to add.

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

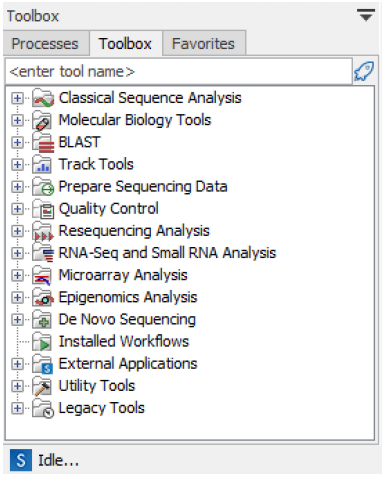


Figure 2.16: *This Workbench is connected to a CLC Server, as indicated by the blue server icon in the status bar. External applications have been configured and enabled on that CLC Server, so an External Applications folder is listed, which contains those external applications. The server icon within that folder's icon is a reminder that these are only available when logged into the CLC Server.*

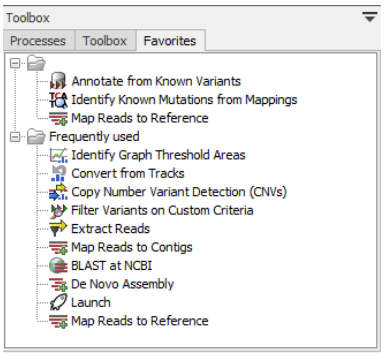


Figure 2.17: *Tools manually added to the Favorites tab are listed at the top. Tools under the "Frequently used" section are added automatically, based on usage.*

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 Processses 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 revealed by clicking on the small icon () next to a given process, as shown in figure 2.18).

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

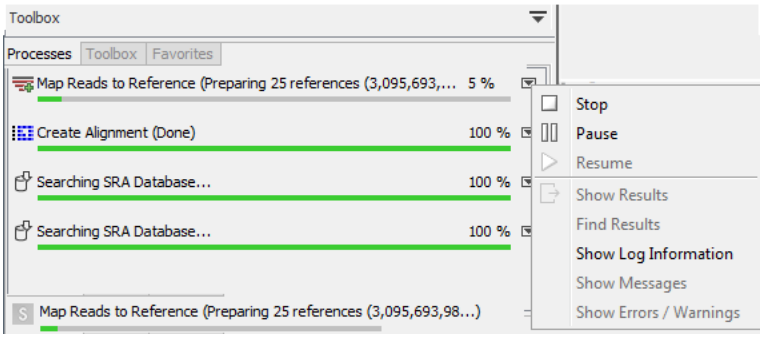


Figure 2.18: *A database search and an alignment calculation are running. Clicking the small icon next to the process lists actions you can take for that process.*

- **Show results** Open the results generated by that process in the Viewing Area. (Relevant if results were saved, as described in section 9.2.)
- **Find results** Highlight the results in the Navigation Area. (Relevant if results were saved, as described in section 9.2.)
- **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 and paused 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:

View | **Remove Finished Processes ( )** .

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 Workspace

If you are working on a project and have arranged the views for this project, you can save this arrangement using **Workspaces**. A Workspace remembers the way you have arranged the views, and you can switch between different workspaces. The name of a non-default workspace is displayed in the Workbench title bar.

The Navigation Area always contains the same data across workspaces. It is, however, possible to open different folders in the different workspaces. Consequently, the program allows you to display different clusters of the data in separate workspaces.

All workspaces are automatically saved when closing down *CLC Genomics Workbench*. The next time you run the program, the workspaces are reopened exactly as you left them.

Note! It is not possible to run more than one version of *CLC Genomics Workbench* at a time. Use two or more workspaces instead.

Create Workspace When working with large amounts of data, it might be a good idea to split the work into two or more workspaces. As default the *CLC Genomics Workbench* opens one workspace. Additional workspaces are created in the following way:

Workspace in the Menu Bar | **Create Workspace** | **enter name of Workspace** | **OK**

Initially, the folders of the **Navigation Area** are collapsed and the View Area is empty and ready to work with.

Select Workspace When there is more than one workspace in the *CLC Genomics Workbench*, there are two ways to switch between them:

Workspace ( ) in the Toolbar | **Select the Workspace to activate**

or **Workspace in the Menu Bar** | **Select Workspace ( )** | **choose which Workspace to activate** | **OK**

Delete Workspace Deleting a workspace can be done in the following way:

Workspace in the Menu Bar | **Delete Workspace** | **choose which Workspace to delete** | **OK**

Note! Be careful to select the right Workspace when deleting. The delete action cannot be undone. (However, no data is lost, because a workspace is only a representation of data.)

It is not possible to delete the default workspace.

2.6 List of shortcuts

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

Action	Windows/Linux	macOS
Adjust selection	Shift + arrow keys	Shift + arrow keys
Adjust workflow layout	Shift + Alt + L	⌘ + Shift + Alt + L
Back to Navigation Area	Alt + Home	⌘ + Home
	or Alt + fn + left arrow	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
Sort folder	Ctrl + Shift + R	⌘ + Shift + R
Split Horizontally	Ctrl + T	⌘ + T
Split Vertically	Ctrl + J	⌘ + J
Switch tabs in View Area	Ctrl + PageUp/PageDown	Ctrl + PageUp/PageDown
	or Ctrl + fn + arrow up/down	or Ctrl + fn + arrow up/down
Switch views	Ctrl + Shift + PageUp/arrow up	Ctrl + Shift + PageUp/arrow up
	Ctrl + Shift + PageDown/arrow down	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
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 . . . 63
3.1.1 Data structure . . . 64
3.1.2 Create new folders . . . 67
3.1.3 Sorting folders . . . 67
3.1.4 Multiselecting elements . . . 68
3.1.5 Moving and copying elements . . . 68
3.1.6 Change element names . . . 69
3.1.7 Delete, restore and remove elements . . . 70
3.1.8 Show folder elements in a table . . . 70
3.2 Working with tables . . . 72
3.2.1 Filtering tables . . . 73
3.3 Customized attributes on data locations . . . 76
3.3.1 Filling in values . . . 78
3.3.2 What happens when a clc object is copied to another data location? . . . 80
3.3.3 Searching . . . 80
3.4 Local search . . . 81
3.4.1 Quick search . . . 81
3.4.2 Advanced search . . . 85

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 through local data. The use of metadata tables in *CLC Genomics Workbench* is described separately, in chapter 10.

3.1 Navigation Area

The **Navigation Area** (see figure 3.1) is used for organizing and navigating data. Its behavior is similar to the way files and folders are usually displayed on your computer.

Just above the area with the listing of data are 4 icons. From left to right, these are:

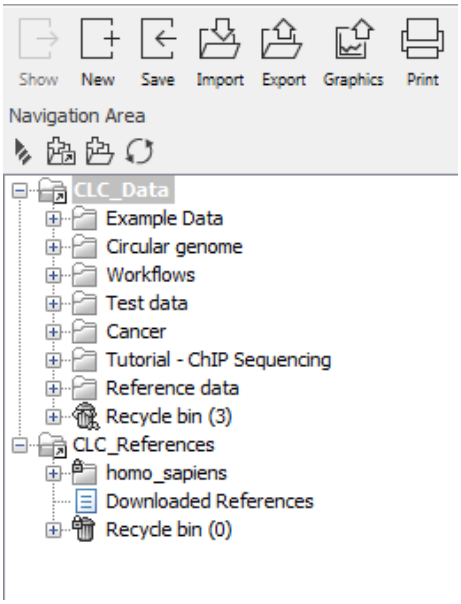


Figure 3.1: *The Navigation Area.*

- **Collapse all** ( ). This closes all the open folders in the Navigation Area.
- **Add File Location** ( ). This is explained in section 3.1.1.
- The Create Folder icon ( ), which is used to create new folders within a configured File Location.
- The Update All icon ( ), which refreshes the view of the Navigation Area.

To provide more space for viewing data, you can hide **Navigation Area** and the **Toolbox** by clicking on the hide icon ( ) in the top right hand side of the Navigation Area.

3.1.1 Data structure

The data in the **Navigation Area** is organized into a number of **Locations**. A Workbench data 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 the location points to.

This is explained visually in figure 3.2. The full path to the system folder can be seen by mousing over the data location folder icon as shown in figure 3.3.

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 (unless your computer administrator has configured the installation otherwise). There will also be a location called *CLC_References*.

The *CLC_References* location is intended for storing genomic references and associated data, downloaded using the Reference Data Manager, as described in section 8.

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 Workbench Navigation Area, as illustrated in figure 3.4.

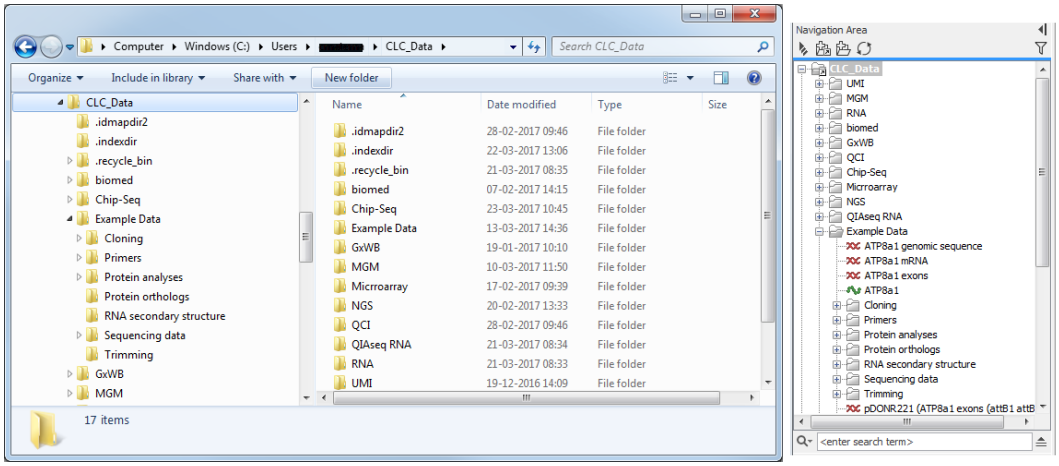


Figure 3.2: *In this example the location called "CLC_Data" points to the folder at C:\Users\<username>\CLC_Data.*

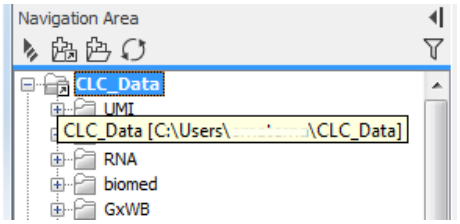


Figure 3.3: *Mousing over the location called 'CLC_Data' shows the full path to the system folder, which in this case is C:\Users\<username>\CLC_Data.*

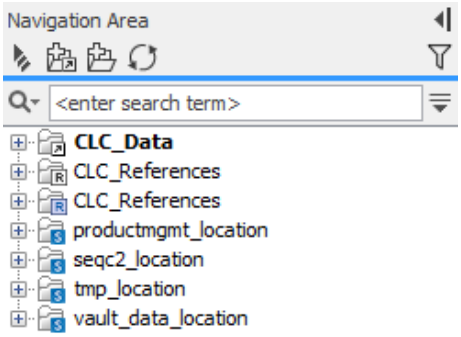


Figure 3.4: *Data areas on the CLC Server are highlighted with blue square icons in the Navigation Area.*

Adding locations

When a Workbench is first installed it will have one data area aleady configured and visible in the **Navigation Area** By default, this is a folder called 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

You can easily add more locations, which will then be visible in the **Navigation Area**. Go to:

File | New | Location ( )

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

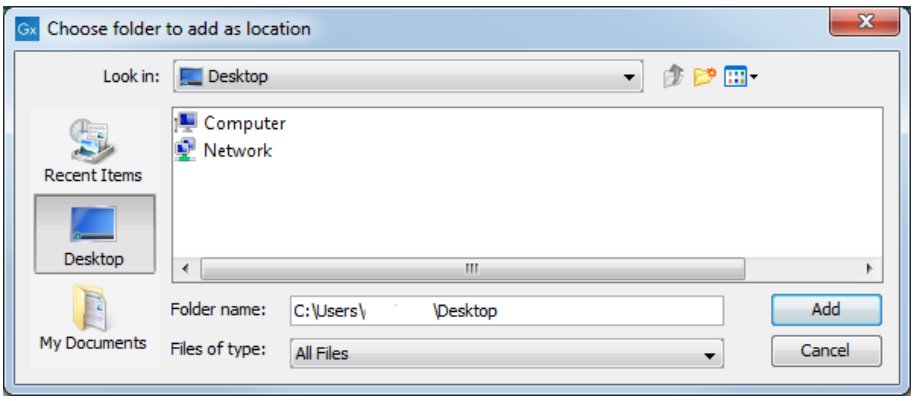


Figure 3.5: *Navigating to a folder to use as a new location.*

When you click **Open**, the new location is added to the **Navigation Area** as shown in figure 3.6.

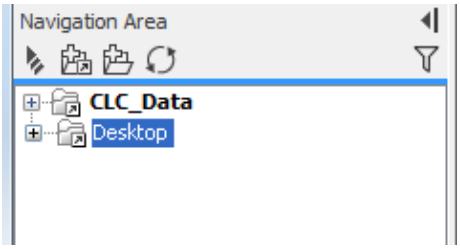


Figure 3.6: *The new location has been added.*

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 ( ).

You can configure any folder on a network drive or a removable drive as a location. The only restrictions are that you need permissions to access that folder, and it should **not** be a subfolder of an area already being used as a CLC Workbench or CLC Server location.

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. This refreshes the view of the Navigation Area, and all available locations will then be shown as active. There can be sometimes be a short delay before the interface update completes.

Sharing data is possible when a network drive is available to multiple Workbenches. In this case, you can add the same folder as a Data Location on each Workbench. However, it is important to note that data sharing is not actively supported: we do not support concurrent alteration of data and 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. In addition, 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 any Workbench that adds a Data Location already used by other Workbenches uses as a Data Location **the exact same folder from the network drive file system hierarchy as the other Workbenches have used**. Indicating a folder higher up or lower down in the hierarchy will

cause problems with the indexing of the files, meaning that newly created objects by Workbench A will not be found by Workbench B and vice versa.

Opening data

The elements in the Navigation Area are opened by:

Double-clicking on the element

or **Clicking once on the element** | **Show () in the Toolbar**

or **Right-click on the element** | **Show ()** or **Show (the one without an icon)** | **Select the desired way to view the element from the menu that appears when mousing over "Show"**

This will open a view in the View Area, which is described in section 2.1.

Adding data

Data can be added to the Navigation Area in a number of ways.

Files can be imported from the file system (see chapter 6).

Furthermore, an element can be added by dragging it into the Navigation Area. This could be views that are open, elements on lists, e.g. search hits or sequence lists, and files located on your computer.

Finally, you can add data by adding a new location (see section 3.1.1).

If a file or another element is dropped on a folder, it is placed at the bottom of the folder. If it is dropped on another element, it will be placed just below that element.

If the element already exists in the Navigation Area a copy will be created with the name extension "-1", "-2" etc. if more than one copy exist.

3.1.2 Create new folders

In order to organize your files, they can be placed in 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.

3.1.3 Sorting folders

You can sort the elements in a folder alphabetically:

Right-click the folder | **Sort Folder**

On Windows, subfolders will be placed at the top of the folder, and the rest of the elements will be listed below in alphabetical order. On Mac, both subfolders and other elements are listed together in alphabetical order.

3.1.4 Multiselecting elements

Multiselecting elements means that you select more than one element at the same time. This can be done in the following ways:

- 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 curser with the arrow-keys while holding down the <Shift> key, enables you to increase the number of elements selected.

3.1.5 Moving and copying elements

Elements can be moved and copied in several ways:

- Using **Copy** ( ) or **Cut** ( ) and **Paste** ( ) from the **Edit** menu or in the **Toolbar**.
- Using Ctrl + C (⌘ + C on Mac) or Ctrl + X (⌘ + X on Mac) and Ctrl + V (⌘ + V on Mac).
- Using drag and drop to move elements. Note that if you move data between locations, the original data is kept. This means that you are essentially doing a copy instead of a move operation.
- Using drag and drop while pressing Ctrl / Command to copy elements.

If the element already exists in the folder where it is moved or copied, the name of the copied/moved element will get an extension "-1", "-2", etc.

When you have cut an element, it is "grayed out" until you activate the paste function. If you change your mind, you can revert the cut command by copying another element.

Note that drag and drop allows you to:

- **Move** elements between different folders in the Navigation Area.
- **Open** the element in the View Area when dragged from the Navigation Area.
- **Save** an element when dragging its tab from the View Area to the desired location in the Navigation Area.

The use of drag and drop is supported throughout the program, also to open and re-arrange views (see section 2.1.5).

When copying an element or folder, it creates a link that can be pasted in a text editor (including email, skype conversation, etc.) This allows you to share the location of a particular element

with colleagues, provided that they have access to the same server. Your colleagues can, once their workbench is connected to the shared server, paste the link in the Search field above the Navigation Area and press Enter to find the element or folder in the shared location.

3.1.6 Change element names

This section describes two ways of changing the names of sequences in the **Navigation Area**. In the first part, the sequences themselves are not changed - it's their representation that changes. The second part describes how to change the name of the element.

Change how sequences are displayed Sequence elements can be displayed in the **Navigation Area** with different types of information:

- 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).

Whether sequences can be displayed with this information depends on their origin. Sequences that you have created yourself or imported might not include this information, and you will only be able to see them represented by their name. However, sequences downloaded from databases like GenBank will include this information.

To change how sequences are displayed:

right-click any element or folder in the Navigation Area | **Sequence Representation** | **select format**

This will only affect sequence elements, and the display of other types of elements, e.g. alignments, trees and external files, will be not be changed. If a sequence does not have this information, there will be no text next to the sequence icon.

Rename element Renaming a folder or an element in the **Navigation Area** can be done in two different ways:

select the element | **Edit in the Menu Bar** | **Rename**

or **select the element** | **F2**

When you can rename the element, you can see that the text is selected and you can move the cursor back and forth in the text. When the editing of the name has finished, press **Enter** or select another element in the **Navigation Area**. If you want to discard the changes instead, press the **Esc**-key.

For renaming annotations instead of folders or elements, see section 12.3.3.

3.1.7 Delete, restore and remove elements

When one deletes data held in a Workbench data location, it is moved to the recycle bin within that data location. Each data location has its own recycle bin. From the recycle bin, data can then be restored, or completely removed. Removal of data from the recycle bin frees disk space.

Deleting a folder or an element from a Workbench data location can be done in two ways, using the **Delete (** **)** option from the **Edit** menu, the right-click menu of an element, or in the **Toolbar**, or by just using the **Delete key** of your keyboard.

This will cause the element to be moved to the **Recycle Bin** ( ) where it is kept until the recycle bin is emptied or until you choose to restore the data object to your data location.

For deleting annotations instead of folders or elements, see section 12.3.4.

Items in a recycle bin can be restored in two ways: by dragging the elements with the mouse into the folder where they used to be, or by right-clicking the element and choosing the option **Restore**. Once restored, you can continue to work with that data.

All contents of the recycle bin can be removed by choosing to empty the recycle bin using the **Empty** command in the **Edit** menu or from the right-click menu on the **Recycle Bin** ( ). This deletes the data and frees up disk space.

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

Deleting data held on a CLC Server

You can delete data that you have "write" permission for from *CLC Server* data areas when logged into a server from your Workbench. The method of deleting data is the same as described above when deleting data held in Workbench data locations. The deleted data is placed in a **Recycle bin** ( ) on the *CLC Server*. The data in the server-based recycle bin can only be accessed by you and the server administrator. Note that the server administrator may have configured the recycle bin to be automatically emptied at regular intervals.

3.1.8 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.

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.

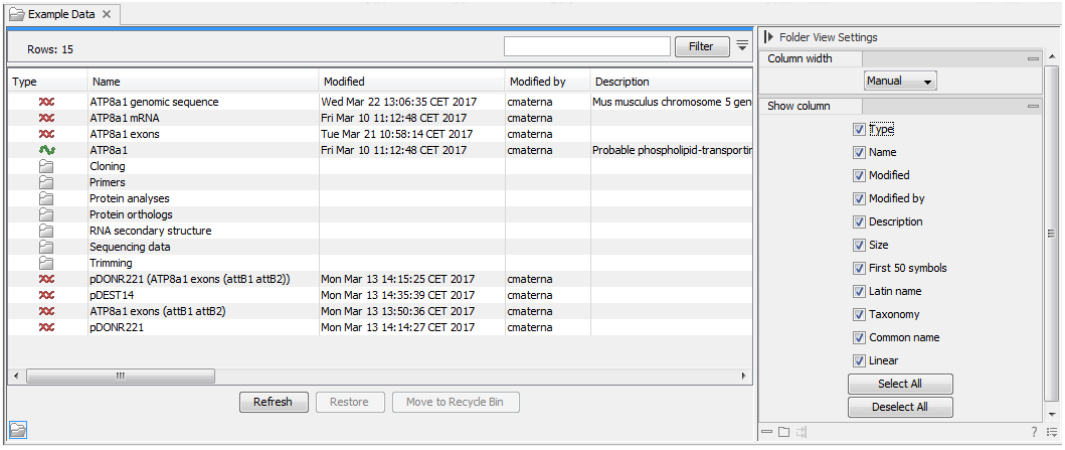


Figure 3.7: *Viewing the elements in a folder.*

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.

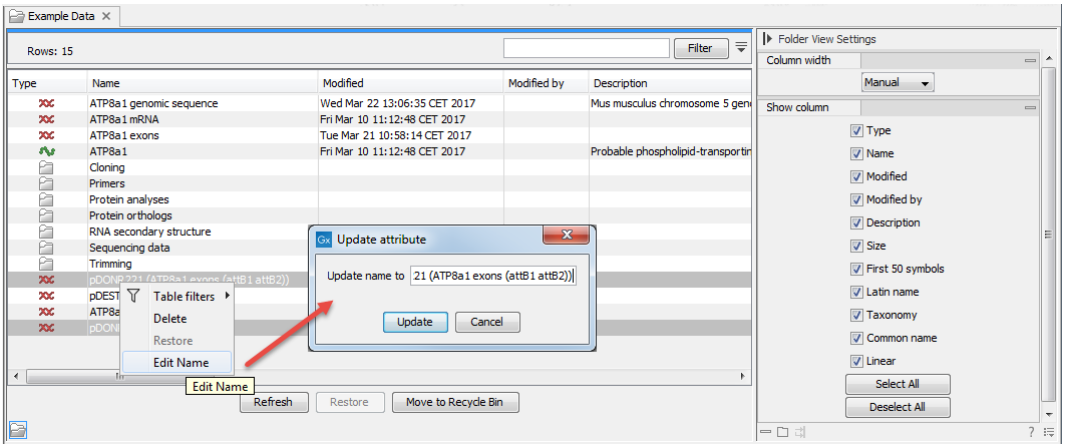


Figure 3.8: *Changing the common name of two sequences.*

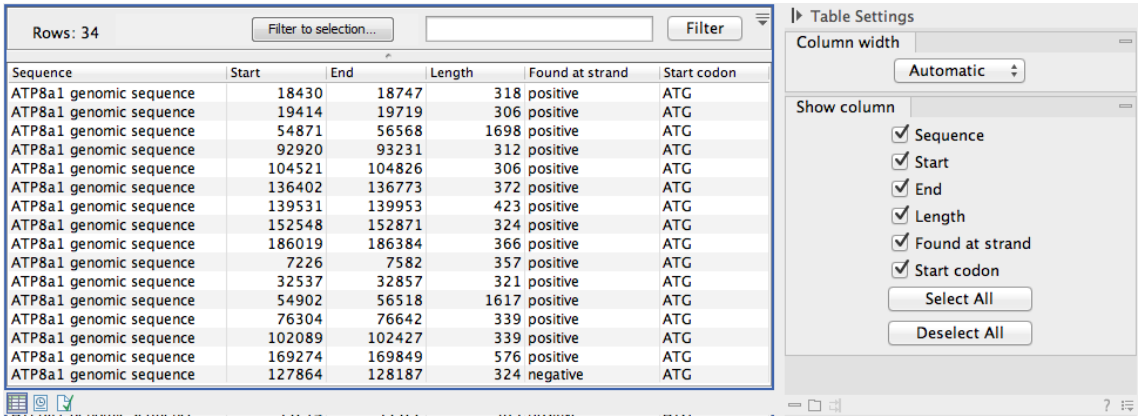
Note! This information is directly saved and you cannot undo.

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 tables

Tables are used in a lot of places in the *CLC Genomics Workbench*. There are some general features for all tables, irrespective of their contents, that are described here.

Figure 3.9 shows an example of a typical table. This is the table result of **Find Open Reading Frames** ( ). We use this table as an example to illustrate concepts relevant to all kinds of tables.



Sequence	Start	End	Length	Found at strand	Start codon
ATP8a1 genomic sequence	18430	18747	318	positive	ATG
ATP8a1 genomic sequence	19414	19719	306	positive	ATG
ATP8a1 genomic sequence	54871	56568	1698	positive	ATG
ATP8a1 genomic sequence	92920	93231	312	positive	ATG
ATP8a1 genomic sequence	104521	104826	306	positive	ATG
ATP8a1 genomic sequence	136402	136773	372	positive	ATG
ATP8a1 genomic sequence	139531	139953	423	positive	ATG
ATP8a1 genomic sequence	152548	152871	324	positive	ATG
ATP8a1 genomic sequence	186019	186384	366	positive	ATG
ATP8a1 genomic sequence	7226	7582	357	positive	ATG
ATP8a1 genomic sequence	32537	32857	321	positive	ATG
ATP8a1 genomic sequence	54902	56518	1617	positive	ATG
ATP8a1 genomic sequence	76304	76642	339	positive	ATG
ATP8a1 genomic sequence	102089	102427	339	positive	ATG
ATP8a1 genomic sequence	169274	169849	576	positive	ATG
ATP8a1 genomic sequence	127864	128187	324	negative	ATG

Figure 3.9: *A table showing the results of an open reading frames analysis.*

Table viewing options in the Side Panel Options relevant to the view of the table can be configured in the **Side Panel** on the right.

The Column width can be set to **Automatic** or **Manual**. By default, the first time you open a table, it will be set to **Automatic**. The default selected columns are hereby resized to fit the width of the viewing area. When changing to the **Manual** option, column widths will adjust to the actual header size, and each column size can subsequently be adjusted manually. When the table content exceeds the size of the viewing area, a horizontal scroll becomes available for navigation across the columns.

You can choose which columns can be displayed in the table by checking or unchecking the boxes beside column names in the Side Panel. In some tables, a single checkbox can be used to hide or show a whole set of columns belonging to a certain category. Two buttons called **Select all** and **Deselect all** allow you to select or deselect all columns from that Side Panel section in one click.

Finally, in some table types (such as Expression Browsers), the content of some columns can be modified using settings in the Side Panel (such as Expression values and Grouping in figure 3.10).

Working with tables using the right-click menu

Right-clicking on a table reveals standard menu options, as well as table-specific options, found under these submenus:

- **Table filters** Lists advanced filters relevant to the column that you right-click upon. See section 3.2.1 for details.
- **File** Includes the option **Export Table**, which allows the table to be exported in .csv or Excel

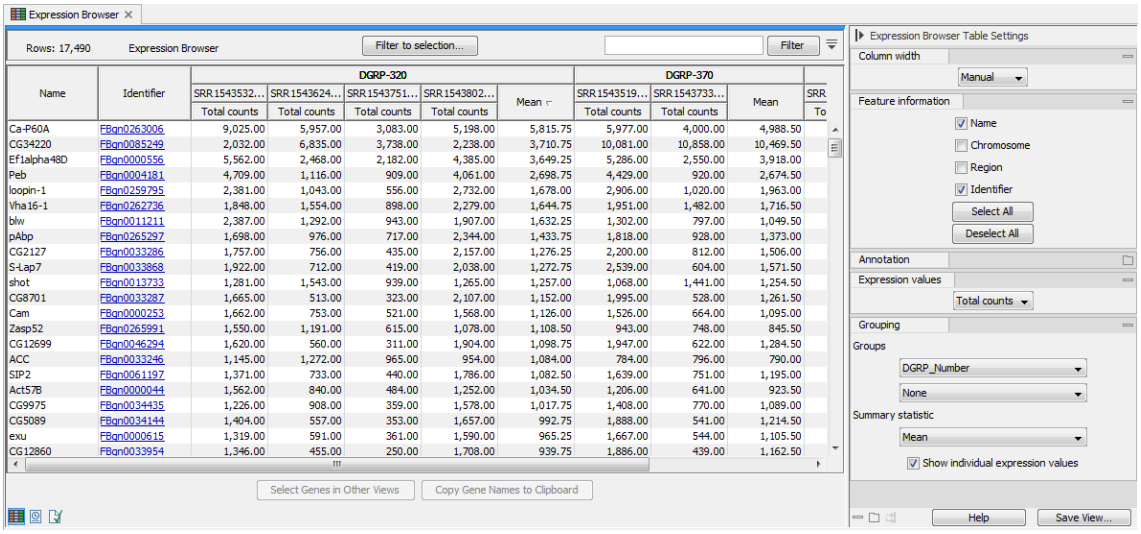


Figure 3.10: *A table showing the results of an RNA-seq analysis.*

format. This option respects the filtering, sorting and column selections that have been applied to the table.

- **Edit** Includes the option **Copy Cell**, which copies the contents of individual cells to the clipboard.

Sorting tables You can **sort** table 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.

Adjusting the column order The column order for viewing and exporting purposes can be changed. To move a column, click on its heading and, keeping the mouse button depressed, drag it to the desired location in the table. Files exported from a table at this point, such as .csv files, will reflect the new column order.

It is not possible to save the revised column order in the Workbench.

3.2.1 Filtering tables

Filters can be set using the functionalities located at the top of any table in the Workbench: a Filter to Selection button, a simple filter mode and an advanced filter mode. A counter in the upper left corner tells you the number of rows that passed the filter.

Filter to selection A button called **Filter to selection** allows for reducing the size of a table to a few pre-selected rows. The option **Filter to selected rows** will keep in the table view only the rows that are selected, whether they were selected manually, or by using the function "Select in other views" available for some tables (for example when the table is associated with a graphical view such as a Venn diagram, or a volcano plot). Restore the complete table by choosing the option **Clear selection filter**.

Simple filter The simple mode is the default and is applied simply by typing text or numbers (see an example in figure 3.11).

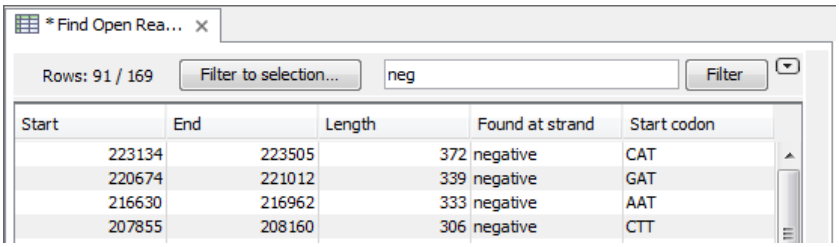


Figure 3.11: *Typing "neg" in the filter in simple mode.*

Typing "neg" in the filter will only show the rows where "neg" is part of the text in any of the columns. The text does not have to be in the beginning, thus "ega" would give the same result. This simple filter works fine for fast, textual and non-complicated filtering and searching. Filtering is automatic once you start typing, unless you are working with a table with more than 10000 rows, in which case you have to actually click the **Filter** button for the filtering to take effect.

The following characters have special meanings when used in simple filtering of tables in the workbench:

- **Space** (separates search items unless inside quotes)
- **Backslash** (escapes characters, in particular those mentioned in this list)
- **Single and double quotes ' and "** (define entire phrases to search for)
- **Minus -** (specifies words or phrases to exclude)
- **Colon :** (searches in a specific table column)

These characters cannot be used in the Advanced filter described below, because the Advanced Filter functionality makes it easy to include/exclude specific terms or limit searches to a particular column by providing the appropriate fields.

Also not that typing `cat dog` in the Simple filter field will return all rows with cat and dog in them, in any order. But the same search term put in the Advanced filter field (visible once you click on the little arrow to the right of the simple filter field), will only return rows with the exact phrase "cat dog".

Advanced filter In the advanced mode, you can make use of numerical information or make more complex filter combinations using more than one criterion in the filter. Click the **Advanced filter** ( ) button to open the first criterion of the advanced filter. Criteria can be added or removed by clicking the **Add** ( ) or **Remove** ( ) buttons. At the top, you can choose whether all the criteria should be fulfilled (**Match all**), or if just one of the needs to be fulfilled (**Match any**).

For each filter criterion, you first have to select which **column** it should apply to.

Next, you choose an **operator**. For numbers, you can choose between:

- **=** (equal to)
- **<** (smaller than)

- **>** (greater than)
- **<>** (not equal to)
- **abs. value <** (absolute value smaller than. This is useful if it doesn't matter whether the number is negative or positive)
- **abs. value >** (absolute value greater than. This is useful if it doesn't matter whether the number is negative or positive)

Note, that the number of digits displayed is a formatting option which can be set in the View Preferences. The true number may well be (slightly) larger. This behaviour can lead to problems when filtering on exact matches using the **=** (equal to) operator on numbers. Instead, users are advised to use two filters of inequalities (**<** (smaller than) and **>** (greater than)) delimiting a (small) interval around the target value.

For text-based columns, you can choose between:

- **starts with** (the text starts with your search term)
- **contains** (the text does not have to be in the beginning)
- **doesn't contain**
- **=** (the whole text in the table cell has to match, also lower/upper case)
- $\neq$ (the text in the table cell has to not match)
- **is in list** (The text in the table cell has to match one of the items of the list. Items are separated by comma, semicolon, or space. This filter is not case-sensitive.)
- **is not in list** (The text in the table cell must not match any of the items of the list. Items are separated by comma, semicolon, or space. This filter is not case-sensitive)

Once you have chosen an operator, you can enter the **text or numerical value** to use.

The advanced filter criterion mentioned above are also available from a menu that appears by right-clicking on a value in a table: just specify the operator, and the column and value where you right-clicked for the menu to appear will define the two other fields of the advanced filter.

If you wish to reset the filter, simply remove () all the search criteria. Note that the last one will not disappear - it will be reset and allow you to start over.

Figure 3.12 shows an example of an advanced filter which displays the open reading frames larger than 400 that are placed on the negative strand.

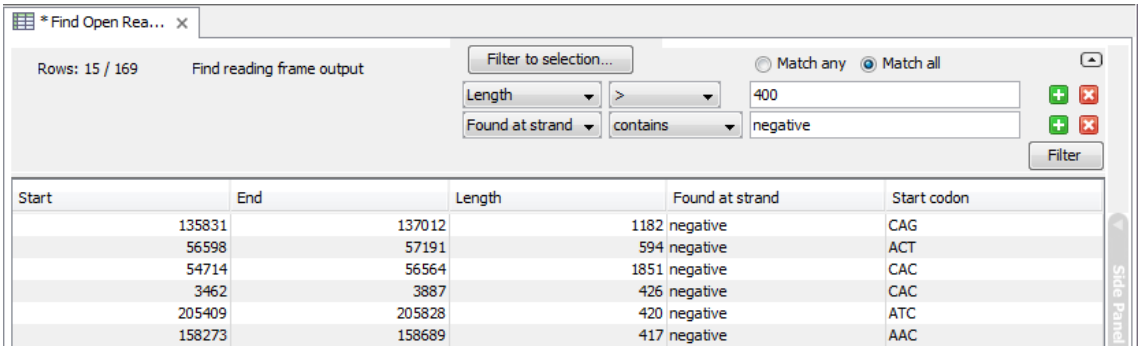


Figure 3.12: *The advanced filter showing open reading frames larger than 400 that are placed on the negative strand.*

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 available[1] go to the Workbench:

right-click the data location | Location | Attribute Manager

This will display the dialog shown in figure 3.13.

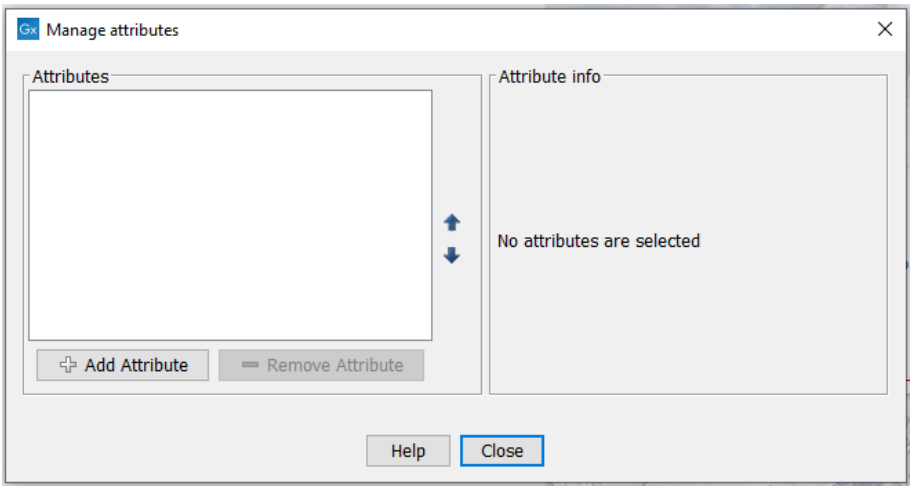


Figure 3.13: *Adding attributes.*

Click the **Add Attribute** ( ) button to create a new attribute. This will display the dialog shown in figure 3.14.

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

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

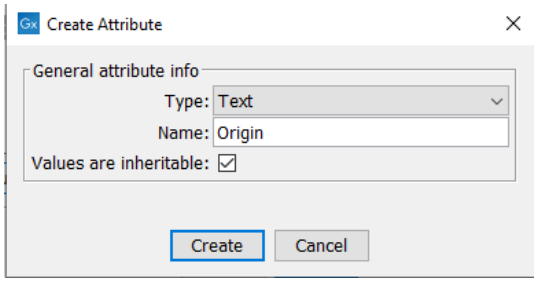


Figure 3.14: *The list of attribute types.*

and yes/no).

- **Text**. For simple text with no constraints on what can be entered.
- **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 3.15).

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.

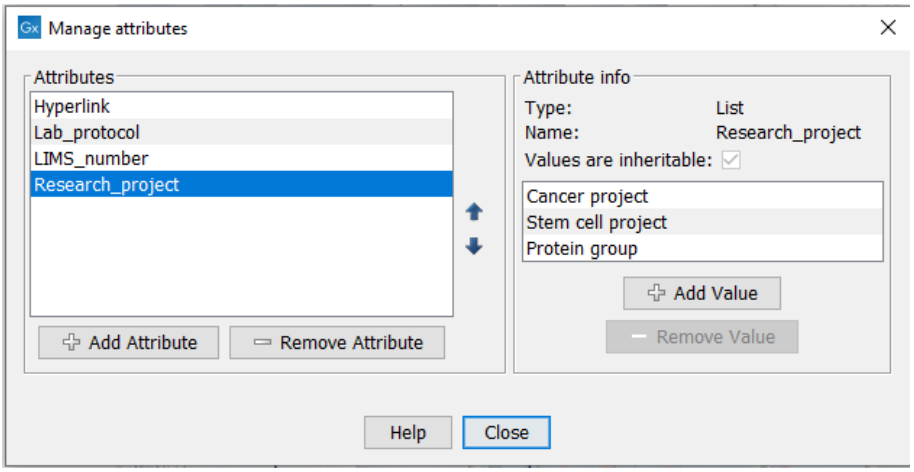


Figure 3.15: *Defining items in a list.*

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 3.16), the end users can start filling in information.

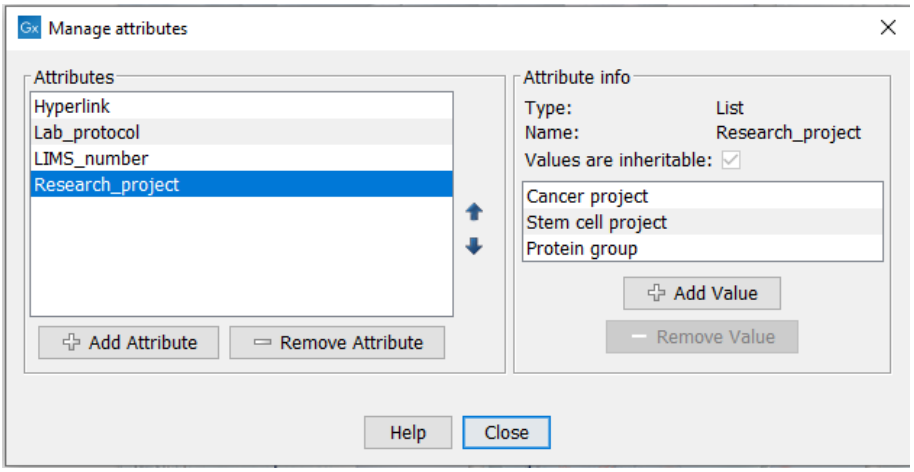


Figure 3.16: *A set of attributes defined in the attribute manager.*

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 3.17.

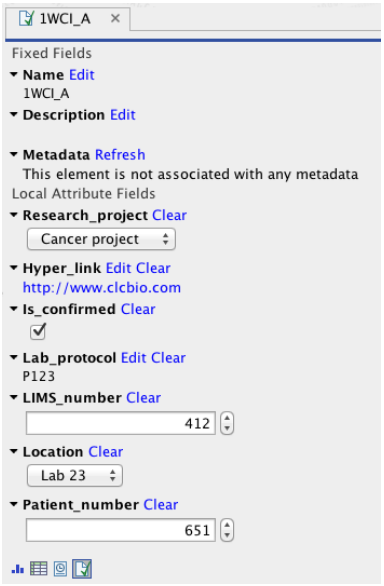


Figure 3.17: *Adding values to the attributes.*

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 3.18).

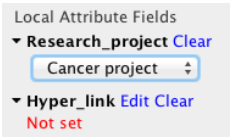


Figure 3.18: *An attribute which has not been set.*

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 3.18, you will *not* be able to find 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 (see section 3.1.8).

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

When an attribute has been created, it will automatically be available for searching. This means that in the **Local Search** (), you can select the attribute in the list of search criteria (see figure 3.19).

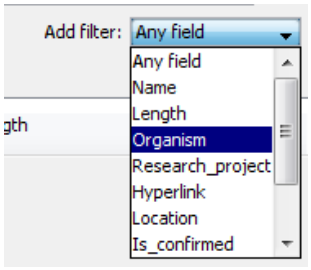


Figure 3.19: *The attributes from figure 3.16 are now listed in the search filter.*

It will also be available in the **Quick Search** below the **Navigation Area** (press Shift+F1 (Fn+Shift+F1 on Mac) and it will be listed - see figure 3.20).

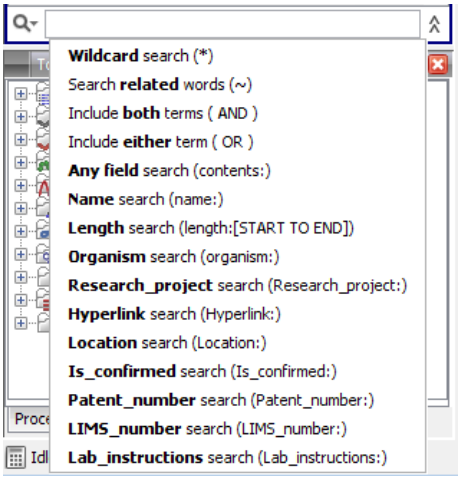


Figure 3.20: *The attributes from figure 3.16 are now available in the Quick Search as well.*

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

3.4 Local search

There are two ways of searching for data based on its name or other attributes, or for metadata that contains particular information:

- **Quick search**, using the search box above the **Navigation Area**, described in section 3.4.1
- **Advanced search** using the tool under the Edit menu called **Local Search**, described in section 3.4.2

Search index The *CLC Genomics Workbench* automatically maintains an index of all data in all locations in the **Navigation Area**. It is this index that is used when searching for data.

Problems with search results can reflect a problem with the index. If you suspect the index is out of sync with the data, you can rebuild it by right-clicking on the relevant data location in the Navigation Area and then selecting:

Location | **Rebuild Index**

Rebuilding the index can take some time.

If you wish to stop the index building process, this can be done in the process area, see section 2.4.

3.4.1 Quick search

Using the search box just above the Navigation Area on the left side of the *CLC Genomics Workbench*, shown in figure 3.21, you can search for

- Data elements with names or other text-based attributes that match the search term entered.
- Metadata tables containing information matching the search term.
- A data element based on its CLC URL.

When multiple terms are entered, they are searched for individually, equivalent to putting OR between the terms. See the **Advanced search expressions** section below for information on creating more specific queries.

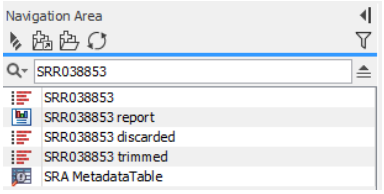


Figure 3.21: *The quick search functionality can be used to look for data elements and for information contained within metadata tables. Here, the search term "SRR1543627" has returned data elements with names or other attributes matching that term, as well as metadata tables that contain that term.*

The following list of characters have special meanings when searching:

: + - && || ! ( ) ^ [ ] “~* ? : \ /.

To search using these characters themselves, put quotes around the search expression. Further details are provided in **Advanced search expressions** section below.

Search results

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

The number of hits to be listed initially can be configured in the Workbench Preferences, as described in section 4).

If no hits are found, you will be asked if you wish to search for matches that start with your search term. If you accept this, a wild card (*) will be appended to the search term.

Keeping the Alt key depressed when you click on a search result will move the focus to that data element in the **Navigation Area**.

Quick search history

You can access the 10 most recent searches by clicking the icon () next to the search field (see figure 3.22).

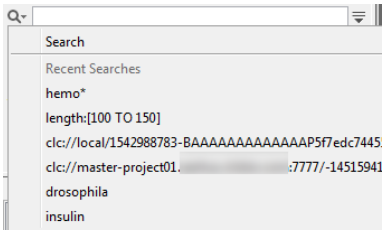


Figure 3.22: *Recent searches.*

Clicking one of the recent searches runs the search again.

Advanced search expressions

Press **Shift+F1** when entering a search term in the quick search box to bring up terms that can be used in search expressions, as shown in figure 3.23.

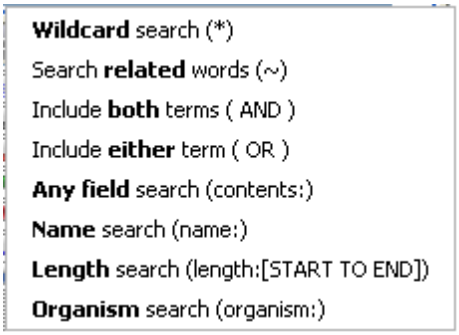


Figure 3.23: *A list of available terms that can be used when creating advanced search expressions pops up if you press Shift+F1 after clicking in the quick search box.*

Click on any entry in the list to include it in your search. For example, to search for sequences named BRCA1, select "Name search (name:)", and then type "BRCA1" to get the search expression: "name:BRCA1". The full list presented depends on the attributes available. If you have added attributes (see section 3.3), these will also appear on the list when pressing **Shift+F1**.

An explanation of some available options:

- **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.
- **Include both terms (AND)**. Putting AND between each of two or more search terms means that all these terms must be present for the element to be returned. E.g. A search for `brca1 AND human` will find sequences where *both* terms are present. `&&` is equivalent to AND, e.g. `brca1 && human`
- **Include either term (OR)**. Putting OR between each of two or more search terms means that at least one of the search terms must be present for the element to be returned. E.g. a search for `brca1 OR brca2` will find sequences where *either* of these terms is present. `||` is equivalent to OR, e.g. `brca1 || human`
- **Do not include term (NOT)** If you write a term after not, then elements with these terms will not be returned.
- **Name search (name:)**. Search only for elements with names that match the search term. E.g. `name:ATPA1`
- **Length search (length:[START TO END])**. Search for sequences of a specific length. E.g. searching for `length:[1000 TO 2000]` would return sequences elements with lengths between 1000 and 2000 residues. The square brackets are vital for this search to return the expected results.
- **Organism search (organism:)**. For sequences, you can specify the organism to search for. This will look in the "Latin name" field which is seen in the **Sequence Info** view (see section 12.4).

Note: Wildcards cannot be used in conjunction with searches where the type of the data to search for, or in the case of metadata table contents, the columns to search within, have been specified. So, for example the search terms `BRCA*` and `name:BRCA1` are fine, but `name:BRCA*` is not a valid search term.

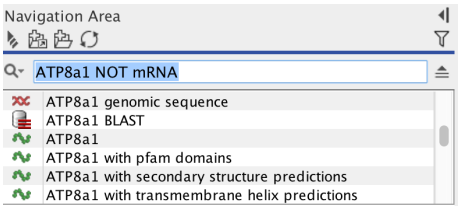


Figure 3.24: *Elements with information that includes the term "ATP8" but not the term "mRNA" are returned using the search shown.*

Searching for information in metadata tables

Searches for metadata table content can be made more specific by including a column name before the value of interest. For example, if a metadata table contains a column named "Family" containing rows with the value "Smith", you can search for `Family:Smith`. Forgoing the column name, i.e. searching with just the term `Smith`, will also find the relevant metadata tables, but will also find any other data elements that have information matching that search criteria.

The creation and use of metadata tables in *CLC Genomics Workbench* are described in chapter 10.

Of note when including metadata table column names in searches:

- If you import an Excel file as metadata, or are working with workflow result metadata tables, all columns in these are of type **Text** by default. Thus most searches will involve using syntax appropriate for searching text fields. This is true even if the contents of a text-type column look like numbers or dates.
- Attribute names take precedence over column names. So, for example, if you have a metadata table called "Name", searching with Name:Smith will search for data elements with the name Smith, rather than for metadata tables with the entry "Smith" in columns called Name. Common reserved attribute names include:
  - ID
  - Name
  - Length
  - Organism
  - Contents
  - Path
  - Type
  - Subtype

To avoid running into such problems, we would generally recommend searching with just the term you wish to find, rather than specifying a column name.

Searches can be done on the contents of metadata columns of types other than Text, as described briefly below. How to specify metadata column types is described in section 10.5.

- **Whole number column type** Specific numbers or ranges can be searched for. Here, the column name should be specified. E.g. `Age:17` or `Age:[17 TO 20]`.

- **Date or Date and time column types** Specific dates, or dates and times, or ranges of these can be searched for. In both cases, you can choose whether to prepend a column name. E.g. a date search might take the form `2019-03-25` or `Collection\ time:2019-03-25`. Times are specified after the date. Examples of the forms recognized for dates and times, with or without a column name being prepended, are:

  - `"2019-11-01 17:30"` The quotes are necessary due to the special characters (space and colon).
  - `2019-11-01T1730` A T separates the date from the time, and no colon is included in the time.
  - `2019-11-01\ 17\:30` A backslash is included before the special characters (space and colon).

  Date ranges, or date and time ranges are specified by including the start and end points within square brackets, with "TO" in between. E.g.

  - `["2019-11-01 17:30" TO "2019-11-31 12:00"]` The quote are necessary due to the space character.
  - `[2019-11-01T17:30 TO 2019-11-31T12:00]` A T has replaced the space character between the dates and times.
  - `[2019-11-01T1730 TO 2019-11-31T1200]` Another representation that does not need to be enclosed in quotes.

- **Yes / No column type** Searches for values in this column type (boolean values) must include the column name followed by the value "True" or "False". These terms are not case sensitive, so "true" and "TRUE" are also fine. e.g. `Infected:true`.

Searching using CLC URLs

A particular data element in the Navigation Area can be quickly found by entering its CLC URL into the quick search box. One example of when this can be useful is when working using a CLC Server and sharing the location of particular data elements with another user of that server.

A simple example is shown in figure 3.25.

To obtain a CLC URL for a particular data element, right click on the element name and choose the option Copy. When you then paste that information, for example in the search box or in a file or email, it is the CLC URL that is recorded.

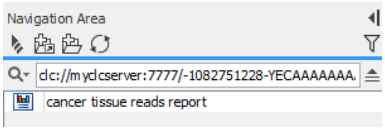


Figure 3.25: *Data elements can be located using a CLC URL.*

3.4.2 Advanced search

As a supplement to the **Quick search** described in the previous section you can use the more advanced search:

Edit | **Local Search (** **)**

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

The first thing you can choose is which location should be searched. All the active locations are shown in this list. You can also choose to search all locations. Read more about locations in section 3.1.1.

Furthermore, you can specify what kind of elements should be searched:

- All sequences
- Nucleotide sequences
- Protein sequences
- All data, which will also search for values contained in metadata tables.

When searching for sequences, you will also get alignments, sequence lists etc as result, if they contain a sequence which match the search criteria.

Below are the search criteria. First, select a relevant search filter in the **Add filter:** list. For sequences you can search for

- Name
- Length
- Organism

See section 3.4.1 for more information on individual search terms.

For all other data, you can only search for name.

If you use **Any field**, it will search all of the above plus the following:

- Description
- Keywords
- Common name
- Taxonomy name

To see this information for a sequence, switch to the **Element Info** ( ) view (see section 12.4).

For each search line, you can choose if you want the exact term by selecting "is equal to" or if you only enter the start of the term you wish to find (select "begins with").

An example is shown in figure 3.26.

This example will find human nucleotide sequences (organism is *Homo sapiens*), and it will only find sequences shorter than 10,000 nucleotides.

Note that a search can be saved ( ) for later use. You do not save the search results - only the search parameters. This means that you can easily conduct the same search later on when your data has changed.

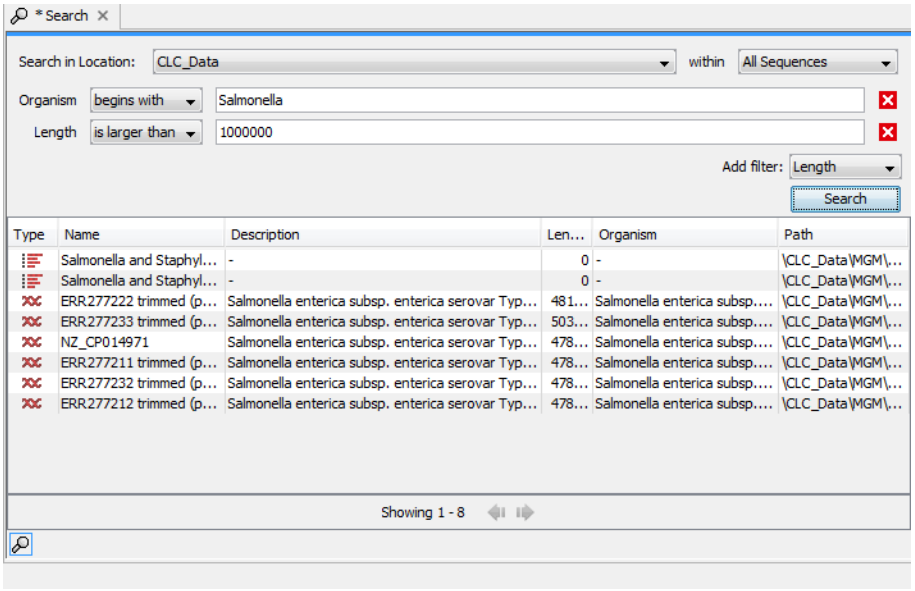


Figure 3.26: *Searching for Salmonella sequences larger than 1 million nucleotides.*

Chapter 4

User preferences and settings

Contents

4.1 General preferences 89
4.2 View preferences 90
4.2.1 Import and export Side Panel settings 92
4.3 Data preferences 93
4.4 Advanced preferences 93
4.5 Export/import of preferences 95
4.6 View settings for the Side Panel 96

The first three sections in this chapter deal with the general preferences that can be set for *CLC Genomics Workbench* using the **Preferences** dialog. The next section explains how the settings in the **Side Panel** can be saved and applied to other views. Finally, you can learn how to import and export the preferences.

The **Preferences** dialog offers opportunities for changing the default settings for different features of the program.

The **Preferences** dialog is opened in one of the following ways and can be seen in figure 4.1:

Edit | **Preferences (** **)**

or **Ctrl + K (⌘ + ; on Mac)**

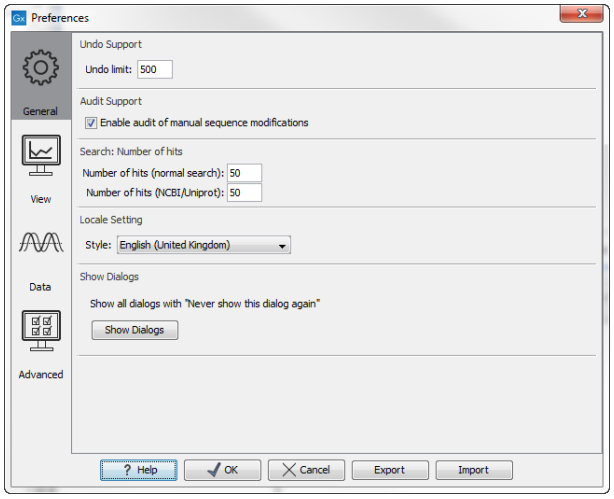


Figure 4.1: *Preferences include General preferences, View preferences, Data preferences, and Advanced settings.*

4.1 General preferences

The **General preferences** include:

- **Undo Limit**. As default the undo limit is set to 500. By writing a higher number in this field, more actions can be undone. Undo applies to all changes made on molecules, sequences, alignments or trees (see section 2.1.5).
- **Audit Support**. If this option is checked, all manual editing of sequences will be marked with an annotation on the sequence (see 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 also recorded in the History log ( ) (see section 2.1.1).



Figure 4.2: *Annotations added when the sequence is edited.*

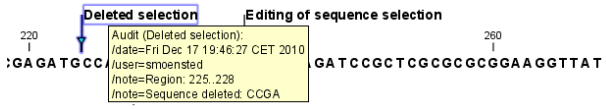


Figure 4.3: *Details of the editing.*

- **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 punctation is used in numbers all over the program.
- **Show Dialogs**. A lot of information dialogs have a checkbox: "Never show this dialog again". When you see a dialog and check this box in the dialog, the dialog will not be shown again.

If you regret and wish to have the dialog displayed again, click the button in the General Preferences: **Show Dialogs**. Then all the dialogs will be shown again.

- **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 six groups of default **View** settings:

1. **Toolbar** lets you choose the size of the toolbar icons, and whether to display names below the icons (figure 4.4).

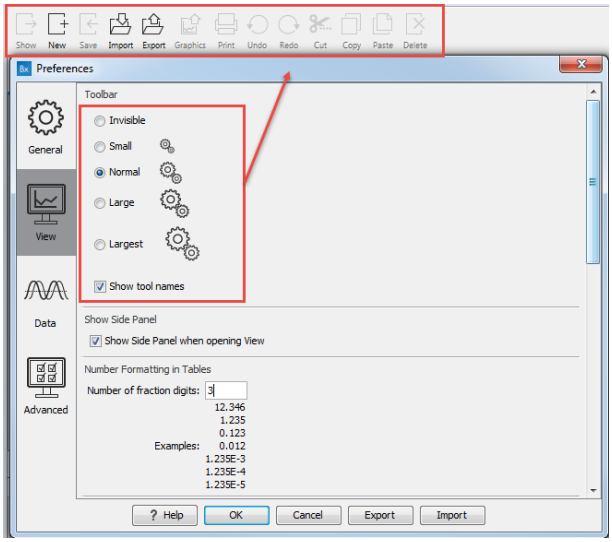


Figure 4.4: *Number formatting of tables.*

2. **Show Side Panel** allows you to choose whether to display the side panel when opening a new view. Note that 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** specifies how the numbers should be formatted in tables (see figure 4.5). The examples below the text field are updated when you change the value so that you can see the effect. After you have changed the preference, you have to re-open your tables to see the effect.

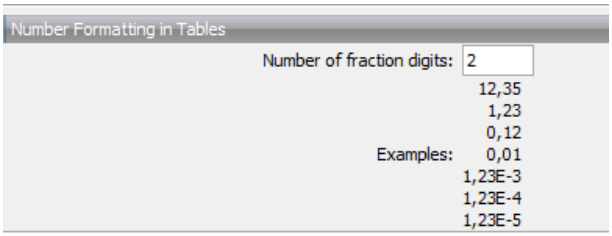


Figure 4.5: *Number formatting of tables.*

4. **Sequence Representation** allows you to change the way the elements appear in the Navigation Area. The following text can be used to describe the element:

   - 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).

5. **User Defined View Settings** gives you an overview of the different Side Panel settings that are saved for each view. See section 4.6 to learn more about how to create and save style sheets. If there are other settings beside CLC Standard Settings, you can use this overview to choose which of the settings should be used per default when you open a view (see an example in figure 4.6).

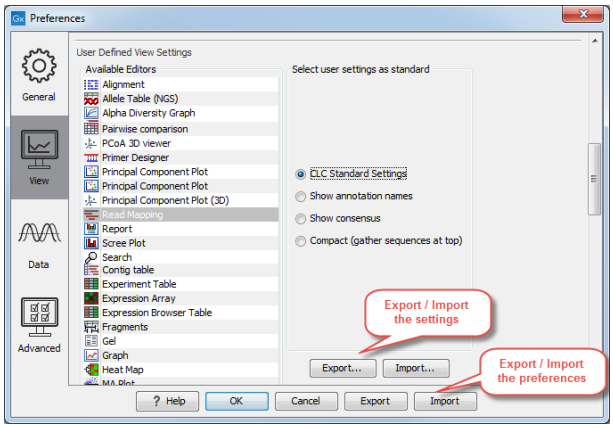


Figure 4.6: *Selecting the default view setting.*

Note that the content of this list depends on the nature of the elements that are saved in the Navigation Area. When the list grows, you may have to scroll up or down to find the relevant settings.

6. **Molecule Project 3D Editor** gives you the option to turn off the modern OpenGL rendering for **Molecule Projects** (see section 14.2).

4.2.1 Import and export Side Panel settings

If you have created a special set of settings in the **Side Panel** that you wish to share with other CLC users, you can export the settings in a file. The other user can then import the settings.

To export the **Side Panel** settings, first select the views that you wish to export settings for. Use Ctrl+click (⌘ + click on Mac) or Shift+click to select multiple views. Next click the **Export...** button that is situated below the list of possible settings (see figure 4.6), and not the Export button at the very bottom of the dialog, as this one will export the **Preferences** (see section 4.5).

A dialog will be shown (see figure 4.7) that allows you to select which of the settings you wish to export.

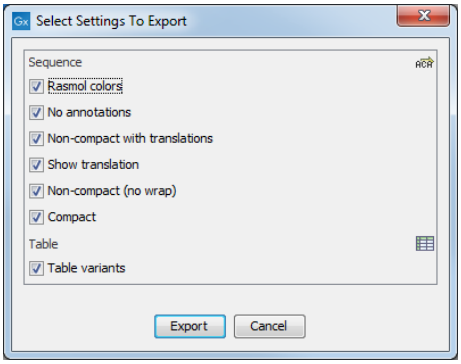


Figure 4.7: *Exporting all settings for circular views.*

When multiple views are selected for export, all the view settings for the views will be shown in the dialog. Click **Export** and you will now be able to define a save folder and name for the exported file. The settings are saved in a file with a .vsf extension (View Settings File).

Similarly, to import a **Side Panel** settings file, make sure you are at the bottom of the **View** panel of the **Preferences dialog**, and click the **Import...** button. Note that there is also another import button at the very bottom of the dialog, but this will import the other settings of the **Preferences** dialog (see section 4.5).

Select the *.vsf file where the settings are saved. The following dialog asks if you wish to overwrite existing **Side Panel** settings, or if you wish to merge the imported settings into the existing ones (see figure 4.8).

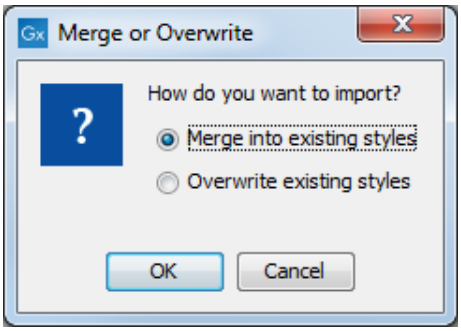


Figure 4.8: *When you import settings, you are asked if you wish to overwrite existing settings or if you wish to merge the new settings into the old ones.*

WARNING! If you choose to overwrite the existing settings, you will loose ALL the Side Panel settings that were previously saved.

To avoid confusion of the different import and export options, here is an overview:

- Import and export of **bioinformatics data** such as sequences, alignments etc. (described in section 6.1).
- **Graphics** export of the views which creates image files in various formats (described in section 6.7).
- Import and export of **Side Panel Settings** as described above.
- Import and export of all the **Preferences** except the Side Panel settings. This is described in the previous section.

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 20.4.1).
- Linkers for importing 454 data (see section 35.3).

4.4 Advanced preferences

Proxy Settings The Advanced settings include the possibility to set up a proxy server. This is described in section 1.6.

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 3.1.1). 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.

Internal compression of CLC data was introduced in CLC Genomics Workbench 12.0, CLC Main Workbench 8.1 and CLC Genomics Server 11.0. CLC format files with internal compression are not compatible with older versions of the software. 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.

To work with specific data sets in older CLC software versions, we recommend exporting the data to CLC or zip format and turning on the export option "Maximize compatibility with older CLC products". This is described in more detail in section 6.6.4.

NCBI Integration Without an API key, access to NCBI from asingle 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.

SRA Download The following options are available:

- **Use Aspera when available** Per default, Aspera is automatically used if installed. This option makes it possible to disable Aspera.
- **Limit Aspera download speeds to [ ] Mb/s (Mac and Linux only)** Using Aspera may take up a lot of network resources. Use this option to specify a maximum download speed (in megabit per second). Note that this option is only available on Mac and Linux. For Windows users, it is possible to limit the maximum download speed by modifying the `aspera.conf file`, which can be found in `C:\Program Files (x86)\Aspera\Aspera Connect`. See `http://download.asperasoft.com/download/docs/csrv/3.3.4/linux/html/index.html` and `http://download.asperasoft.com/download/docs/csrv/3.3.4/linux/html/fasp/setting-global-bandwidth.html` for more details.

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: This setting is relevant when the "On Server" option is chosen in the Reference Data Manager - as it is by default (figure 4.9): data will be downloaded directly to CLC Genomics Server.

However, if CLC Genomics Server has no access to the external network, but the Workbench does, the "CLC Workbench" option can be used. In this case, data is downloaded via the Workbench and then moved to the reference data area on the Server. When the "CLC Workbench" option is selected, the Workbench must be left running throughout the data download process.

CLC Server Login It is possible to Save the login username and password, initiate automatic server login when opening the workbench, and bypass a proxy server.

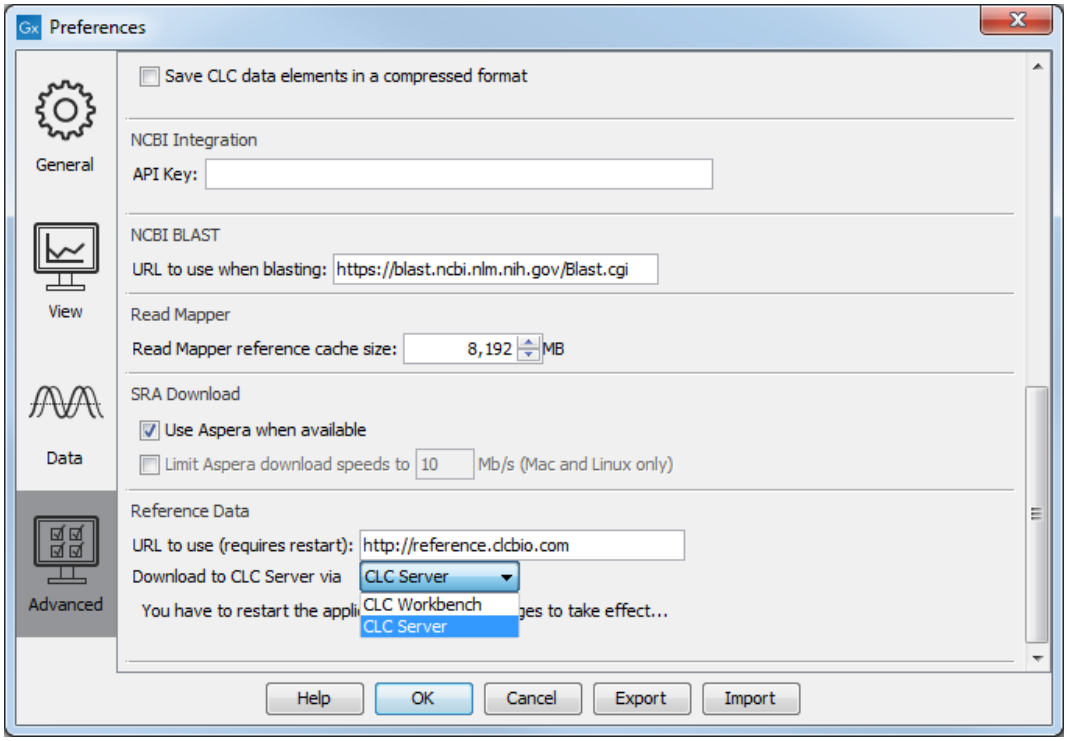


Figure 4.9: *By default, data downloaded to a CLC Genomics Server using the Reference Data Manager will be downloaded directly to the server, without going via the Workbench.*

4.5 Export/import of preferences

The user preferences of the *CLC Genomics Workbench* can be exported to other users of the program, allowing other users to display data with the same preferences as yours. You can also use the export/import preferences function to backup your preferences.

To export preferences, open the **Preferences** dialog and click on the Export bottom at the bottom of the Preferences dialog. Select the relevant preferences and click Export to choose a location to save the exported file(see figure 4.10).

Note! The format of exported preferences is *.cpf. This notation must be submitted to the name of the exported file in order for the exported file to work.

Before exporting, you are asked about which of the different settings you want to include in the exported file. One of the items in the list is "User Defined View Settings". If you export this, only the information about which of the settings is the default setting for each view is exported. If you wish to export the **Side Panel Settings** themselves, see section 4.2.1.

The process of importing preferences is similar to exporting: click the Import button and browse to the *.cpf file.

To avoid confusion of the different import and export options, you can find an overview here:

- Import and export of **bioinformatics data** such as molecules, sequences, alignments etc. (described in section 6.1).
- **Graphics** export of the views that create image files in various formats (described in section 6.7).

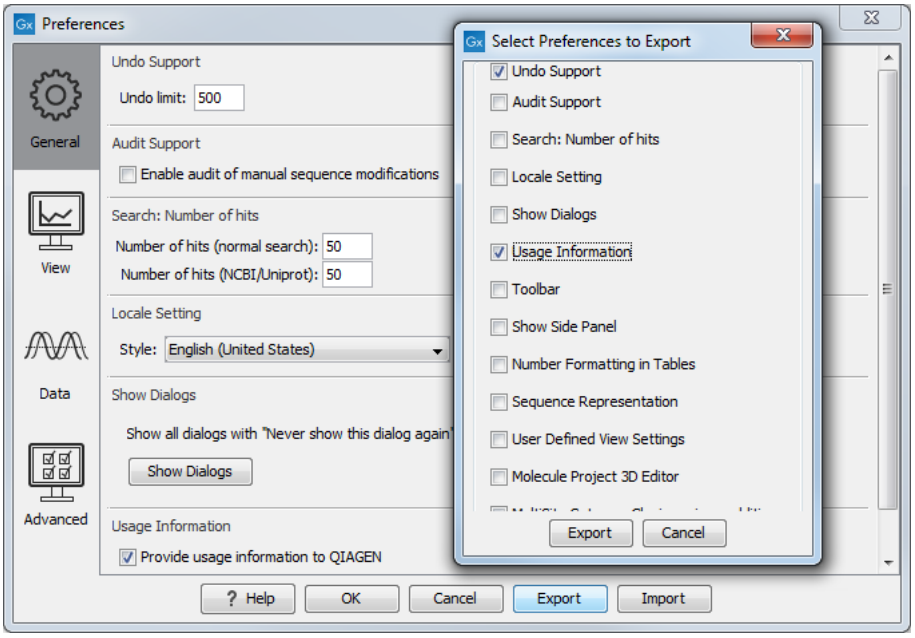


Figure 4.10: *Select which of the preferences you want to export.*

- Import and export of **Side Panel Settings** as described in the next section.
- Import and export of all the **Preferences** except the Side Panel settings. This is described above.

4.6 View settings for the Side Panel

The **Side Panel** is shown to the right of all views that are opened in the View Area. Settings are specific to the type of view. Hence, when you save settings of a circular view, they will not be available if you open the sequence in a linear view (see section 2.1.8).

The options for saving and applying are available at the bottom of the **Side Panel** (see figure 4.11).

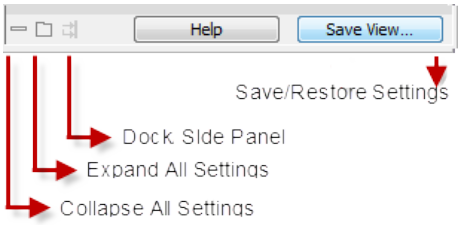


Figure 4.11: *Functionalities found at the bottom of the Side Panel.*

Opening a view type (e.g., a circular sequence, a variant table, or a PCA) for the first time will display the element using the CLC Standard Settings for that type of view. You can then adjust the settings using all the options available to you in the side panel. When you have adjusted a view to your preference, the new settings can be saved (see figure 4.12).

Saving can be done two ways. Write a name for the particular settings you just set, and choose to save:

- For that view alone, so that the settings will be available to you the next time you open this

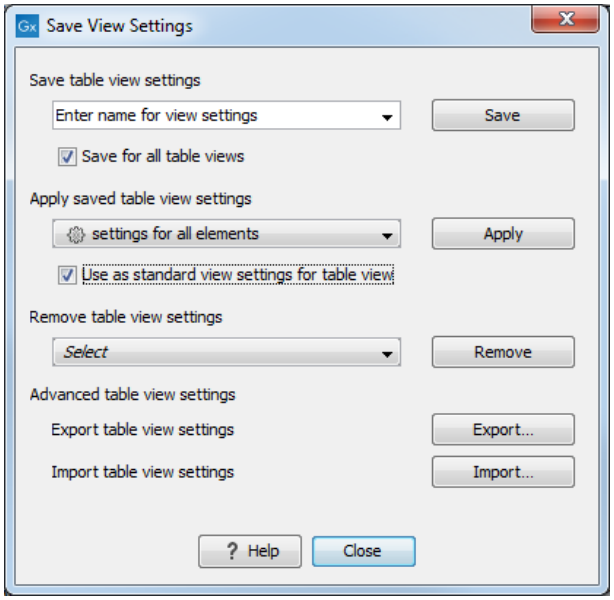


Figure 4.12: *Functionalities found at the bottom of the Side Panel.*

particular element. The settings are saved with only this element, and will be exported with the element if you later select to export the element to another destination.

- For all other views, when the option "Save for all element views" is checked, so that the settings will be available to you the next time you open any element for which this type of view is available.

Similarly, applying can be done two ways:

- For that view alone, so that the settings are applied the next time you open this particular element.
- For all other elements, when the option "Use as standard view settings for element view" is checked, so that the settings are applied each time you open any element for which this type of view is available. These "general" settings are user specific and will not be saved with or exported with the element.

"General" settings can be shared and imported with other workbench users using the **Export** and **Import** buttons at the bottom of the dialog. Exporting and importing saved settings can also be done in the **Preferences** dialog under the **View** tab (see section 4.2.1).

It is possible to remove a saved setting using the saved settings list from the drop-down menu and clicking **Remove**.

Chapter 5

Printing

Contents

5.1 Selecting which part of the view to print 99
5.2 Page setup 100
5.3 Print preview 102

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 6.7) 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.

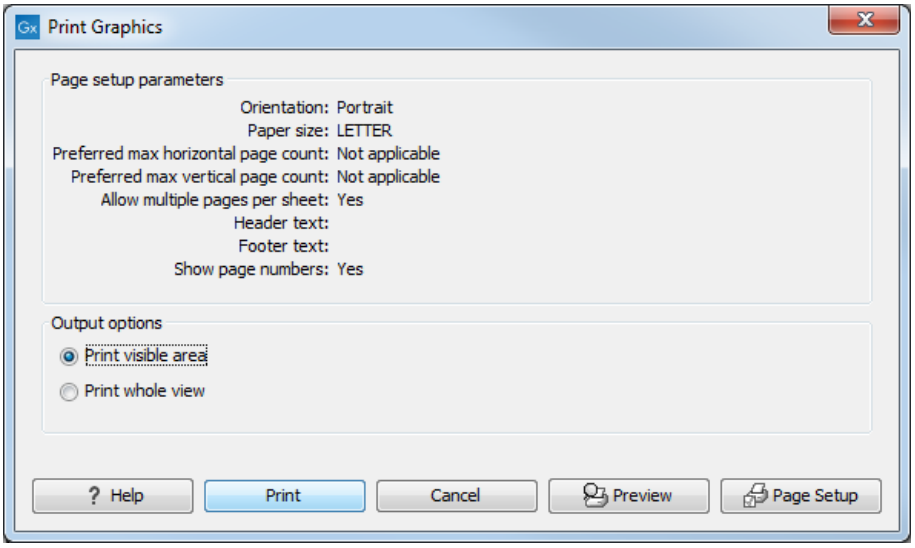


Figure 5.1: *The Print dialog.*

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.

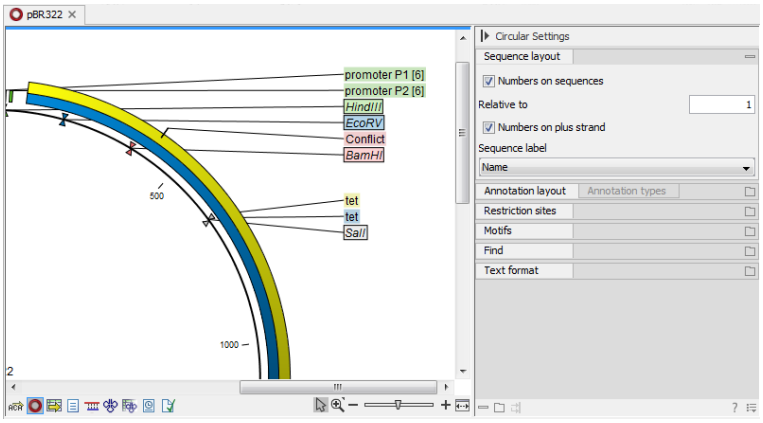


Figure 5.2: *A circular sequence as it looks on the screen.*

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.

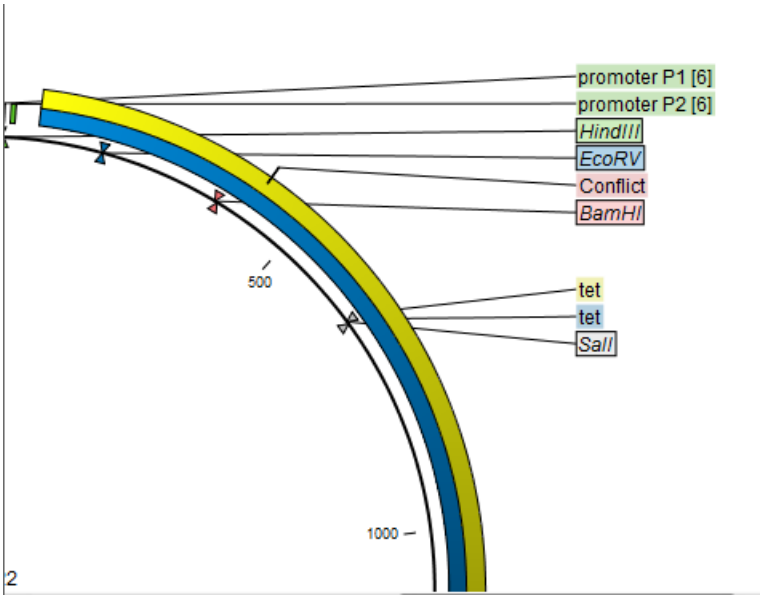


Figure 5.3: *A print of the sequence selecting Print visible area.*

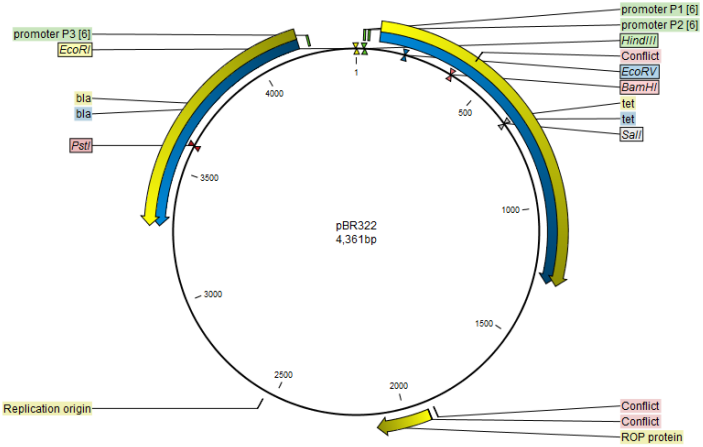


Figure 5.4: *A print of the sequence selecting Print whole view. The whole sequence is shown, even though the view is zoomed in on a part of the sequence.*

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.

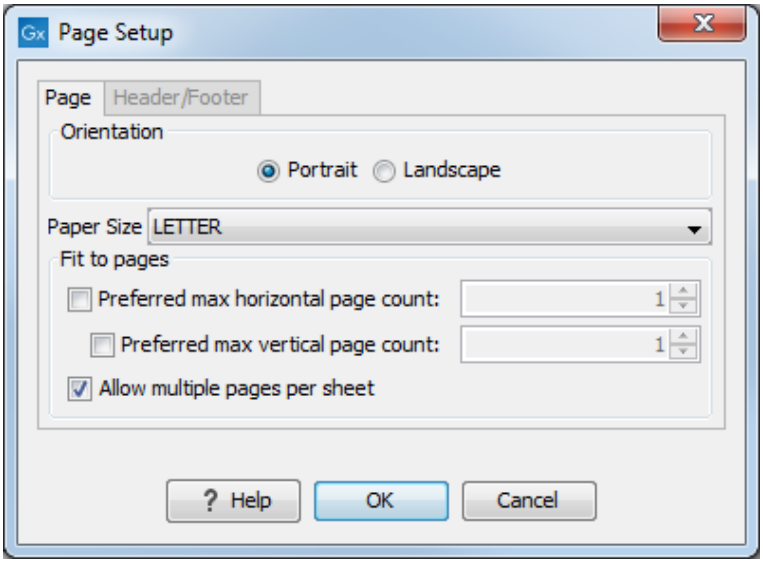


Figure 5.5: *Page Setup.*

- **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.

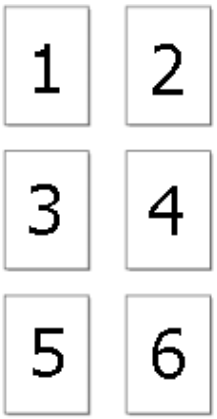


Figure 5.6: *An example where Fit to pages horizontally is set to 2, and Fit to pages vertically is set to 3.*

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.

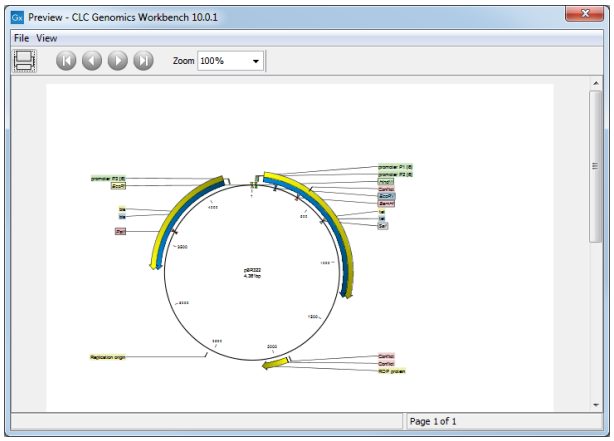


Figure 5.7: *Print preview.*

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

Import/export of data and graphics

Contents

6.1 Standard import . . . 104
6.1.1 External files . . . 105
6.2 Import tracks . . . 106
6.2.1 GFF3 format . . . 109
6.2.2 VCF import . . . 112
6.3 Import high-throughput sequencing data . . . 114
6.3.1 QIAGEN GeneReader . . . 116
6.3.2 Illumina . . . 116
6.3.3 PacBio . . . 121
6.3.4 Fasta read files . . . 122
6.3.5 Sanger sequencing data . . . 124
6.3.6 Ion Torrent . . . 125
6.3.7 General notes on handling paired data . . . 126
6.3.8 SAM and BAM mapping files . . . 126
6.4 Import RNA spike-in controls . . . 129
6.5 Import Primers . . . 130
6.5.1 Import Primer Pairs . . . 130
6.6 Data export . . . 131
6.6.1 Export formats . . . 132
6.6.2 Export parameters . . . 134
6.6.3 Specifying the exported file name(s) . . . 135
6.6.4 Export of folders and data elements in CLC format . . . 136
6.6.5 Export of dependent elements . . . 137
6.6.6 Export of tables . . . 138
6.6.7 Export in VCF format . . . 139
6.6.8 JSON export . . . 144
6.6.9 Graphics export . . . 147
6.6.10 Export history . . . 150
6.6.11 Backing up data from the CLC Workbench . . . 150
6.7 Export graphics to files . . . 152

6.7.1 File formats 154
6.8 Export graph data points to a file 156
6.9 CLC Server data import and export 157
6.10 Copy/paste view output 158

CLC Genomics Workbench handles a large number of different data formats. In order to work with data in the Workbench, it has to be imported ( ). Data types that are not recognized by the Workbench are imported as "external files" which means that when you open these, they will open in the default application for that file type on your computer (e.g. Word documents will open in Word).

This chapter first deals with importing and exporting data in bioinformatic data formats and as external files. Next comes an explanation of how to export graph data points to a file, and how to export graphics.

For **import of NGS data**, please see section 6.3.

6.1 Standard import

CLC Genomics Workbench has support for a wide range of bioinformatic data such as molecules, sequences, alignments etc. See a full list of the data formats in section I.1.1.

These data can be imported through the Import dialog, using drag/drop or copy/paste as explained below.

For **import of NGS data**, please see section 6.3 For import of tracks, please see section 6.2.

Import using the import dialog To start the import using the import dialog:

click Import ( ) in the Toolbar

and choose **Standard Import**

This will show a dialog similar to figure 6.1. You can change which kind of file types that should be shown by selecting a file format in the **Files of type** box.

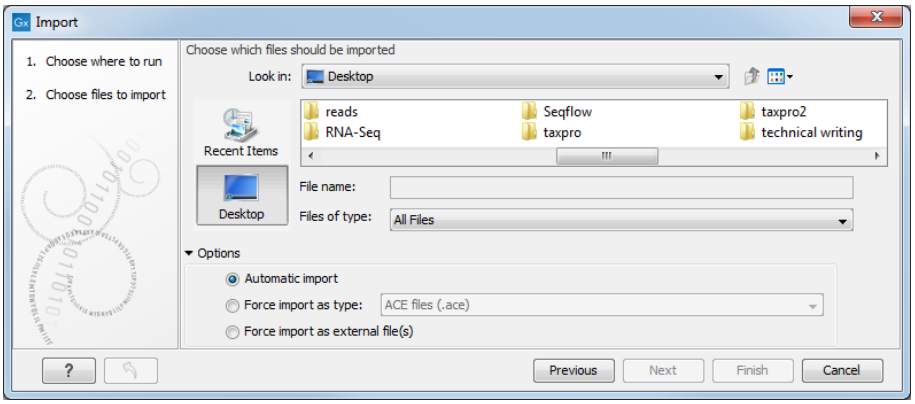


Figure 6.1: *The import dialog.*

Next, select one or more files or folders to import and click **Next** to select a place for saving the result files. If you import one or more folders, the contents of the folder is automatically

imported and placed in that folder in the Navigation Area. If the folder contains subfolders, the whole folder structure is imported.

In the import dialog (figure 6.1), there are three import options:

Automatic import This will import the file and *CLC Genomics Workbench* will try to determine the format of the file. The format is determined based on the file extension (e.g. SwissProt files have .swp at the end of the file name) in combination with a detection of elements in the file that are specific to the individual file formats. If the file type is not recognized, it will be imported as an external file. In most cases, automatic import will yield a successful result, but if the import goes wrong, the next option can be helpful:

Force import as type This option should be used if *CLC Genomics Workbench* cannot successfully determine the file format. By forcing the import as a specific type, the automatic determination of the file format is bypassed, and the file is imported as the type specified.

Force import as external file This option should be used if a file is imported as a bioinformatics file when it should just have been external file. It could be an ordinary text file which is imported as a sequence.

Import using drag and drop It is also possible to drag a file from e.g. the desktop into the **Navigation Area** of *CLC Genomics Workbench*. This is equivalent to importing the file using the **Automatic import** option described above. If the file type is not recognized, it will be imported as an external file.

Import using copy/paste of text If you have e.g. a text file or a browser displaying a sequence in one of the formats that can be imported by *CLC Genomics Workbench*, there is a very easy way to get this sequence into the **Navigation Area**:

Copy the text from the text file or browser | Select a folder in the Navigation Area | Paste ( )

This will create a new sequence based on the text copied. This operation is equivalent to saving the text in a text file and importing it into the *CLC Genomics Workbench*.

If the sequence is not formatted, i.e. if you just have a text like this: "ATGACGAATAGGAGTTC-TAGCTA" you can also paste this into the **Navigation Area**.

Note! Make sure you copy all the relevant text - otherwise *CLC Genomics Workbench* might not be able to interpret the text.

6.1.1 External files

In order to help you organize your research projects, *CLC Genomics Workbench* lets you import all kinds of files. E.g. if you have Word, Excel or pdf-files related to your project, you can import them into the **Navigation Area** of *CLC Genomics Workbench*. Importing an external file creates a copy of the file which is stored at the location you have chosen for import. The file can now be opened by double-clicking the file in the **Navigation Area**. The file is opened using the default application for this file type (e.g. Microsoft Word for .doc-files and Adobe Reader for .pdf).

External files are imported and exported in the same way as bioinformatics files (see section 6.1). Bioinformatics files not recognized by *CLC Genomics Workbench* are also treated as external files.

There is a special tool for importing data from Vector NTI. This tool is a plugin which can be downloaded and installed in the *CLC Genomics Workbench* using the plugin manager (see section 1.5).

6.2 Import tracks

Tracks (see chapter 24) are imported in a special way, because extra information is needed in order to interpret the files correctly.

Tracks are imported using: **click Import ( ) in the Toolbar | Tracks**
This will open a dialog as shown in figure 6.2.

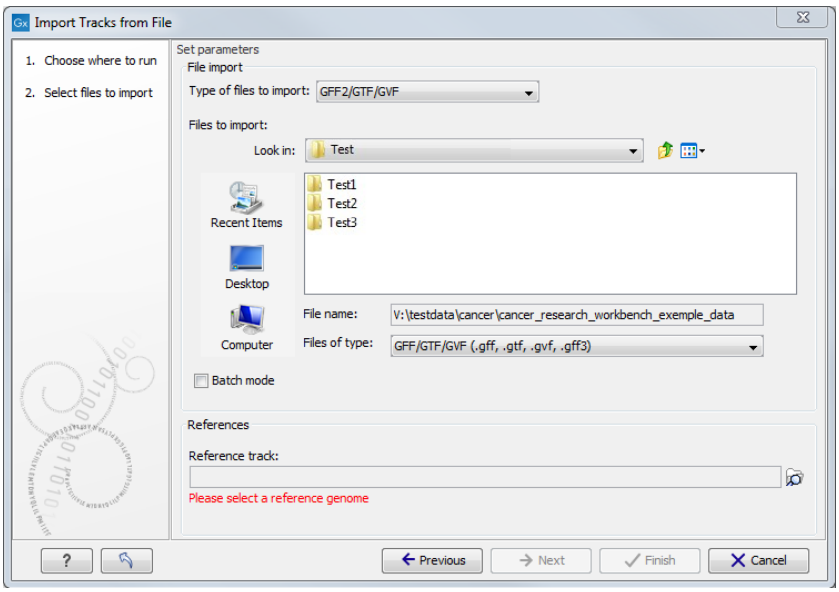


Figure 6.2: *Select files to import.*

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 6) and subsequently converting the sequence or sequence list to a track (see section 24.7).

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 http://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 workbench. You can also read more about these formats at http://www.sanger.ac.uk/resources/software/gff/spec.html, http://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 workbench handles GFF3 format in section 6.2.1.

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 http://www.1000genomes.org/data#DataAccess. Learn more about how the workbench handles VCF format in section 6.2.2.

BED Simple format for annotations. Read more at http://genome.ucsc.edu/FAQ/FAQformat.html#format1. This format is typically used for very simple annotations, for example target regions for sequence capture methods. The file to import must have the first three columns (chromosome, start and end positions) matching the UCSC specifications. Remaining columns that do not match these requirements will be imported as Var1, Var2, etc.

Wiggle The Wiggle format as defined by UCSC (http://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 http://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: http://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: http://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 UCSC web site here: http://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 Complete mutation data: CosmicCompleteTargetedScreensMutantExport.tsv
- COSMIC Mutation Data (Genome Screens): CosmicGenomeScreensMutantExport.tsv
- COSMIC Mutation Data : CosmicMutantExport.tsv
- All Mutations in Census Genes : CosmicMutantExportCensus.tsv

From version 91, COSV IDs are used instead of COSM, with each COSV ID imported as a single variant with information from all relevant transcripts and samples.

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 27.1.6.)

Note: Import of version 90 COSMIC TSV files is not supported, due to issues with that version.

Please see chapter I.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 http://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. When the data cannot be recognized as variant data, one track is created for each annotation type. Genome / gene annotation tracks can be automatically imported from relevant databases as described here: http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Selecting_data_types_download.html.

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.

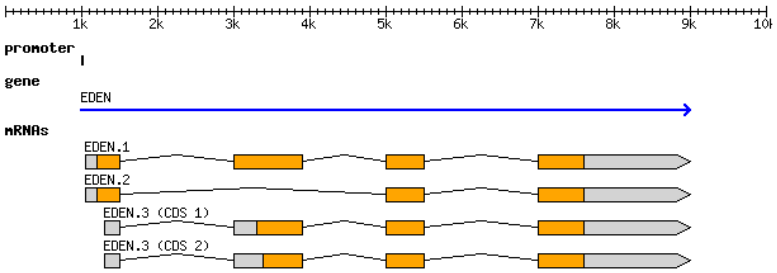
6.2.1 GFF3 format

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 6.3 exemplifies how tags are used to create annotations.




```
##gff-version 3.2.1
##sequence-region ctg123 1 1497228
ctg123 . gene            1000  9000  .  +  .  ID=gene00001;Name=EDEN
ctg123 . TF_binding_site 1000  1012  .  +  .  ID=tfbs00001;Parent=gene00001
ctg123 . mRNA            1050  9000  .  +  .  ID=mRNA00001;Parent=gene00001;Name=EDEN.1
ctg123 . mRNA            1050  9000  .  +  .  ID=mRNA00002;Parent=gene00001;Name=EDEN.2
ctg123 . mRNA            1300  9000  .  +  .  ID=mRNA00003;Parent=gene00001;Name=EDEN.3
ctg123 . exon            1300  1500  .  +  .  ID=exon00001;Parent=mRNA00003
ctg123 . exon            1050  1500  .  +  .  ID=exon00002;Parent=mRNA00001,mRNA00002
ctg123 . exon            3000  3902  .  +  .  ID=exon00003;Parent=mRNA00001,mRNA00003
ctg123 . exon            5000  5500  .  +  .  ID=exon00004;Parent=mRNA00001,mRNA00002,mRNA00003
ctg123 . exon            7000  9000  .  +  .  ID=exon00005;Parent=mRNA00001,mRNA00002,mRNA00003
ctg123 . CDS             1201  1500  .  +  0  ID=cds00001;Parent=mRNA00001;Name=edenprotein.1
ctg123 . CDS             3000  3902  .  +  0  ID=cds00001;Parent=mRNA00001;Name=edenprotein.1
ctg123 . CDS             5000  5500  .  +  0  ID=cds00001;Parent=mRNA00001;Name=edenprotein.1
ctg123 . CDS             7000  7600  .  +  0  ID=cds00001;Parent=mRNA00001;Name=edenprotein.1
ctg123 . CDS             1201  1500  .  +  0  ID=cds00002;Parent=mRNA00002;Name=edenprotein.2
ctg123 . CDS             5000  5500  .  +  0  ID=cds00002;Parent=mRNA00002;Name=edenprotein.2
ctg123 . CDS             7000  7600  .  +  0  ID=cds00002;Parent=mRNA00002;Name=edenprotein.2
ctg123 . CDS             3301  3902  .  +  0  ID=cds00003;Parent=mRNA00003;Name=edenprotein.3
ctg123 . CDS             5000  5500  .  +  1  ID=cds00003;Parent=mRNA00003;Name=edenprotein.3
ctg123 . CDS             7000  7600  .  +  1  ID=cds00003;Parent=mRNA00003;Name=edenprotein.3
ctg123 . CDS             3391  3902  .  +  0  ID=cds00004;Parent=mRNA00003;Name=edenprotein.4
ctg123 . CDS             5000  5500  .  +  1  ID=cds00004;Parent=mRNA00003;Name=edenprotein.4
ctg123 . CDS             7000  7600  .  +  1  ID=cds00004;Parent=mRNA00003;Name=edenprotein.4
```

Figure 6.3: *Example of a GFF3 file and the corresponding annotations from https://github.com/The-Sequence-Ontology/Specifications/blob/master/gff3.md.*

In the workbench, the GFF3 importer will create an output track for each feature type present in the file. In addition, the 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 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 6.4 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 6.4: *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 6.5 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 6.5: *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 6.6.



Figure 6.6: *Naming of features.*

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 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.
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.

6.2.2 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, or AO. Where more than one of these is present, they are prioritized in the following order:

1. CLCAD2
2. AD
3. AO

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 6.2. 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: newAD=origAD*(refoPloidy-numReplaced)/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: 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 section 6.6.7.

6.3 Import high-throughput sequencing data

CLC Genomics Workbench has dedicated tools for importing data from the following High-throughput sequencing systems:

- QIAGEN GeneReader
- Illumina Genome Analyzer, Nextseq, HiSeq and MiSeq
- PacBio
- Ion Torrent

Sequencing data from these systems, as well as Sanger and Fasta format files, can be imported using dedicated tools. Alternatively, this data can be imported using the on-the-fly functionality available in workflows, described in section 11.4.

Importing other NGS related formats

- There are dedicated NGS importers for Sanger or Fasta format data.
- There is a dedicated import tool for read mappings in SAM/BAM format. Alignments of *Complete Genomics* data can be imported using this.

- An importer for Roche 454 sequencing data is available in the Legacy Tools folder.
- Complete Genomics master VAR files can be converted to VCF using tools provided by Complete Genomics, and imported into the *CLC Genomics Workbench* using the VCF track importer.

Once imported, data originating from any sequencing platform can be analyzed in the *CLC Genomics Workbench*.

Clicking on the **Import** () button in the top toolbar will bring up a list of the supported data types as shown in figure 6.7. Select the appropriate format to launch the importer.

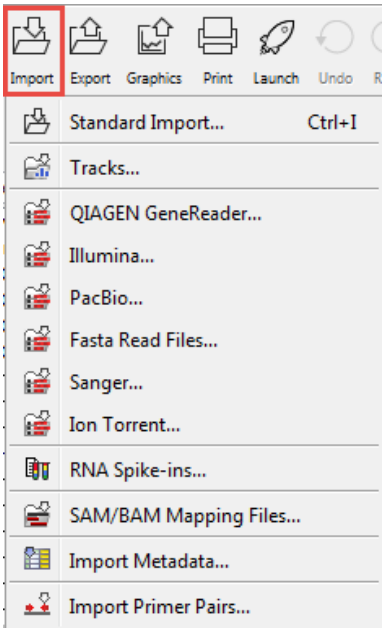


Figure 6.7: *Choosing what kind of data you wish to import.*

To specify the files to import, select either **Add folders**, in which case you then choose one or several folders from which all the files should be imported, or **Add files**, in which case you select individual files to import. Once files have been selected, configure the import options, which are described in the following sections.

Files can be removed from the list by selecting them and clicking on the **Remove** button.

If the wrong NGS importer was used to import your data, please check, and edit if necessary, the "Read Group" information in the "Element Info" view. To edit this information, choose from the drop-down menu the sequencing platform used to generate the data (figure 6.8) and click **OK**.

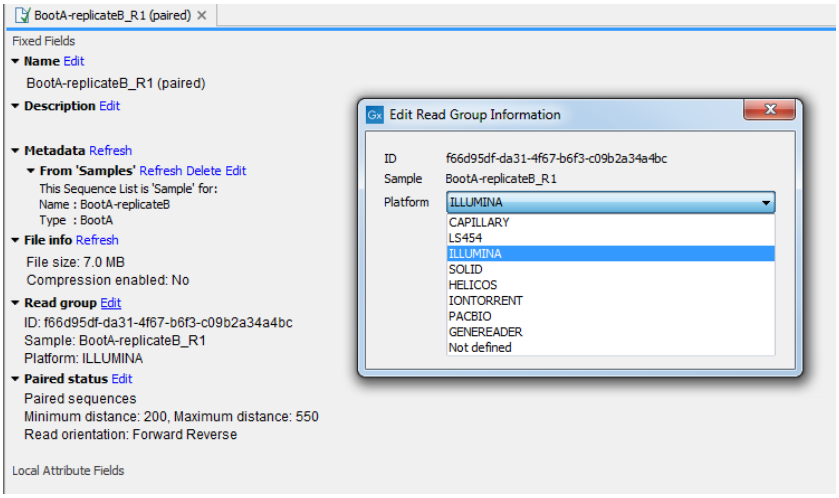


Figure 6.8: *Editing the platform used to generate the data in the "Element Info" view.*

6.3.1 QIAGEN GeneReader

CLC Genomics Workbench supports data from QIAGEN GeneReader. Choosing the QIAGEN GeneReader import will open the dialog shown in figure 6.9. This data type can also be imported using the on-the-fly import functionality available in workflows, described in section 11.4.

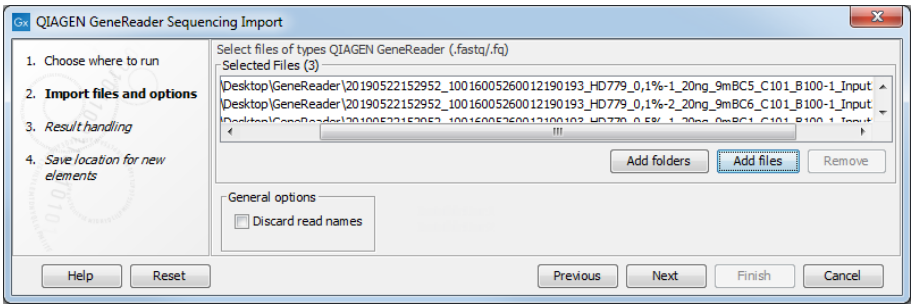


Figure 6.9: *Importing data from QIAGEN GeneReader.*

The file formats accepted are Fastq *.fastq and *.fq. You can choose to discard read names during import.

6.3.2 Illumina

CLC Genomics Workbench supports data from Illumina's Genome Analyzer, HiSeq 2000, NextSeq and the MiSeq systems. Choosing the Illumina importer opens the dialog shown in figure 6.10. This data type can also be imported using the on-the-fly import functionality described in section 11.4.

File format Fastq format files and fastq format files that compressed using gzip (.gz), zip (.zip) or bzip2 (.bz2) can be imported using the Illumina importer.

General Options The settings in the **General options** area of the dialog are:

- **Paired reads**. Enable this option when importing **Paired-end** or **Mate-pair** data.

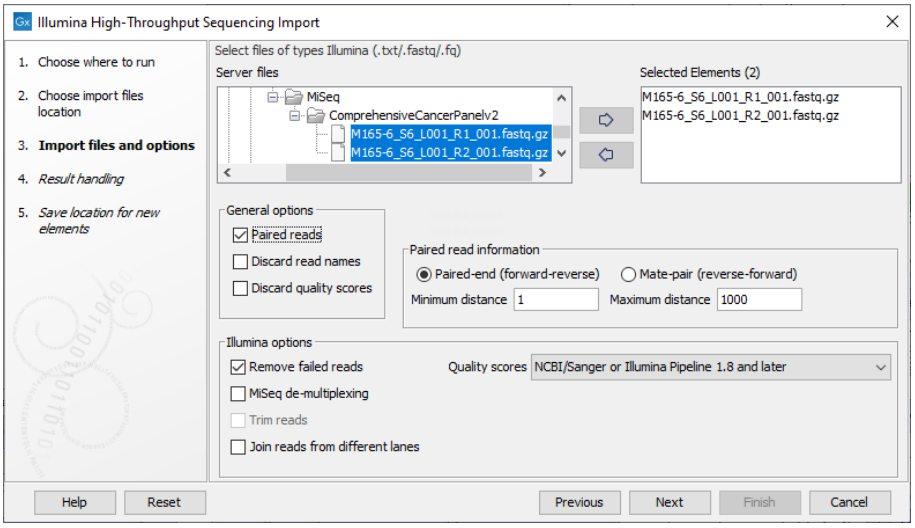


Figure 6.10: *Importing data from Illumina systems.*

When enabled, you can specify the paired data type and the expected distance range in the "Paired read information" section of the dialog, which is described later in this section.

For paired data, pairs of files should be selected. The first reads of read pairs are expected in one file and the second reads of the pairs in another file. Multiple pairs of files can be selected. To determine which files form pairs, the files are sorted based on their names. Rules are then applied to determine whether the pairs of files are valid pairs.

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 following checks are then carried out for each prospective pair of files to determine whether those files form a valid pair:

- If the file names appear to follow the following naming format: `<sample name>_L<at least one` then the name of each file in the pair must have the same sample name and lane information. If they do not, no data is imported from those files and a message is printed in the log.

- If the file names do not follow the naming format described above, but do contain "R1" or "R2" in their names, then the first file of the pair must contain "R1" in the name and the second file name must contain "R2". If this condition is not met, no data is imported from those files and a message is printed in the log. Note that if "R1" or "R2" appear more than once in a filename, the last instance in the name is used.
- If the file names do not match either of the cases above, then import is allowed to proceed. I.e. No further checks are done to attempt to validate if the pairs of files, as per their order in the sorted list, are a valid pair based on their filenames.

If the **Join reads from different lanes** option, in the Illumina options section of the dialog, is enabled, 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.

- **Discard read names**. For high-throughput sequencing data, the naming of the individual reads is often irrelevant given the huge amount of reads. This option allows you to discard read names to save disk space.

  Note: If you do not choose to discard read names, you can quickly check that the imported data contains the expected pairs by looking at the first few sequence names of the imported sequence list in the *CLC Genomics Workbench*. The first two sequences should have the same name, except for a `1` or a `2` somewhere in the read name line.

- **Discard quality scores**. Quality scores are visible in the mapping view and they are used during variant detection. If this is not relevant for your work, you can enable the **Discard quality scores** option. This can reduce disk space usage and memory consumption. Read more about the quality scores of Illumina data below.

Paired read information When the **Paired reads** option is enabled, options in the "Paired read information" section of the dialog can be edited. Here, you specify the type of paired data, Paired-end (forward-reverse) or Mate-pair (reverse-forward), and the expected distance range.

In the Workbench, the only difference between paired-end (forward-reverse) or mate-pair (reverse-forward) is the expected orientation of the reads: forward-reverse in the case of paired end data and reverse-forward in the case of mate pairs.

The paired read distance includes the full read sequence, i.e. from the beginning of the forward read to the beginning of the reverse read (figure 6.11). The distances are usually defined during the library preparation of your sequencing experiment, but in doubt you can enter default values: for paired-end the distances are between 1 and 1000 bp while mate-pair reads typically have longer distances between 1000-5000 bp (and sometimes up to 10000). Note that the tools usually used subsequently to process Illumina reads (such as Map Reads to Reference or RNA-Seq Analysis) have an "Auto-detect paired distances" option that is enabled by default. As long as this option is used, mis-specifying the distances during import should bear no consequences.

Read more about handling paired data in section 6.3.7.

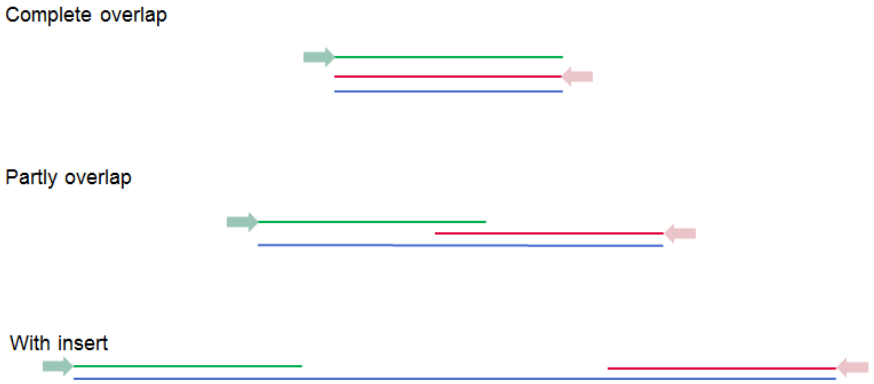


Figure 6.11: *Green lines represent forward reads, red lines reverse reads, and in blue is shown the distance of the sequenced DNA fragment. Thus, if there is a complete overlap, the minimum distance will not be 0, but the length of the overlap.*

Illumina options

- **Remove failed reads**. If you check **Remove failed reads**, reads that did not pass a quality filter (as indicated within the fastq files) will be ignored during import.

  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 enabled, 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 enabled, 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 patterns occurs more than once in a filename, the last instance in the name is used. For example, if filenames were `myFile_L001_L1.fastq` then the lane information is taken to be L1.

In the next wizard step, options are presented for how to handle the results (see section 9.2). 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 9.3).

Quality scores in the Illumina platform

When using the Illumina importer, you can select the quality score scheme applicable for your data at the bottom of the dialog (figure 6.12).

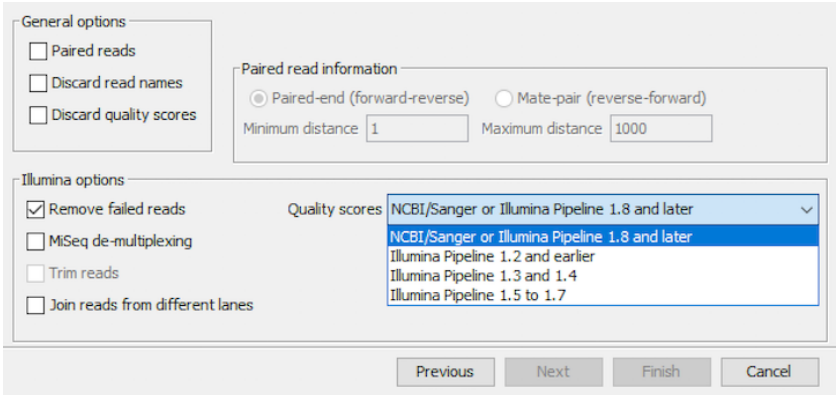


Figure 6.12: *Selecting the quality score scheme.*

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 http://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
```

```
TTTTTGTAAGGAGGGGGGTCATCAAAATTTGCAAAA
+SRR001926.1 FC00002:7:1:111:750 length=36
IIIIIIIIIIIIIIIIIIIIIIIIIFIIII'IB<IH
@SRR001926.7 FC00002:7:1:110:453 length=36
TTATATGGAGGCTTTAAGAGTCATAGGTTGTTCCCC
+SRR001926.7 FC00002:7:1:110:453 length=36
IIIIIIIIII:'III?=IIIIII+&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
ccccccccccccccccccc]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`]_aaaaaaaaa[ER`abaaa\aaaaaaaa[
```

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 http://en.wikipedia.org/wiki/Ascii#ASCII_printable_characters.

6.3.3 PacBio

Choosing the PacBio importer will open the dialog shown in figure 6.13. This data type can also be imported using the on-the-fly import functionality described in section 11.4.

We support import of the following file formats containing PacBio reads:

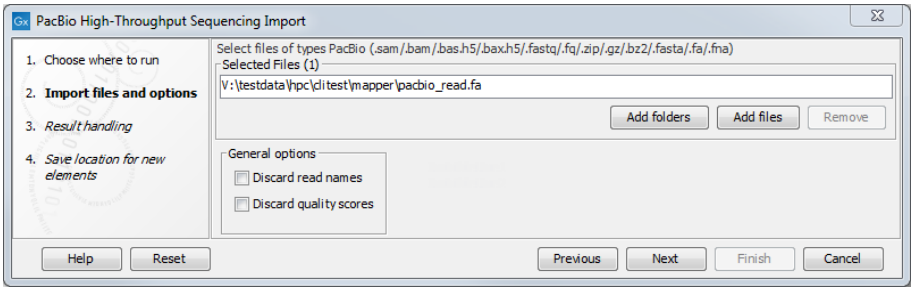


Figure 6.13: *Importing data from PacBio.*

- 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 files (`.fastq`) which contain sequence data and quality scores. Compressed Fastq (`.fastq.gz`) files are also supported.
- Fasta files (`.fasta`) which contain sequence data. Compressed Fasta (`.fasta.gz`) files are also supported.
- SAM or BAM files (`.sam/.bam`) .The mapping information is discarded during import.

Under **General options** you have the following choices:

- **Discard read names**. For high-throughput sequencing data, the naming of the individual reads is often irrelevant given the huge amount of reads. This option allows you to discard read names to save disk space.
- **Discard quality scores**. Quality scores can be visualized in the mapping view and used for SNP detection. If this is not relevant for your work, you can choose to **Discard quality scores**. Discarding quality scores will reduce both disk space usage 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.

Click **Next** and choose how the result of the import should be handled. We recommend choosing **Save** which will save the results directly to the disk.

When opening the "Element info" of sequence lists imported with the PacBio importer, the item "Platform" will display the mention PACBIO. For PacBio reads imported without the PacBio importer, it is possible to edit that field to "PACBIO" by clicking **Edit** next to the "Read Group" section in the Element Info view. Having the platform set to PacBio will ensure that the read mapper will perform better on PacBio reads.

6.3.4 Fasta read files

The **Fasta** importer is designed for high volumes of read data such as high-throughput sequencing data (NGS reads). When using this import option the read names can be included but the

descriptions from the fasta files are ignored. This data type can also be imported using the on-the-fly import functionality available in workflows, described in section 11.4.

For import of other fasta format data, such as reference sequences, please use the**Standard Import** ( ) as this import format also includes the descriptions. To have a reference in track format, use the **Tracks** ( ) option and set the "Type of file to import" to FASTA.

The dialog for importing data in fasta format is shown in figure 6.14.

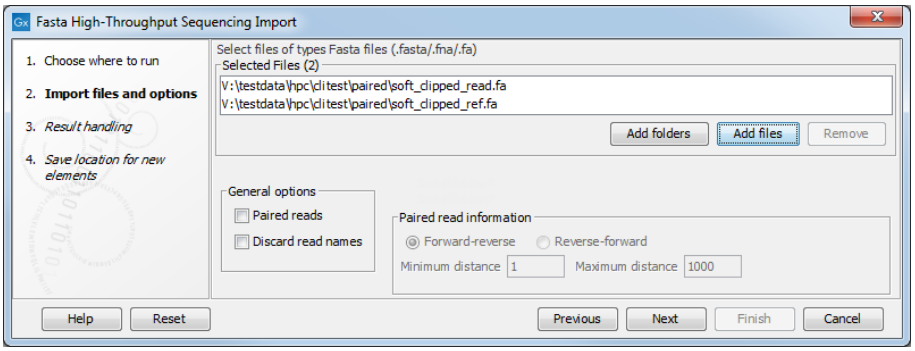


Figure 6.14: *Importing data in fasta format.*

Compressed data in gzip format is also supported (.gz).

The **General options** to the left are:

- **Paired reads**. For paired import, the Workbench expects the forward reads to be in one file and the reverse reads in another. The Workbench will sort the files before import and then assume 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 6.3.7.
- **Discard read names**. For high-throughput sequencing data, the naming of the individual reads is often irrelevant given the huge amount of reads. This option allows you to discard this option to save disk space.
- **Discard quality scores**. This option is not relevant for fasta import, since quality scores are not supported.

Click **Next** to adjust how to handle the results (see section 9.2). 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 handy for better organizing subsequent analysis results and for batching (see section 9.3).

6.3.5 Sanger sequencing data

Although traditional sequencing data (with chromatogram traces like abi files) is usually imported using the standard **Import** (), see section 6, this option has also been included in the High-Throughput Sequencing Data import. It is designed to handle import of large amounts of sequences, and there are three differences from the standard import:

- All the sequences will be put in one sequence list (instead of single sequences).
- The chromatogram traces will be removed (quality scores remain). This is done to improve performance, since the trace data takes up a lot of disk space and significantly impacts speed and memory consumption for further analysis.
- Paired data is supported.

With the standard import, it is practically impossible to import up to thousands of trace files and use them in an assembly. With this special High-Throughput Sequencing import, there is no limit. The import formats supported are the same: ab, abi, ab1, scf and phd. For all formats, compressed data in gzip format is also supported (.gz).

This data type can also be imported using the on-the-fly import functionality available in workflows, described in section 11.4.

The dialog for importing data Sanger sequencing data is shown in figure 6.15.

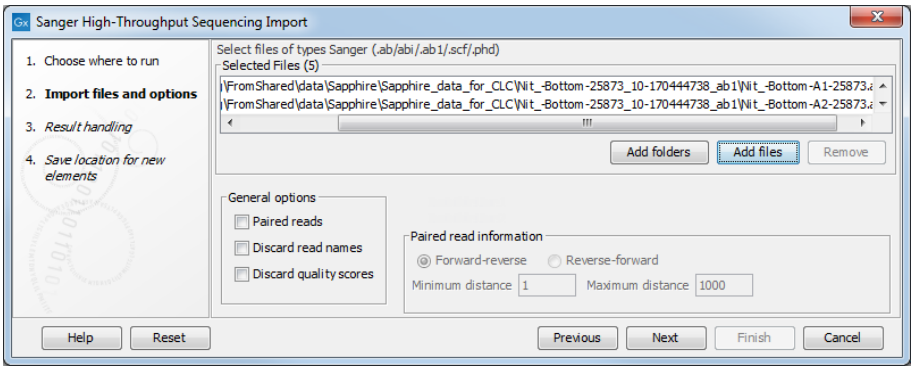


Figure 6.15: *Importing data from Sanger sequencing.*

The **General options** to the left are:

- **Paired reads**. The Workbench will sort the files before import and then assume 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` for the forward read and `sample1_rev` for the reverse reads. 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 6.3.7.
- **Discard read names**. For high-throughput sequencing data, the naming of the individual reads is often irrelevant given the huge amount of reads. This option allows you to discard this option to save disk space.

- **Discard quality scores**. Quality scores are visualized in the mapping view and they are used for SNP detection. If this is not relevant for your work, you can choose to **Discard quality scores**. One of the benefits from discarding quality scores is that you will gain a lot in terms of reduced disk space usage and memory consumption.

Click **Next** to adjust how to handle the results (see section 9.2). 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 handy for better organizing subsequent analysis results and for batching (see section 9.3).

6.3.6 Ion Torrent

Choosing the Ion Torrent import will open the dialog shown in figure 6.16. This data type can also be imported using the on-the-fly import functionality available in workflows, described in section 11.4.

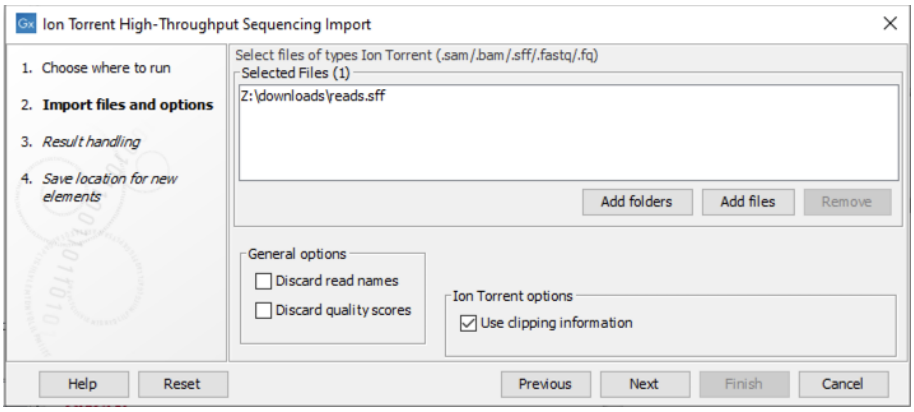


Figure 6.16: *Importing data from Ion Torrent.*

The following file formats from the Ion Torrent system can be imported:

- SFF (`.sff`) Sff files may provide extra information about adapter regions or regions of low quality.
- Fastq (`.fastq`). Quality scores are expected to be in the NCBI/Sanger format (see section 6.3.2). Compressed data in gzip format is also supported (.gz).
- SAM or BAM (`.sam`/`.bam`). Mapping information is discarded during import.

The **General options**, at the bottom left of the import dialog are:

- **Discard read names**. For high-throughput sequencing data, the names of the individual reads is often irrelevant. This option allows you to discard the read names to save disk space.
- **Discard quality scores**. Quality scores are visualized in the mapping view and they are used for variant detection. If this is not relevant for your work, you can select this option to discard the quality scores, to save disk space and decrease memory consumption.

The option **Use clipping information**, at the bottom right, applies to the import of sff format files, indicating whether clipping information in the files should be used or not.

6.3.7 General notes on handling paired data

During import, information about paired data (distances and orientation) can be specified (see figure 6.10) and stored by the *CLC Genomics Workbench*. All subsequent analyses automatically take differences in orientation into account. Once imported, both reads of a pair will be stored in the sam sequence list. The forward and reverse reads (e.g. for paired-end data) simply alternate so that the first read is forward, the second read is the mate revere read; the third is again forward and the fourth read is the mate reverse read and so on. When manipulating sequence lists with paired data, be careful not break this order.

You can view and edit the orientation of the reads after they have been imported by opening the read list in the Element information view ( ), see section 12.4, as shown in figure 6.17.

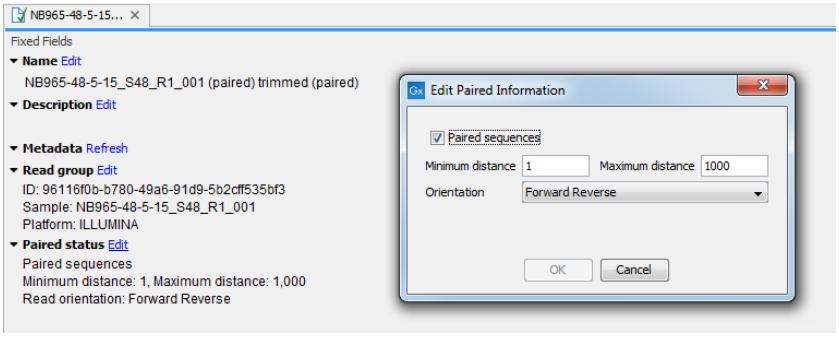


Figure 6.17: *The paired orientation and distance.*

In the **Paired status** part, you can specify whether the *CLC Genomics Workbench* should treat the data as paired data, what the orientation is and what the preferred distance is. The orientation and preferred distance is specified during import and can be changed in this view. If the "Paired sequences" box is unchecked, the sequences will be handled as single (non paired) data.

Note that the **paired distance** measure that is used throughout the *CLC Genomics Workbench* is always *including the full read sequence*. For paired-end libraries it means from the beginning of the forward read to the beginning of the reverse read.

6.3.8 SAM and BAM mapping files

The *CLC Genomics Workbench* supports import and export of files in SAM (Sequence Alignment/Map) and BAM format, which are designed for storing large nucleotide sequence alignments. Read more and see the format specification at `http://samtools.sourceforge.net/`

The Workbench includes support for importing SAM and BAM files from **Complete Genomics**.

Note! If you wish to import the reads in a SAM/BAM file as a sequence list, disregarding any mapping information, please use the Standard import tool instead (see section 6.1).

For a detailed explanation of the SAM and BAM files exported from *CLC Genomics Workbench*, please see Appendix J.

Input data for importing a mapping from a SAM/BAM file To import a mapping from a SAM/BAM file containing mapping data into the Workbench, you need to:

- Provide the SAM/BAM file
- Specify the reference sequences that are referred to within that file. The references can either be sequences already imported into the Workbench, or, if appropriately recorded in the SAM/BAM file, can be fetched from URLs specified in the SAM/BAM file.

The mapping is built up within the Workbench using the reference sequence data, the reads and the information from the SAM/BAM file about how the reads are associated with a particular reference.

Data created in the Workbench after importing a SAM/BAM mapping file

- Reads recorded as mapping to a particular reference that is known inside the Workbench are imported as part of the mapping for that reference.
- Reads recorded as not mapping to any reference are imported into a sequence list.
  - If they are part of an intact pair, they are imported into a sequence list of paired data.
  - If they 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.

  One list is made per read group, with the potential that several such lists could be produced from a single mapping import. If you do not wish to import the unmapped reads, deselect the **Import unmapped reads** option in the final step of the tool dialog.
- Reads recorded as mapping to a reference sequence that is **not** known within the Workbench are not imported.

When setting up the import, you are given the option of creating a track-based mapping, or a stand-alone mapping. In the latter case, if there is only one reference sequence, the result will be a single read mapping ( ). When there is more than one reference sequence, a multi- mapping object ( ) is created.

Please note that mappings within the *CLC Genomics Workbench* do not allow for an individual read sequence to map to more than one location. In cases where a SAM/BAM file contains multiple alignment records for a single read, only one such record will be used to build the mapping.

Running the SAM/BAM Mapping Files importer Click on the Import button on the toolbar or go to:

File | **Import ( )** | **SAM/BAM Mapping Files ( )**

This will open a dialog where you select the SAM/BAM file to import as well as the reference sequences to be used (Figure 6.18).

When you select the reference sequence(s) two options exist:

1. Select a matching reference sequence that has already been imported into the Workbench. Click on the "Find in folder" icon ( ) to localize the reference sequence.
2. If the SAM/BAM file already contains information about where to find the reference sequence, tick the "Download references" box to automatically download the reference sequence.

The selected reference sequence(s) will be listed under "References in files" with "Name", "Length", and "Status". Whenever the correct reference sequence (with the correct name and sequence length) has been selected the "Status" field will indicate this with an "OK". The length of your reference sequence must **match exactly** the length of the reference specified in the SAM/BAM file. The name is more flexible as it allows a range of different "synonyms" (with no distinction between capital and lowercase letters). E.g. for chromosome 1 the allowed synonyms would be: 1, chr1, chromosome_1, nc_000001, for chromosome M: m, mt, chrm, chrmt, chromosome_m, chromosome_mt, nc_001807, for chromosome X: x, chrx, chromosome_x, nc_000023, and for chr Y: y, chry, chromosome_y, nc_000024.

If there are inconsistencies in the names or lengths of the reference sequences being chosen and those recorded in the SAM/BAM file, a comment (for example, "Length differs" or "Input missing") will appear in the "Status" column of the table "References in files". Note that if you are using a CLC Genomics Server to import files located on the Server (rather than locally), checks for corresponding reference names and lengths cannot be carried out and this table will remain empty. This means that you will be able to continue to launch the import regardless of whether the correct references were specified, leading to an error in cases where the references were incorrect.

Unmatched reads (reads that are mapped to an unmatched reference e.g. a SAM reference for which there is no CLC reference counterpart) are not imported. The same is the case whenever inconsistencies have occurred with respect to name or length. The log lists all mapping data or unmatched reads that were not imported and marks whether import failed because of unmatched reads being present in the SAM/BAM file or because of inconsistencies in name/length.

Notes regarding reference sequence naming Reference sequences in a SAM/BAM file **cannot contain spaces**. If the name of a reference sequence in the Workbench contains spaces, the Workbench assume that the names of the references in the SAM file will be the same as the names of the References within the Workbench, but with all spaces removed. For exapmple, if your reference sequence in the Workbench was called `my reference sequence`, the Workbench would recognize a reference in the SAM file as the appropriate reference if it was of the same length and had the name `myreferencesequence`.

Neither the `@` character nor the `=` character are allowed within reference sequence names in SAM files. Any instances of these characters in the name of a reference sequence in the Workbench will be replaced with a _ for the sake of identifying the appropriate reference when importing a SAM or BAM file. For example, if a reference sequence in the Workbench was called `my=reference@sequence`, the Workbench would recognize a reference in the SAM file as the appropriate reference if it was of the same length and had the name `my_reference_sequence`.

Click **Next** to specify how to handle the results (Figure 6.19). Under **Output options** the "Save downloaded reference sequence" will be enabled if the "Download references" box was ticked in the previous step (which would be the case when the SAM/BAM file contained information about where to find the reference sequence e.g. if the SAM/BAM file came from an external provider).

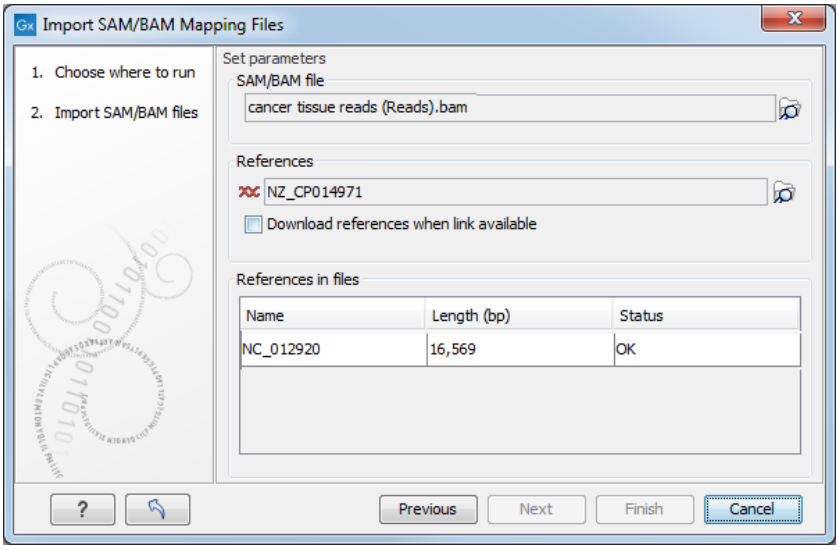


Figure 6.18: *Defining SAM/BAM file and reference sequence(s).*

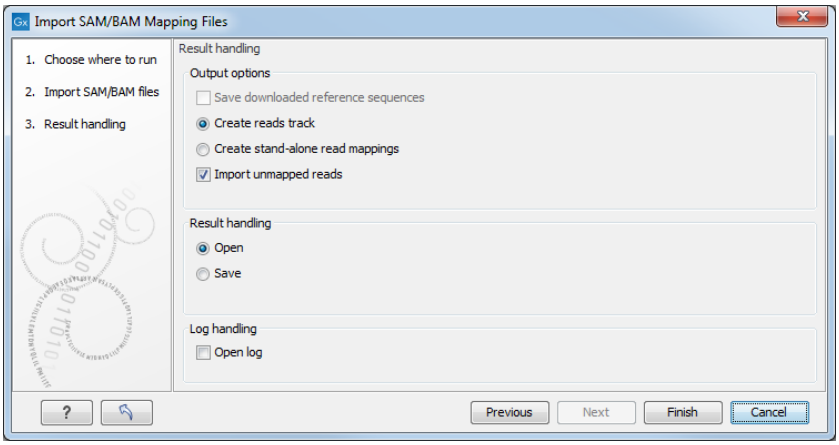


Figure 6.19: *Specify the result handling.*

Ticking the "Create Reads Track" box results in the generation of a track-based mapping. Alternatively, the "Create Stand-Alone Read Mapping" results in a normal read mapping file. By ticking the "Import unmapped reads" box, a sequence list of the unmapped reads will be created. To avoid importing unmapped reads, untick this box.

We recommend choosing **Save** in order to save the results directly to a folder, as you will probably wish to save the data anyway before proceeding with your analysis. For further information about how to handle the results, (see section 9.2).

Note that this import operation is very memory-consuming for large data sets, and particularly those with many reads marked as members of broken pairs in the mapping.

6.4 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 6.20).

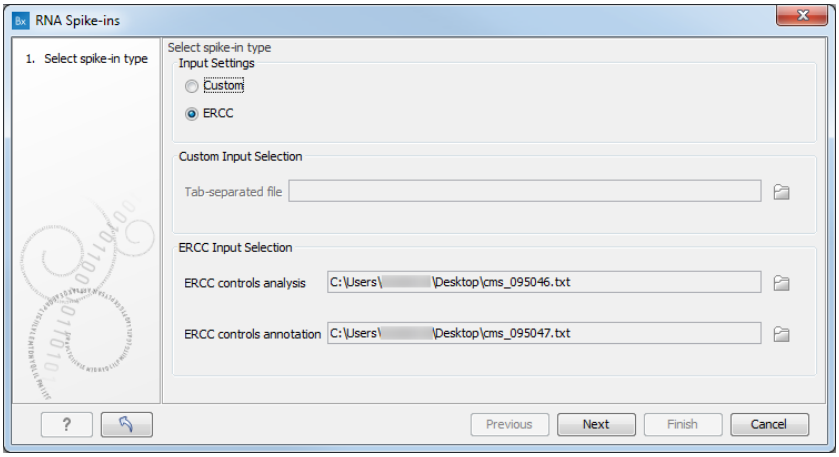


Figure 6.20: *The RNA spike-in controls importer.*

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 "0" 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.

6.5 Import Primers

6.5.1 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 () | **Import Primer Pairs** ()

This will open the wizard shown in figure 6.21. The first step is to select the data to import.

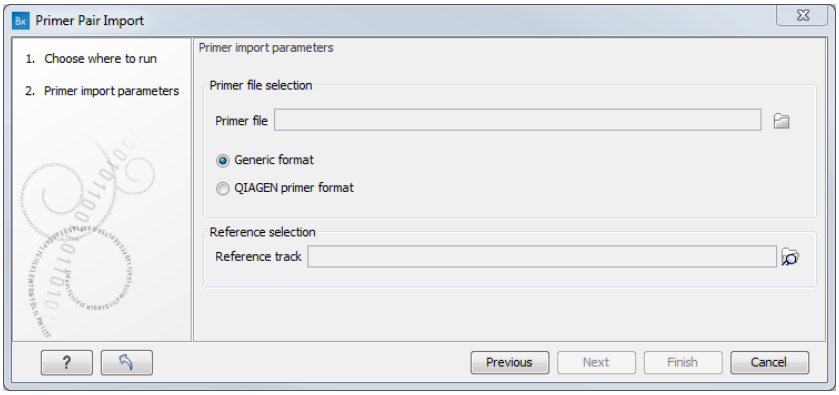


Figure 6.21: *Select files to import.*

- **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.

6.6 Data export

Data can be exported from the *CLC Genomics Workbench* to many standard formats. Supported formats are listed in section I.1.1, but an easy way to see the full list is to launch the Export tool, where they are presented in the first dialog window.

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 6.6.5.

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**.

6.6.1 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 6.22).

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 6.23. 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. This is described in more detail in section 6.6.4.

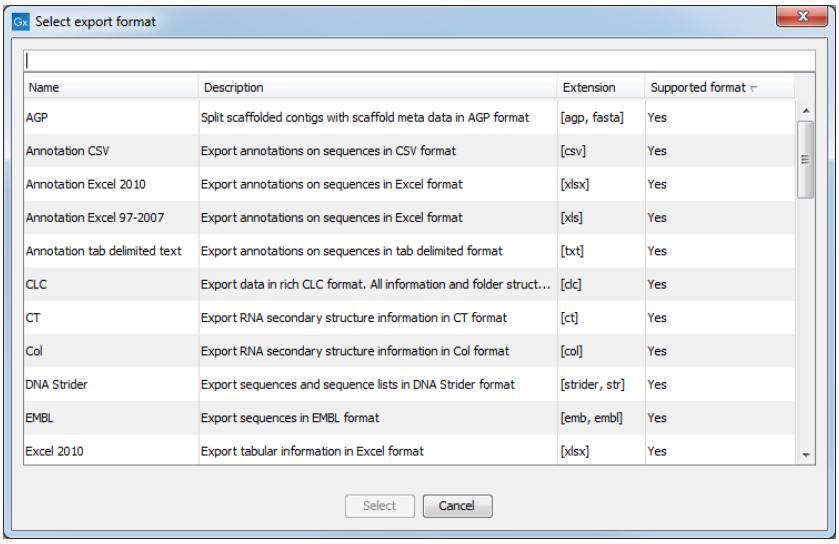


Figure 6.22: *The Select export format dialog. Here, some sequence lists had been selected in the Navigation Area before the Export tool was launched. The formats that the selected data elements can be exported to contain a "Yes" in the Selected format column. Other export formats are listed below the supported ones, with "No" in the Supported format column.*

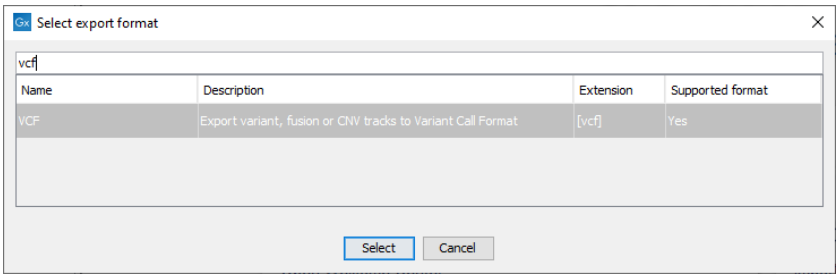


Figure 6.23: *The text field has been used to search for the term "VCF" in the export format name or description field in the Select export dialog.*

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 6.24 we show the selection of a variant track for export to VCF format.

Further information is available about exporting the following types of information:

- Tables: section 6.6.6

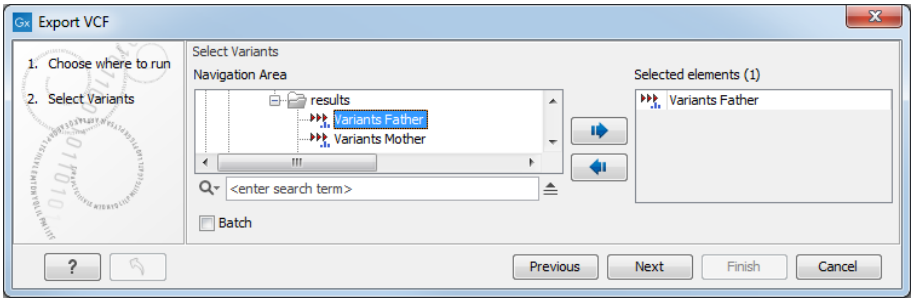


Figure 6.24: *The Select export dialog. Select the data element(s) to export.*

- Variants to VCF format: section 6.6.7
- Reports to JSON format: section 6.6.8
- Graphics to a range of formats: section 6.6.9
- Data element history: section 6.6.10

6.6.2 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 6.25.

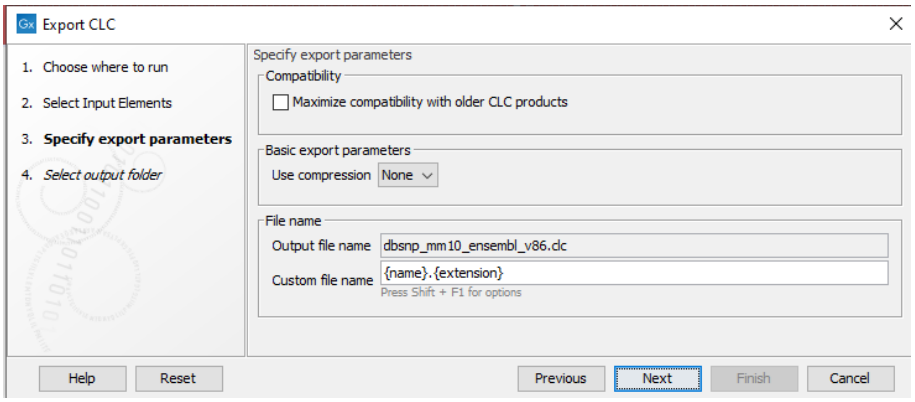


Figure 6.25: *Configure the export parameters. When exporting to CLC format, you can choose to maximize compatibility with older CLC products.*

Examples of configuration options:

- **Maximize compatibility with older CLC products** This is described in more detail in section 6.6.4.
- **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.

After configuration, choose where to save the exported files to.

6.6.3 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 6.26. 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 6.27.

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.

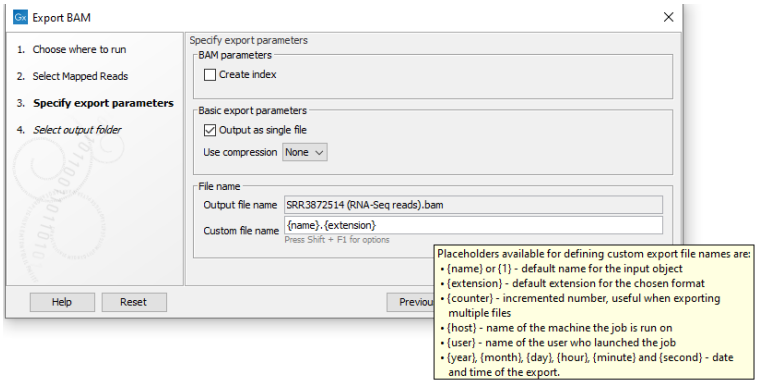


Figure 6.26: *The default placeholders, separate by a "." are being used here. The tooltip for the Custom file name field provides information about these and other available placeholders.*

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.
- **{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 11.3.4.

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.

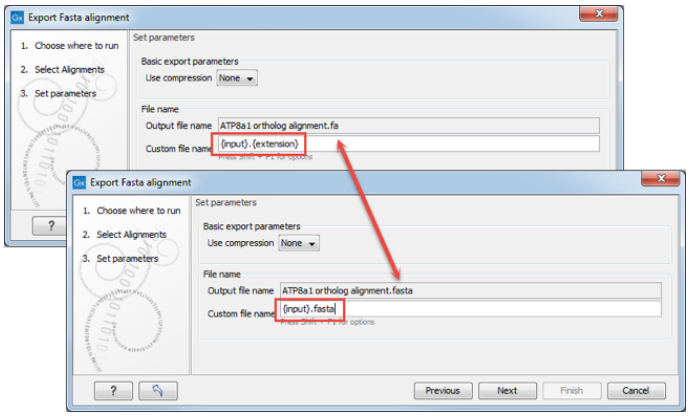


Figure 6.27: *The file name extension can be changed by typing in the preferred file name format.*

6.6.4 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 fomat, 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 may still wish to export to CLC format. A *CLC Genomics Workbench* can be run without a license in Viewing Mode, and CLC format data can be imported in the same way it would be using a licensed Workbench. Viewing Mode is described further in section 1.4.7.

Compatibility of the CLC data format between Workbench versions When exporting to CLC or zip format, an option called **Maximize compatibility with older CLC products** is presented at the **Specify export parameters** step, as can be seen in figure **??**. With this option checked, data will be exported without internal compression. Data exported with this option turned on may be larger than it would be otherwise.

Internal compression of CLC data was introduced in CLC Genomics Workbench 12.0, CLC Main Workbench 8.1 and CLC Genomics Server 11.0 using the LZ4 algorithm, a lossless compression method. This feature decreases the size of the data elements created by the software. However, CLC format files with internal compression are not compatible with older versions of the software. Thus, this option should be enabled when exporting data intended for use with older CLC software versions.

Internal compression is not required for compatibility with Workbench versions released after this feature was introduced. In other words, data generated in older Workbench versions can, in general, be imported into newer Workbench versions. We endeavor to maintain backwards compatibility of CLC format files whenever possible, so that most CLC format files made using an older version of a Workbenches can be imported into newer Workbench versions.

Internal data compression can be turned off, so no data created is internally compressed. How to do this is described in the Workbench Preferences documentation section 4.4.

6.6.5 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 6.6.4.

6.6.6 Export of tables

Tables can be exported in four different formats; CSV, tab-separated, Excel, or html.

When exporting a table in CSV, tab-separated, or Excel format, numbers with many decimals are printed in the exported file with 10 decimals, or in 1.123E-5 format when the number is close to zero.

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 export your data to several worksheets in batches smaller than 66,530 rows.

When exporting a table in html format, data are exported with the number of decimals that have been defined in the workbench preference settings. When tables are exported in html format from the server or using command line tools, the default number of exported decimals is 3.

The Excel exporters, the CSV and tab delimited exporters, and the HTML exporter have been extended with the ability to export only a sub-set of columns from the object being exported. Uncheck the option "Export all columns" and click next to see a new dialog window in which columns to be exported can be selected (figure 6.28).

You can choose to "Export the table as currently shown": This will export the table as shown in the active view, including any filtering, sorting, and dynamically added columns.

You can choose also choose which columns to export one by one, or choose a predefined subset of columns:

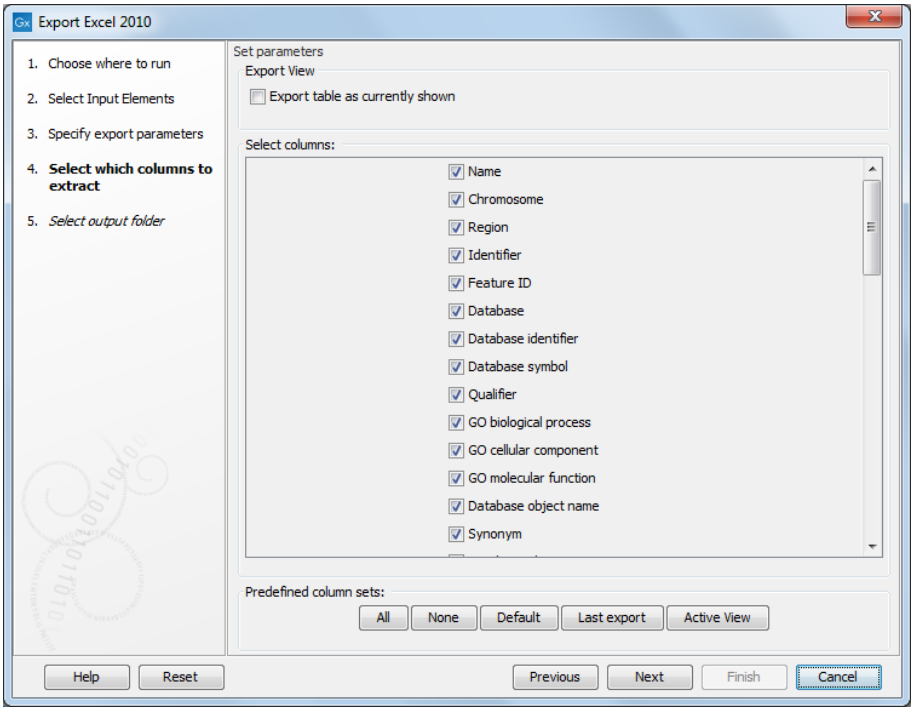


Figure 6.28: *Selecting columns to be exported.*

- All: will select all possible columns.
- None: will clear all preselected column.
- Default: will select the columns preselected by default by the software.
- Last export: will select all windows that were selected during the last export.
- Active view (only if a table is currently open): the columns exported are the same than the ones selected in the Side Panel of the table.

After selecting columns, the user will be directed to the output destination wizard page.

Note about decimals and Locale settings. When exporting to CSV and tab delimited files, decimal numbers are formatted according to the Locale setting of the Workbench (see section 4.1). If you open the CSV or tab delimited file with spreadsheet software like Excel, you should make sure that both the Workbench and the spreadsheet software are using the same Locale.

6.6.7 Export in VCF format

Using this tool, variants, CNV and fusion data are exported to a VCF 4.2 format file.

A number of configuration options are available (figure 6.29). 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.

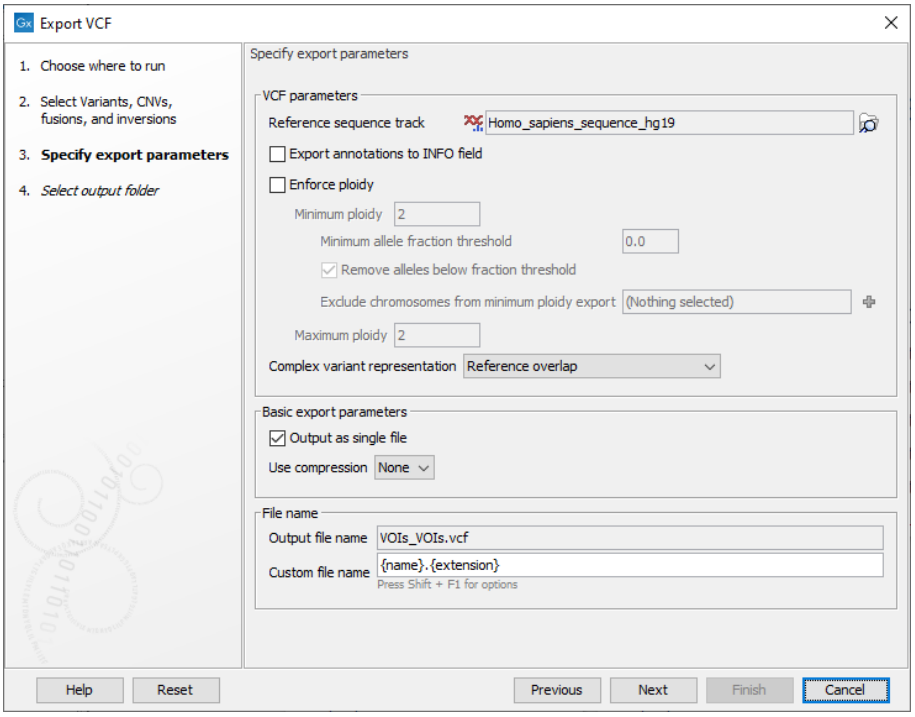


Figure 6.29: *Several options are available when exporting to a VCF format file.*

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 the Remove Information from Variants tool. 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 workbench, where complex overlap does not affect how variants are specified.

Read more about these options in section 6.6.7.

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 6.6.2 and section 6.6.3.

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 6.30:

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,

	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 field[1]	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 export[2]	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 6.30: *Main characteristics of the complex variant representations.*

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: frequency=AD/sum(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 6.31:



	Variants exported									Variants re-imported
	CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	sampleName	
Without overlap specification	1	4	.	GG	TT	200.00	.	.	GT:CLCAD2:DP	1:0,4:10	AA TT GGG
	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	AA TT GGG
	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	GG AA TT
	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	AA TT GGG
	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 6.31: *Example of export and import using the different complex variant representations.*

6.6.8 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 26.4) 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")
```

You can try out the JSON export of RNA-Seq reports and the above script with the data included in the tutorial Expression Analysis using RNA-Seq: http://resources.qiagenbioinformatics.com/tutorials/RNASeq-droso.pdf

6.6.9 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 6.7.
- 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:

- Sequences
- Alignments
- Read mappings
- 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 6.32.

The options available when exporting tracks and track lists to graphics format files are shown in figure 6.33.

The format and size of the exported graphics can be configured using:

- **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:

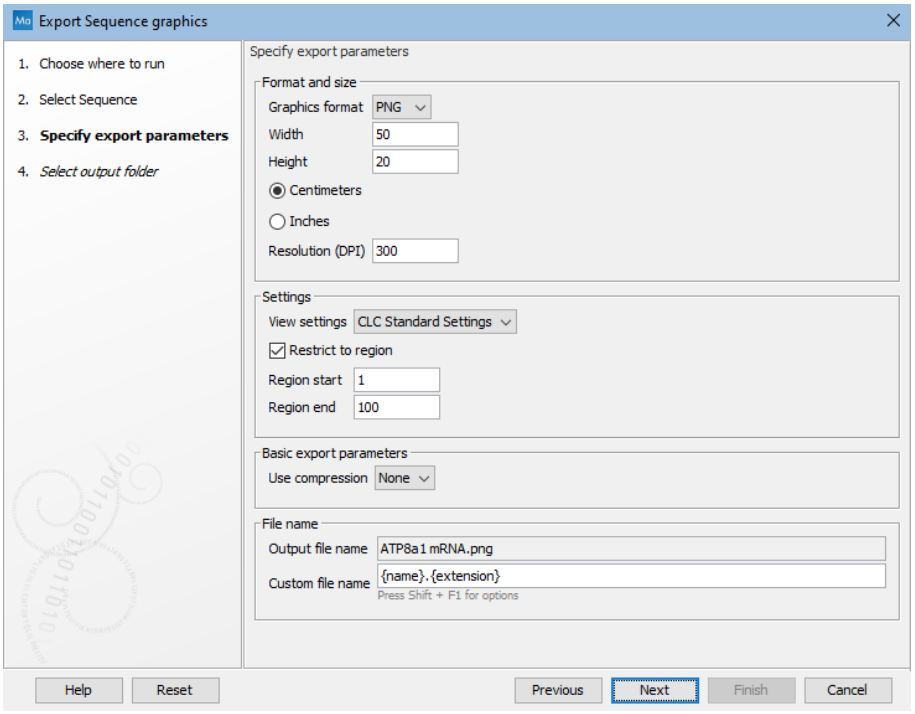


Figure 6.32: *Options available when exporting sequences, alignments and read mappings to graphics format files.*

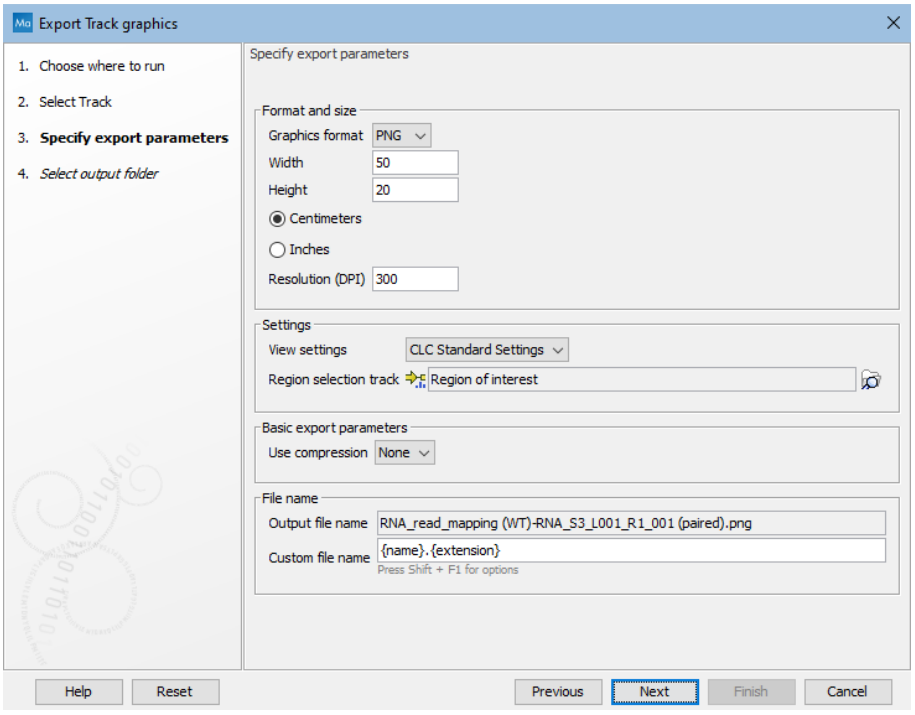


Figure 6.33: *Options available when exporting tracks and track lists to graphics format files.*

- **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.

6.6.10 Export history

Each data element in the Workbench has a history. The history information includes things 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 example, in the case of an alignment, one would see the sequence data used for that alignment listed. You can view this information for each data element by clicking on the Show History view ( ) at the bottom of the viewing area when a data element is open in the Workbench.

This history information can be exported to a pdf document or to a CSV file. To do this:

- (Optional, but preferred) Select the data element (like an alignment) in the **Navigation Area**.
- Start up the exporter tool via the Export button in the toolbar or using the **Export** option under the File menu.
- Select the **History PDF** or History CSV as the format to export to (figure 6.34).
- Select the data to export, or confirm the data to export if it was already selected via the **Navigation Area**.
- Edit any parameters of interest, such as the Page Setup details, the output filename(s) and whether or not compression should be applied (figure 6.35).
- Select where the data should be exported to.
- Click **Finish**.

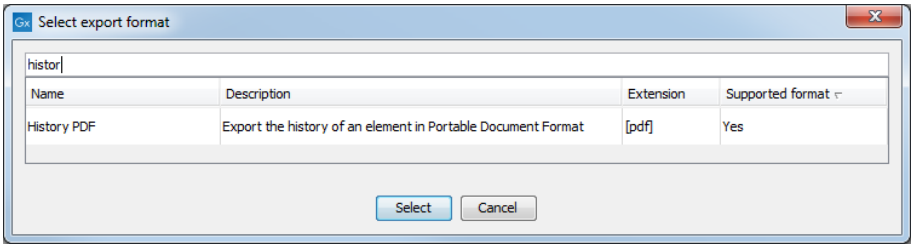


Figure 6.34: *Select "History PDF" for exporting the history of an element as a PDF file.*

6.6.11 Backing up data from the CLC Workbench

Regular backups of your data are advisable.

The data stored in your CLC Workbench is in the areas defined as CLC Data Locations. Whole data locations can be backed up directly (option 1) or, for smaller amounts of data, you could export the selected data elements to a zip file (option 2).

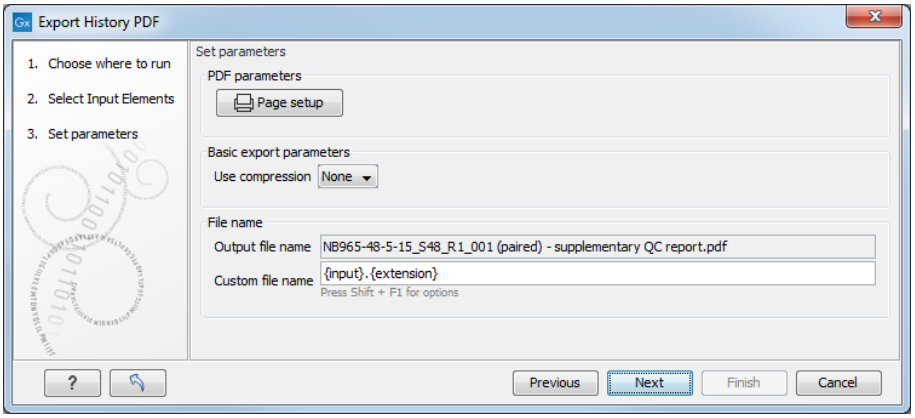


Figure 6.35: *When exporting the history in PDF, it is possible to adjust the page setup.*

Option 1: Backing up each CLC Data Location

The easiest way for most people to find out where their data is stored is to put the mouse cursor over the top level directories, that is, the ones that have an icon like (), in the **Navigation Area** of the Workbench. This brings up a tool tip with the system location for that data location.

To back up all your CLC data, please ensure that all your CLC Data Locations are backed up.

Here, if you needed to recover the data later, you could put add the data folder from backup as a data location in your Workbench. If the original data location is not present, then the data should be usable directly. If the original data location is still present, the Workbench will re-index the (new) data location. For large volumes of data, re-indexing can take some time.

Information about your data locations can also be found in an xml file called model_settings_300.xml This file is 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 Deployment Manual: http://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, certain results files or a whole data location, where that location contains smaller amounts of data. For data that takes up many gigabases of space, this method can be used, but it can be very demanding on space, as well as time.

Select the data items, including any folders, in the Navigation area of your Workbench and choose to export by going to:

File | **Export ()**

and choosing zip format.

The zip file created will contain all the data you selected.

A note about compatibility 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 backing up data that may be used in software versions older than these, then please select the export option **Maximize compatibility with older CLC products**. Further information about this is provided in section 6.6.4

You can import the zip file into a Workbench by going to:

File | Import ( ) | Standard Import

and selecting "Automatic import" in the Options area.

The only data files associated with the *CLC Genomics Workbench* not within a specified data location are BLAST databases. It is unusual to back up BLAST databases as they are usually updated relatively frequently and in many cases can be easily re-created from the original files or re-downloaded from public resources. If you do wish to backup your BLAST database files, they can be found in the folders specified in the BLAST Database Manager, which is started by going to: **Toolbox | BLAST ( ) | Manage BLAST Databases ( )** .

6.7 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 6.6.9.

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 6.36). For 3D structures, the section visible will always be exported, i.e. the equivalent to selecting to export just the visible area.

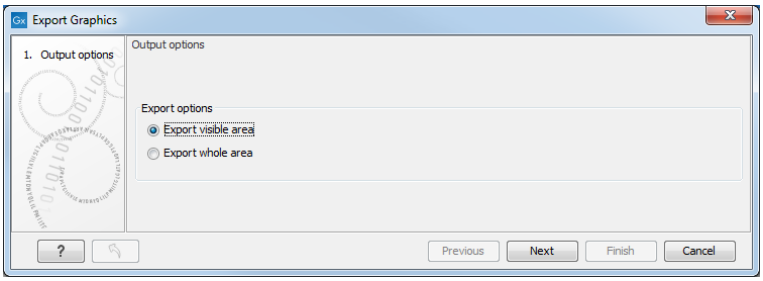


Figure 6.36: *The whole view or just the visible area can be selected for export.*

A view of a circular sequence, zoomed in so that you can only see a part of it, is shown in figure 6.37.

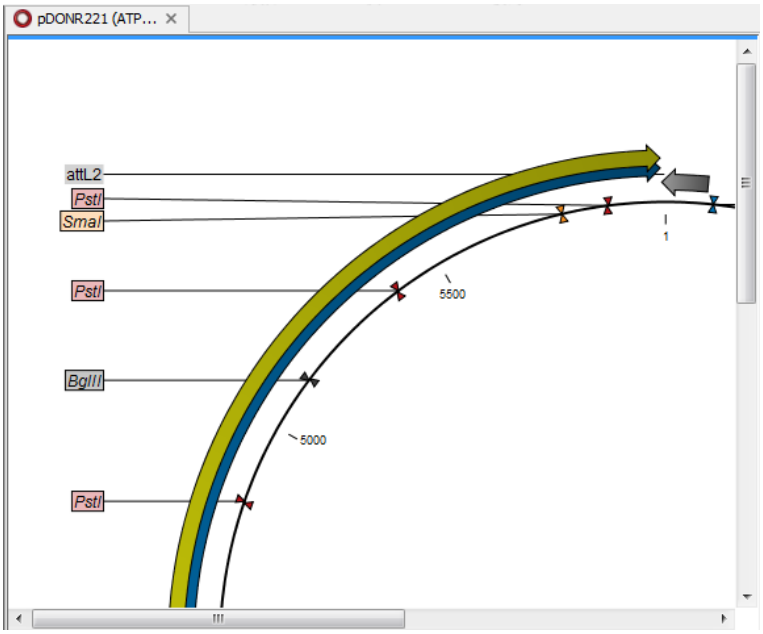


Figure 6.37: *A circular sequence, as it looks on the screen when zoomed in.*

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 6.37 is shown in figure 6.38.

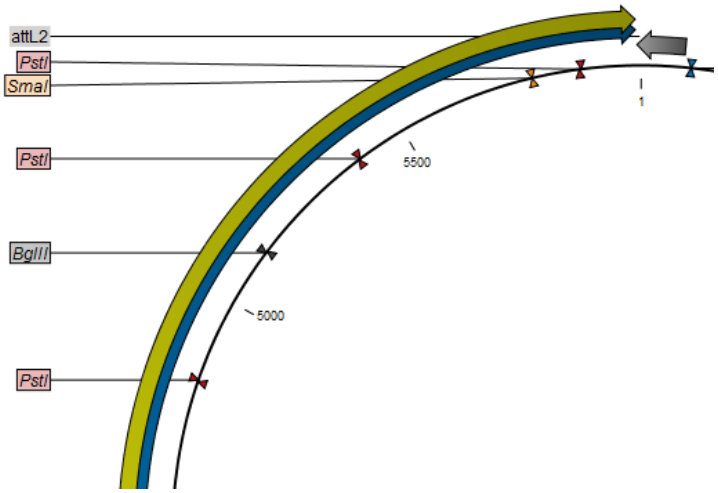


Figure 6.38: *The exported graphics file when Export visible area was selected.*

If you select **Export whole view**, the result would look like that shown in figure 6.39, where the part of the sequence not visible on screen is also exported.

After choosing a name and location to save the output, you may have the option to click on the **Next** or the **Finish** button. Clicking on **Next** will show further information, and options, for the export. If you click on **Finish** at this point, the file will be exported using the settings used the last time this functionality was run, or using default settings.

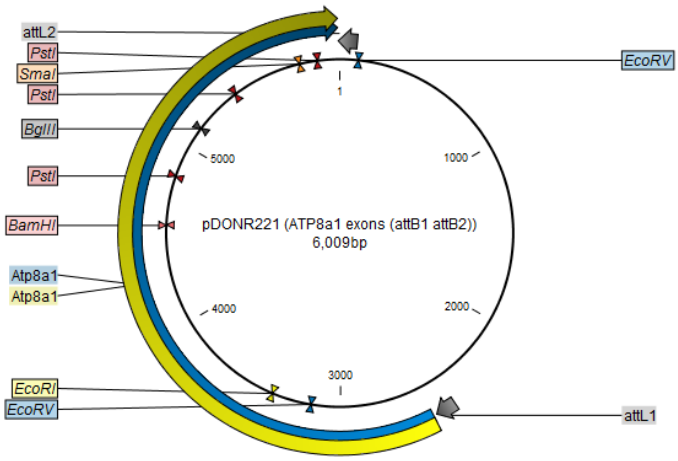


Figure 6.39: *The exported graphics file when Export whole view was selected. The whole sequence is shown, not just the part visible on screen when the view was exported.*

6.7.1 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.

Parameters for bitmap formats For bitmap files, clicking **Next** will display the dialog shown in figure 6.40.

You can adjust the size (the resolution) of the file to four standard sizes:

- Screen resolution
- Low resolution
- Medium resolution

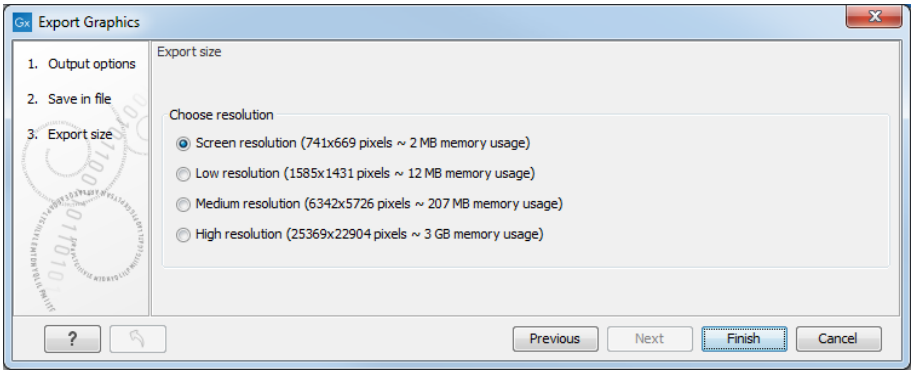


Figure 6.40: *Parameters for bitmap formats: size of the graphics file.*

- 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 6.1.1 for more about importing external files into *CLC Genomics Workbench*.

Parameters for vector formats For PDF format, the dialog shown in figure 6.41 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).

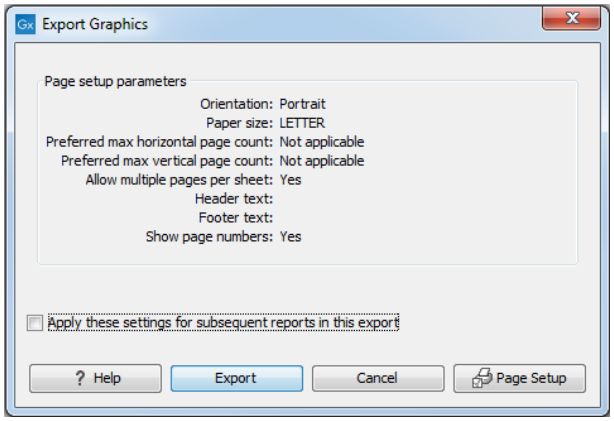


Figure 6.41: *Page setup parameters for vector formats.*

The settings for the page setup are shown. Clicking the **Page Setup** button will display a dialog where these settings can ba 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.

6.8 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 6.42, showing the conservation score of reads in a read mapping.

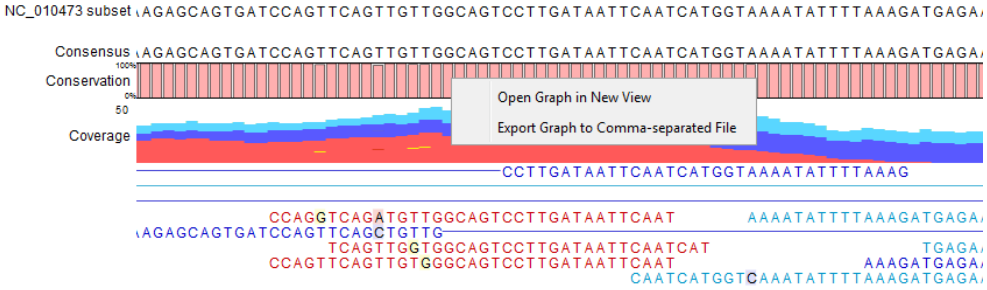


Figure 6.42: *A conservation graph displayed along mapped reads. Right-click the graph to export the data points to a file.*

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 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 6.43 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.

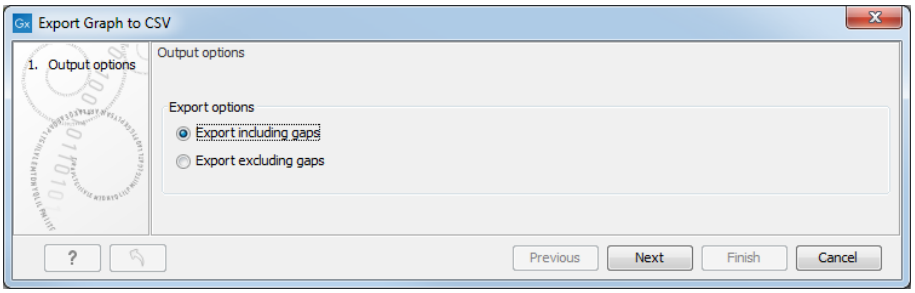


Figure 6.43: *Choosing to include data points with gaps*

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";
...
```

6.9 CLC Server data import and export

Data export from a CLC Server

When the Workbench is connected to a *CLC Server*, data held on the *CLC Server* can be exported to the system a CLC Workbench is running on, or it can be exported to an area the *CLC Server* has been configured to have access to. Such areas are called "Import/export" directories and these must be configured by your server administrator.

To export data to a place the CLC Workbench has access to, choose to run the export task on the Workbench. To export data to an "Import/Export" directory, choose to run the export task on the CLC Server, or Grid, as is appropriate for your setup.

Data import to a CLC Server

When connected to a *CLC Server*, data can be imported from "Import/export" directories that have been configured for the *CLC Server*. On some systems, data can also be imported from areas available to your CLC Workbench. When that is allowed, you will be able to choose where the files to be imported from are, either "File system" or "<servername> (CLC Genomics Server)", as shown in figure 6.44. The former refers to files available to the CLC Workbench, and the latter to files available to the *CLC Server*.

Not all server setups are configured to allow import from local disks, in which case, at least one "Import/export" directory will need to be configured by your server administrator to support import to the *CLC Server*.

If you choose the option "File system" 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. Further details about this can be found in section 2.4.

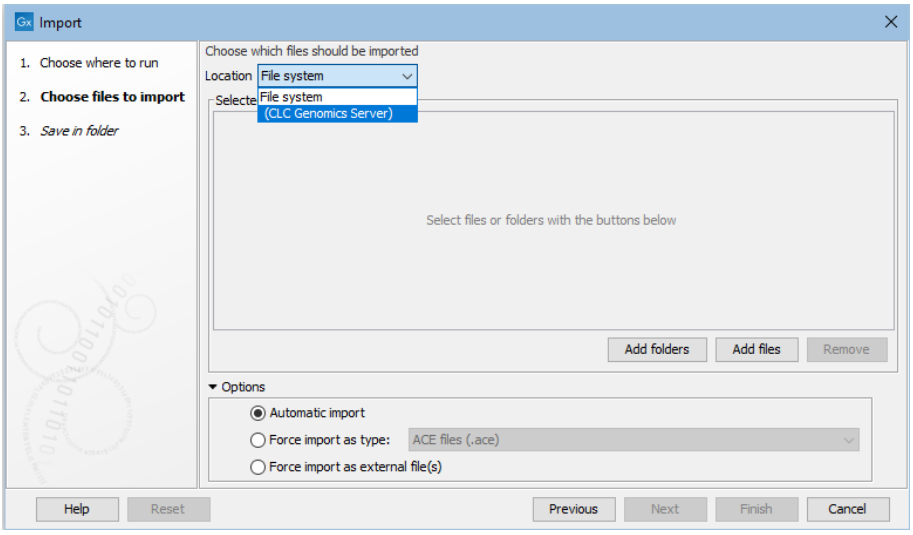


Figure 6.44: *Importing data using a Server.*

6.10 Copy/paste view output

The content of tables (reports, folder lists, and sequence lists) can be copy/pasted into different programs, where it can be edited. *CLC Genomics Workbench* pastes the data in tabulator separated format in various programs in which the copy/paste can be applied. For simplicity, we include one example of the copy/paste function from a **Folder Content** view to Microsoft Excel.

Right click a folder in the Navigation Area and chooses **Show** | **Content**. The different elements saved in that folder are now listed in a table in the View Area. Select one or more of these elements and use the Ctrl + C (or ⌘ + C) command to copy the selected items.

See figure 6.45.

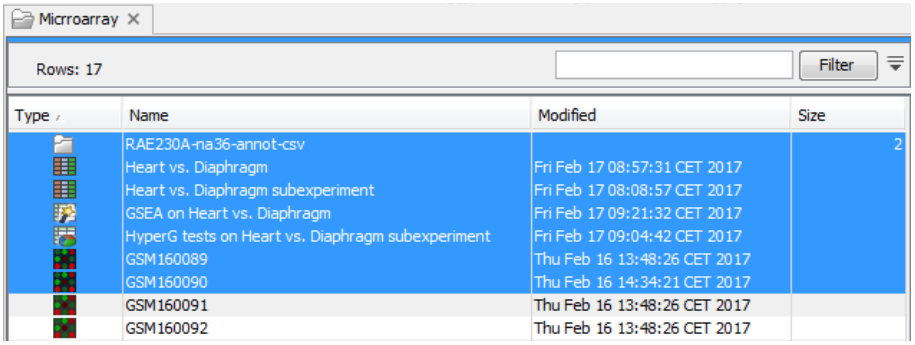


Figure 6.45: *Selected elements in a Folder Content view.*

Then, in a new Excel document, right-click in the cell A1 and paste the items previously copied.

The outcome might appear unorganized, but with a few operations the structure of the view in *CLC Genomics Workbench* can be produced. (Except the icons which are replaced by file references in Excel.)

Note that all tables can also be **Exported** ( ) directly in Excel format.

Chapter 7

Data download

Contents

7.1 Search for Sequences at NCBI 159
7.1.1 NCBI search options 160
7.1.2 Handling of NCBI search results 161
7.2 Search for PDB Structures at NCBI 162
7.2.1 Structure search options 163
7.2.2 Handling of NCBI structure search results 163
7.2.3 Save structure search parameters 165
7.3 Search for Sequences in UniProt (Swiss-Prot/TrEMBL) 165
7.3.1 UniProt search options 165
7.3.2 Handling of UniProt search results 167
7.3.3 Save UniProt search parameters 168
7.4 SRA search 168
7.4.1 SRA search options 169
7.4.2 SRA search output 170
7.4.3 Downloading reads and metadata from SRA 170
7.4.4 How reads are downloaded 172
7.5 Sequence web info 173

CLC Genomics Workbench offers different ways of searching and downloading online data. You must be online when initiating and performing the following searches.

7.1 Search for Sequences at NCBI

This section describes searches for sequences in GenBank - the **NCBI** database using Entrez (see https://www.ncbi.nlm.nih.gov/books/NBK3837/)

Download | Search for Sequences at NCBI () or Ctrl + B (⌘ + B on Mac)

This opens the following view (figure 7.1).

The search view can be saved either using dragging the search tab and and dropping it in the **Navigation Area** or by clicking **Save** (). When saving the search, only the parameters are saved

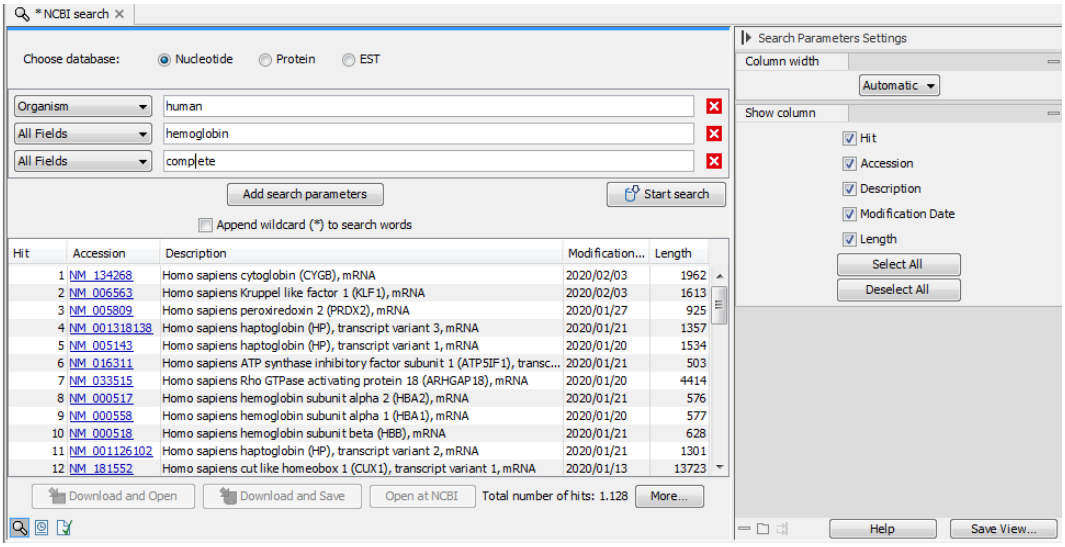


Figure 7.1: *The GenBank search view.*

- not the results of the search. This is useful if you have a special search that you perform from time to time.

Even if you don't save the search, the next time you open the search view, it will remember the parameters from the last time you did a search.

7.1.1 NCBI search options

Conducting a search in the **NCBI Database** from *CLC Genomics Workbench* corresponds to conducting the search on NCBI's website, while having the results available and ready to work with straight away.

You can choose whether you want to search for nucleotide sequences, protein sequences or EST databases.

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. 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.

The following parameters can be added to the search:

- **All fields** Searches in all parameters in the NCBI database at the same time. It also provide an opportunity to search to parameters which are not listed in the dialog (e.g., `CD9 NOT homo sapiens`).
- **Organism**
- **Definition/Title**
- **Modified Since** Choose one option from the drop-down menu, between 30 days and 10 years.
- **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**

You can append a wildcard character by checking the checkbox at the bottom. This means that you only have to enter the first part of the search text, e.g., searching for "genom" will find both "genomic" and "genome".

A "Feature Key" option is available in GenBank when searching for nucleotide sequences: writing `gene[Feature key] AND mouse` will generate hits for one or more genes and where 'mouse' appears somewhere in GenBank file. For more information about how to use this syntax, see http://www.ncbi.nlm.nih.gov/books/NBK3837/

When you are satisfied with the parameters you have entered, click **Start search**. 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.

7.1.2 Handling of NCBI search results

The search result is presented as a list of entries found in the NCBI database. The **View** displays 50 hits at a time by default. This can be changed in the **Preferences** (see chapter 4). Click on the **More...** button at the bottom right of the **View** to load more results.

The five columns in the table contain the following information:

- **Hit** The rank of the sequence found in the search results
- **Accession** The accession identifier for that entry. Clicking on the link opens the entry's page at the NCBI in a web browser.
- **Description** The text from 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

It is possible to exclude one or more of these columns by deselecting them in the "Show column" tab of the settings panel on the right hand side. See section 4.6 for further information about working with view preferences.

Double-clicking on a row will download and open that sequence in a view.

Alternatively, select one or more rows of the table, and then use the buttons at the bottom of the search view to:

- **Download and Open** Sequences will be opened in a new view after download is complete.
- **Download and Save** Sequences will be saved to the location you choose after they are downloaded.

- **Open at NCBI** The sequence entry page(s) at the NCBI will be opened in a web browser.

These options are also available in the menu that appears if you right-click over selected rows.

You can also drag selected rows to a tab area to download and open them in a new tab. Sequences can also be downloaded and saved by selecting rows, copying them (e.g. using Ctrl-C), selecting a folder in the **Navigation Area** and then pasting (e.g. using Ctrl-V).

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 2.4.

7.2 Search for PDB Structures at NCBI

This section describes searches for three dimensional structures from the NCBI structure database `http://www.ncbi.nlm.nih.gov/Structure/MMDB/mmdb.shtml`. For manipulating and visualization of the downloaded structures see section 14.2.

The NCBI search view is opened in this way:

Download | Search for PBD Structures at NCBI ( **)**

or **Ctrl + B (⌘ + B on Mac)**

This opens the view shown in figure 7.2:

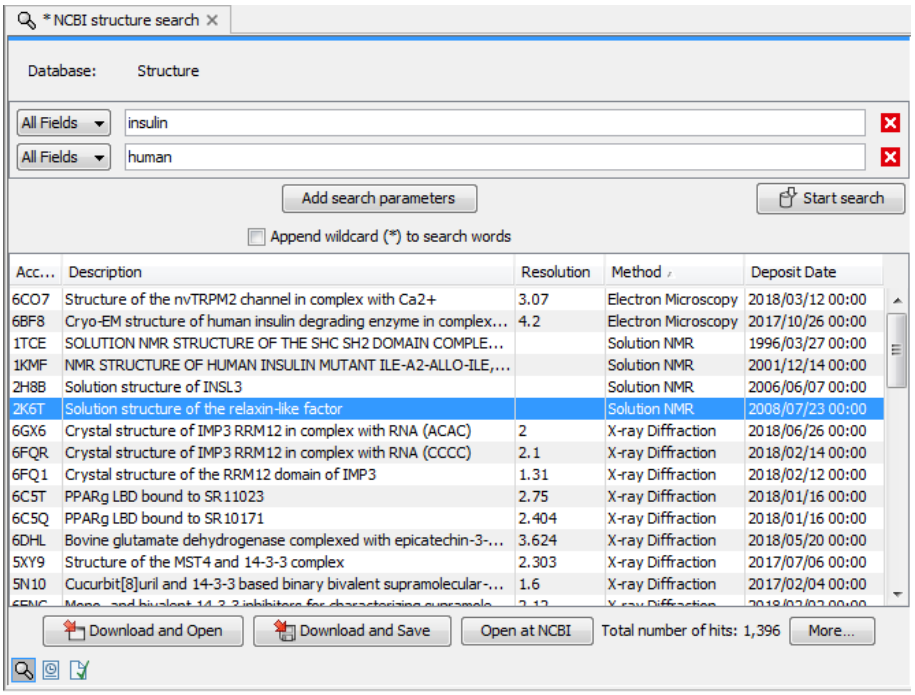


Figure 7.2: *The structure search view.*

7.2.1 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.

7.2.2 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 7.3). 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.

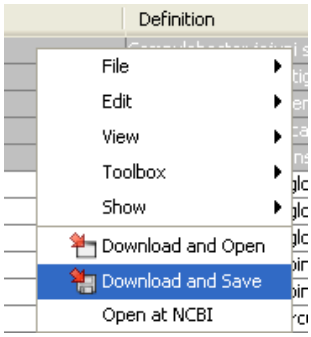


Figure 7.3: *By right-clicking a search result, it is possible to choose how to handle the relevant structure.*

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 **Toolbox** in the **Processes** tab.

7.2.3 Save structure search parameters

The search view can be saved either using dragging the search tab and and dropping it in the **Navigation Area** or by clicking **Save** ( ). When saving the search, only the parameters are saved - not the results of the search. This is useful if you have a special search that you perform from time to time.

Even if you don't save the search, the next time you open the search view, it will remember the parameters from the last time you did a search.

7.3 Search for Sequences in UniProt (Swiss-Prot/TrEMBL)

This section describes searches in UniProt and the handling of search results. UniProt is a global database of protein sequences.

The UniProt search view (figure 7.4) is opened in this way:

> **Download** | **Search for Sequences in UniProt ( )**

7.3.1 UniProt search options

Conducting a search in **UniProt** from *CLC Genomics Workbench* corresponds to conducting the search on UniProt's website. When conducting the search from *CLC Genomics Workbench*, the results are available and ready to work with straight away.

Above the search fields, you can choose which database to search:

- **Swiss-Prot** This is believed to be the most accurate and best quality protein database available. All entries in the database has been currated manually and data are entered

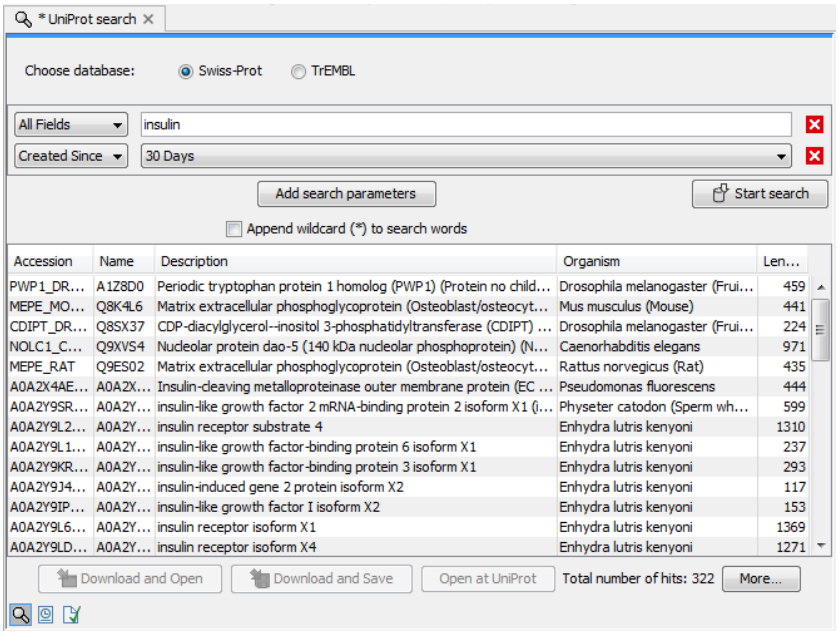


Figure 7.4: *The UniProt search view.*

according to the original research paper.

- **TrEMBL** This database contain computer annotated protein sequences, thus the quality of the annotations is not as good as the Swiss-Prot database.

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 checking the checkbox at the bottom. This means that you only have to enter the first part of the search text, e.g. searching for "genom" will find both "genomic" and "genome".

The following parameters can be added to the search:

- **All fields**. Text, searches in all parameters in the UniProt database at the same time.
- **Organism**. Text.
- **Description**. Text.
- **Created Since**. Between 30 days and 10 years.
- **Feature**. Text.

The search parameters listed in the dialog are the most recently used. The **All fields** allows searches in all parameters in the UniProt database at the same time.

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 UniProt database. This ensures a much faster search.

7.3.2 Handling of UniProt search results

The search result is presented as a list of links to the files in the UniProt 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**. More hits can be displayed by clicking the **More...** button at the bottom left of the **View**.

Each sequence hit is represented by text in three columns:

- Accession
- Name
- Description
- Organism
- Length.

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 sequences can be selected, and by clicking the buttons in the bottom of the search view, you can do the following:

- Download and open, does not save the sequence.
- Download and save, lets you choose location for saving sequence.
- Open at UniProt, searches the sequence at UniProt's web page.

Double-clicking a hit will download and open the sequence. 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 UniProt search results

The sequences from the search results can be opened by dragging them into a position in the **View Area**.

Note! A sequence is not saved until the **View** displaying the sequence is closed. When that happens, a dialog opens: Save changes of sequence x? (Yes or No).

The sequence can also be saved by dragging it into the **Navigation Area**. It is possible to select more sequences and drag all of them into the **Navigation Area** at the same time.

Download UniProt search results using right-click menu

You may also select one or more sequences from the list and download using the right-click menu. Choosing **Download and Save** lets you select a folder or location where the sequences are saved when they are downloaded. Choosing **Download and Open** opens a new view for each of the selected sequences.

Copy/paste from UniProt search results

When using copy/paste to bring the search results into the **Navigation Area**, the actual files are downloaded from UniProt.

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 **Toolbox** under the **Processes** tab) it is possible to continue other tasks in the program. Like the search process, the download process can be stopped, paused, and resumed.

7.3.3 Save UniProt search parameters

The search view can be saved either using dragging the search tab and and dropping it in the **Navigation Area** or by clicking **Save** ( ). When saving the search, only the parameters are saved - not the results of the search. This is useful if you have a special search that you perform from time to time.

Even if you don't save the search, the next time you open the search view, it will remember the parameters from the last time you did a search.

7.4 SRA search

This section describes searches in SRA and the handling of search results. SRA is an NCBI maintained database of NGS data.

The SRA search view (figure 7.5) is opened in this way:

> **Download | Search for Reads in SRA ( )**

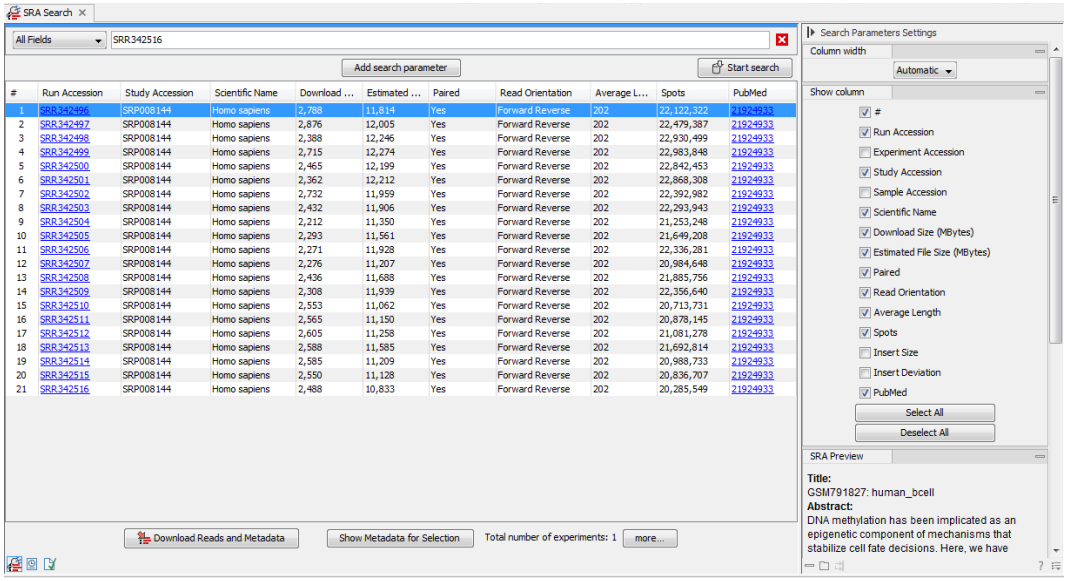


Figure 7.5: *The SRA search view.*

The tool queries the database with SRA accession number (see in figure 7.5) or various entries such as "hindgut" or "genometrakr". It is also possible to look for entries with certain properties, and to form more refined queries such as "paired-end RNA-Seq data from an Illumina HiSeq2500".

7.4.1 SRA search options

Search parameters Different types of search are available from a drop-down menu on the left hand side of the search bar.

Some special types are:

- **Modification/Publication date** This allows the narrowing of results to a particular range of dates specified as [MM] [YYYY]. For example 08 2016 to 08 2016 returns results published from the first day in August 2016 to the last day in August 2016.
- **Strategy** Provides an additional drop-down list of types of experiments e.g., RNA-Seq, ChIP-Seq, etc.
- **Library Selection** Provides an additional drop-down list of known library preparation methods, e.g. Poly(A) and Size fractionation.
- **Platform** Provides an additional drop-down list of NGS sequencing platforms e.g., Illumina, Ion Torrent. Note that download of data from some platforms such as Complete Genomics is not supported.
- **Instrument** Provides an additional drop-down list of individual NGS sequencing machines e.g., HiSeq X Ten, Ion Torrent PGM.
- **Paired Status** Choose between paired end and single end runs.
- **Availability** Choose between dbGaP or public. dbGaP refers to confidential data that can be searched through the tool and accessed upon request at NCBI.
- **PubMed** Choose between "has abstract" or "has full-text article" to find results that have a PubMed abstract or entire publication available.

Any number of search parameters may be added to refine a search, for example to construct a query for "paired-end RNA-Seq data from an Illumina HiSeq2500", but the return search must satisfy every search parameter. In other words, search parameters work as "A and B" and not as "A or B". In consequence, searches performed with two platform parameters for example "Platform = Illumina and Platform = Ion Torrent" will not return any results.

Note also that searches rely on metadata provided by the depositor of the SRA runs. This means that if, for example, an RNA-Seq run was not annotated as being RNA-Seq during submission, it won't be returned by a search for "Strategy = RNA-Seq".

Finally, the search tool uses NCBI's e-utilities, which occasionally experience downtime. If no searches return any results, check https://www.ncbi.nlm.nih.gov/sra/ to see the status of the service.

7.4.2 SRA search output

The search results are displayed with one run per line. Each Run Accession is a hyperlink to the NCBI webpage for the run, where additional information may be found, such as the distribution of nucleotide quality scores and links to external resources. On the right hand side of the search table, a "SRA Preview" panel shows the title and abstract associated to the selected run when available.

When looking for a specific run using the run SRA accession number, the tool will output the run that was searched and may also list additional runs that were submitted as part of the same experiment. In any case, it is safest to perform a new search specifically for the study to make sure the tool retrieves all possible runs. Right-click on a row to get a list of possible searches based on the selected run (figure 7.6), for example searching for more runs from the same sample, experiment or organism.

By default, the tool will output a maximum of 50 runs. The "more..." button below the table retrieves additional search results when the search exceeds 50 runs. The number of additional results returned can be controlled in Edit | Preferences | General | Number of hits (NCBI/Uniprot).

The "Total number of experiments" at the bottom of the search table reports how many experiments were retrieved, and not how many runs are listed in the table.

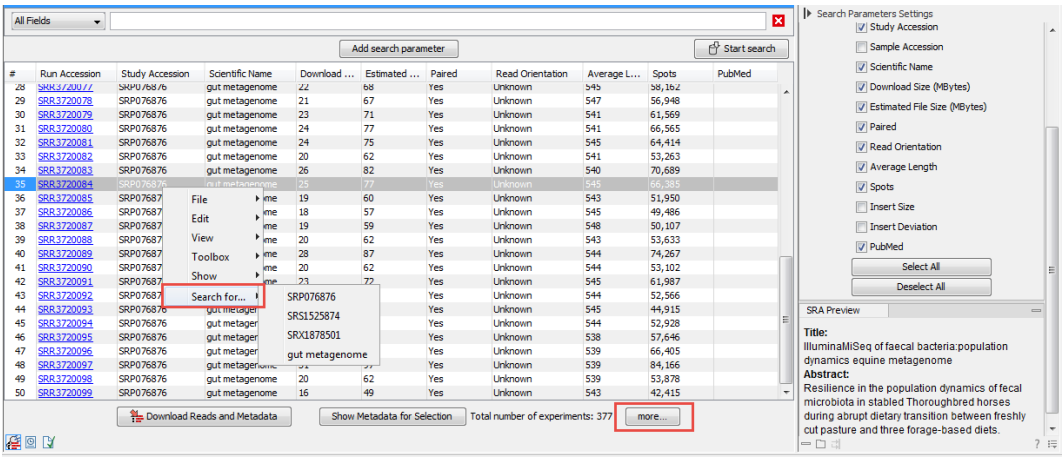


Figure 7.6: *The SRA search result table.*

At the bottom of the table, the button **"Show Metadata for Selection"** will create a metadata table containing sample specific information for the selected run(s). The first columns of the metadata table contain the same database identifiers as in the Search Table. The last columns recover sample details associated with the biosample. The list of all metadata available for the selected run is located in the right hand side panel of the table. This functionality is especially useful when selecting external data for use in a meta-analysis. For example, if your analysis is for a particular cancer type, and controls for the subject's age and gender, the **"Show Metadata for Selection"** facilitates the search for other datasets where the same metadata is available.

7.4.3 Downloading reads and metadata from SRA

Click on **Download Reads and Metadata** to save reads and their associated data. The data is saved in a metadata table and can be later associated to the reads for use in downstream analysis, for example to define factors for differential expression in the Differential Expression

for RNA-Seq tool. Should the metadata table later be deleted, the "Show Metadata for Selection" button can be used to quickly recover a copy without having to re-download all the runs.

The Download Reads and Metadata wizard offers the following options:

Import Options (figure 7.7)

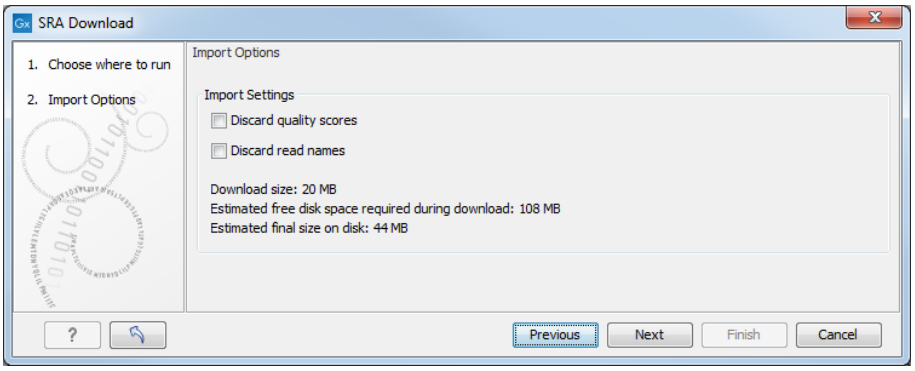


Figure 7.7: *The Download Reads and Metadata Import Options dialog.*

As with other NGS reads importers, it is possible to discard read names and/or quality scores to save space.

- "Download size" is the size of the .sra files that will be downloaded. Note that in some cases, the actual download may be up to 1GB larger than stated, as .sra files can be reference-compressed, meaning that a copy of the genome must also be retrieved before the file can be converted into fastq and imported into the workbench.
- "Estimated free disk space required during download" is a conservative estimate for the total free disk space required to download the selected runs. This is the "Estimated final size on disk" + the size of the largest single run in FASTQ format + the size of the largest single run in SRA format.
- "Estimated final size on disk" is an estimate of the total size of the files after they have been imported into the workbench.

Edit Paired End Settings (figure 7.8)

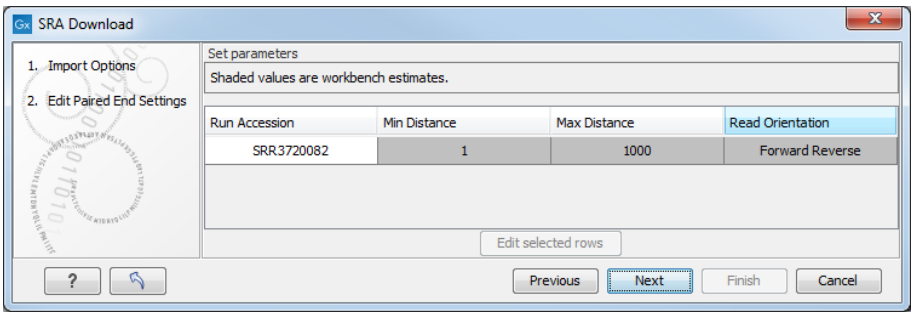


Figure 7.8: *The Download Reads and Metadata Edit Paired End Settings dialog.*

This dialog appears for all runs marked as being Paired (Paired column contains "Yes").

Read orientation is always guessed to be "Forward Reverse" unless otherwise stated.

Minimum distance and **Maximum distance** depend on how much data the depositor supplied with the runs. They are allowed to supply an "Insert Size" and an "Insert Deviation".

- If no insert size is supplied, we use defaults of 1 for minimum and 1,000 for maximum.
- If an insert size is supplied, we make the following calculation:
  - $Mindistance = insertsize - 5 * insertdeviation$
  - $Maxdistance = insertsize + 5 * insertdeviation$
- If no deviation is supplied, we estimate this to be $0.1 * insertsize$ and perform the same calculation as above.

When possible, we generally recommend that SRA data be used in subsequent analyses with the "Auto paired end distance detection" option enabled as the quality of deposited information is low. For example, some depositors report insert size including the length of the reads, and some excluding the length of the reads.

7.4.4 How reads are downloaded

SRA reads are downloaded in the ".sra" format using the NCBI SRA-toolkit. A .sra file is typically 2.5x smaller than an equivalent zipped fastq file. Download uses the NCBI 'prefetch' utility, and the resulting file is read into the workbench using 'fastq-dump'.

Sometimes runs in SRA cannot be downloaded. The affected runs are listed in a Problems panel together with a description of the problem. It is still possible to download 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: http://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.

We support download of reads via the commercial FASP protocol from Aspera. In our testing, Aspera download is up to 10x faster than a normal http download.

To enable this functionality, you have to download the Aspera Connect software from http://downloads.asperasoft.com/connect2/.

On Windows, choose to do a "Custom" install and choose to "Install for all users of this machine". You can then test if the installation worked by downloading a small file. The log (accessible from the Processes tab) will include the line "Downloading via fasp" if everything worked.

It is possible to change the Aspera options using **Preferences | Advanced | SRA Download**.

The following options are available:

- **Use Aspera when available** Per default, Aspera is automatically used if installed. This option makes it possible to disable Aspera.
- **Limit Aspera download speeds to [ ] Mb/s (Mac and Linux only)** Using Aspera may take up a lot of network resources. Use this option to specify a maximum download speed (in megabit per second). Note that this option is only available on Mac and Linux. For Windows users, it is possible to limit the maximum download speed by modifying the `aspera.conf file`, which can be found in `C:\Program Files (x86)\Aspera\Aspera Connect\etc`. See http://download.asperasoft.com/download/docs/csrv/3.3.4/linux/html/index.html and http://download.asperasoft.com/download/docs/csrv/3.3.4/linux/html/fasp/setting-global-bandwidth.html for more details.

7.5 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 12.5). 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 7.9):

Open a sequence or a sequence list | Right-click the name of the sequence | Web Info ( ) | select the desired search function

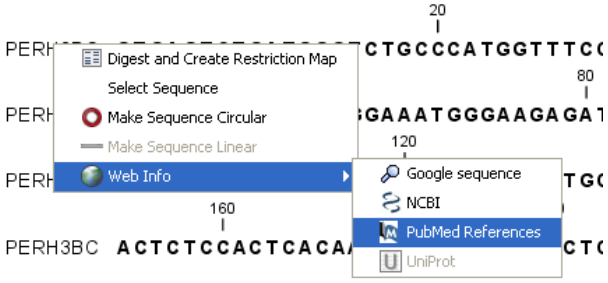


Figure 7.9: *Open webpages with information about this sequence.*

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 http://www.google.com. The resulting web page is equivalent to typing the accession number of the sequence into the search field on http://www.google.com.

NCBI The NCBI search function searches in GenBank at NCBI (http://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 se a dialog and the browser will not open.

UniProt The UniProt search function searches in the UniProt database (http://www.ebi.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 8

References management

Contents

8.1 Download Genomes 178
8.2 QIAGEN Sets 179
8.3 Custom Sets 181
8.3.1 Copy to References 184
8.3.2 Export a Custom Data Set 184
8.3.3 Import a Custom Data Set 185
8.4 Imported Data 186
8.4.1 Exporting reference data outside of the Reference Data Manager framework 187

The Reference Data Manager (figure 8.1) offers an easy way of retrieving popular reference data sources such as genes, variant annotations and genome sequences as tracks.

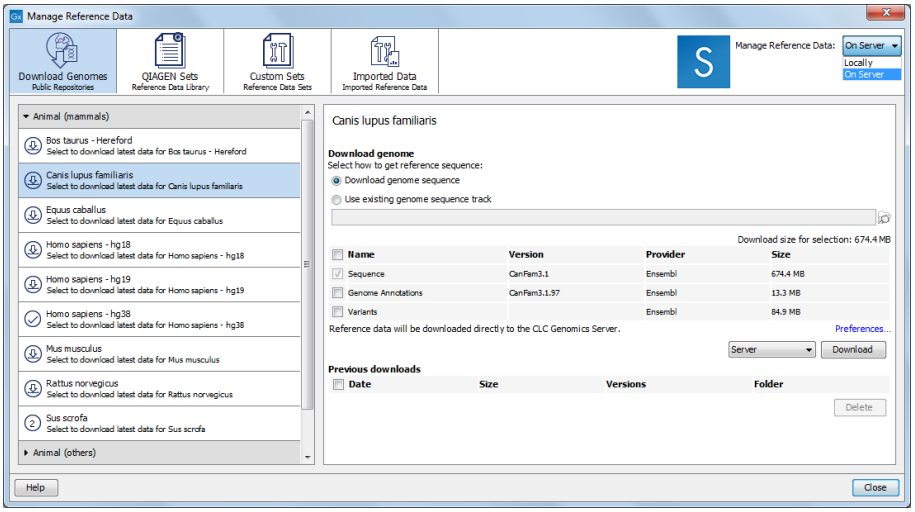


Figure 8.1: *Click on the References button to open the Reference Data Manager. Here, public reference data and QIAGEN Sets can be downloaded, and custom data sets can be imported and configured.*

The total size of the reference data is indicated when selecting the elements to download. The amount of time it will take to download this data depends on your network connection, but can take several hours on slower connections.

Where reference data is downloaded from

Data download using the **Download Genomes** comes from public repositories such as Ensembl, NCBI, UCSC. This type of data is not provided by nor hosted by QIAGEN. The list of organisms is dynamically updated.

The QIAGEN Sets tab allows you to download curated Reference Data Sets directly from a QIAGEN reference data repository. The location of the repository can be changed within the Preferences of the Workbench. This is only relevant if your site is hosting a mirror of this area.

Downloading reference data

By default, data downloaded using the Reference Data Manager is stored in a folder in your home area called **CLC_References**. If such a folder does not already exist, it will be created and added as a Workbench File Location automatically when you first start up the *CLC Genomics Workbench*.

In the top right hand side of the Reference Data Manager, the option "Locally" next to "Manage Reference Data" indicates that data will be downloaded to the *CLC Genomics Workbench*. The amount of free space available is reported just below this (figure 8.2).

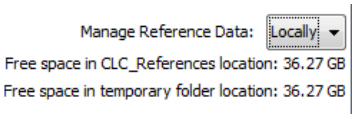


Figure 8.2: *Reference data is downloaded to the CLC_References area of the Workbench when the "Manage Reference Data" option is set to "Locally".*

Downloading data to a CLC_References area on a server

If you are logged into a *CLC Genomics Server* that has been configured with a File System Location called CLC_References, then the "Manage Reference Data" drop-down menu in the top right corner of the Reference Data Manager will show the option "On Server". If this is selected, reference data will be downloaded to the CLC_References area on the server.

If you have chosen "On Server" and your *CLC Genomics Server* is set up to send jobs to grid nodes, you can choose which grid preset to use for downloading data under the **Download Genomes** tab via a drop-down menu to the left of the **Download** button (figure 8.3).

By default, data will be downloaded directly using the CLC_References location on the server. Downloads can be configured to go via the Workbench using settings within the Workbench Preferences. This can be useful if the *CLC Genomics Server* does not have access to the external network but the *CLC Genomics Workbench* does.

Changing the reference data location

You can specify a different location to download reference data to on the *CLC Genomics Workbench*. This is recommended if you do not have enough space in the default area. To do this, go to the **Navigation Area** and:

Right-click on the folder "CLC_References" | **Choose "Location"** | **Choose "Specify Reference Location"**

If it does not already exist, this folder will be created. It is then registered as the place to

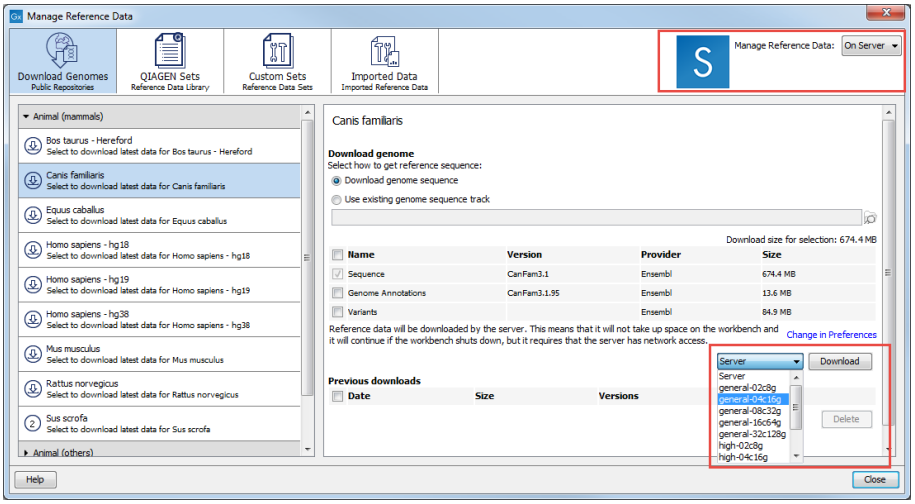


Figure 8.3: *Reference data will be downloaded to the CLC_References area on a* **CLC Genomics Server** *when the "On Server" option is selected. For grid setups, the grid preset to use when downloading data under the Download Genomes tab can be selected, as shown here.*

download reference data to. Workflows configured to use particular Reference Data Sets will look in this new location for those.

This action does **not** remove the old CLC_References folder or its contents. Standard system tools should be used to delete these items if they are no longer needed. Alternatively, this data can be moved to the new location using standard system tools.

Reference data on non-networked systems

CLC Genomics Workbench may be installed on systems without access to the external network. In that case, 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.

8.1 Download Genomes

In the Download Genomes tab, you can access genomes and associated genomic data such as annotations and known variants (figure 8.4). The data is not provided or hosted by QIAGEN. The workbench only provides an easy way to retrieve data that should otherwise have been downloaded and imported.

The list of organisms is dynamically updated by QIAGEN independently of Workbench versions, so you will always see the most recent list of organisms. Select the organism of your choice to see what is available for download, and check the elements you want to include in the downloading process (the size of the download file is updated with each selected element). Please note that the file size displayed in the setup window for the Download Genomes tool refers to the size of the compressed text files, which the tool is retrieving from the provider's depository. The size of the track objects will be, after decompression and conversion from text to the .clc track format, larger.

The reference sequence will be downloaded automatically from Ensembl. You can also choose to select an existing genome sequence track from your Navigation Area to initiate the download. In that case, the reference sequence has to match the genome definition built into the download tool. This means that the name and length of the chromosomes in your reference sequence have to match the genome definition of the tool.

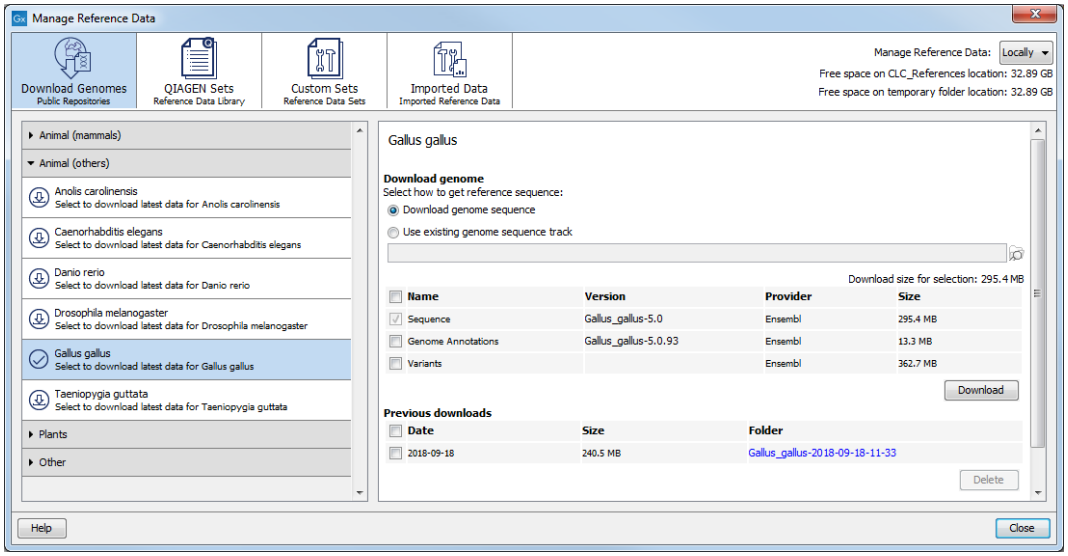


Figure 8.4: *Download genomes and associated data for selected organisms.*

All data downloaded with this tool will be tracks (either sequence tracks or various kinds of annotation tracks).

Which types of annotation are available for download is different from organism to organism and depends on the data sources that QIAGEN has included for download.

The Ensembl human gene annotation download will produce various tracks describing the chromosomal positions of features such as exons, genes, untranslated region (UTRs), transcripts, selenocysteines, coding sequences (CDSs) and mRNAs. The tracks will be saved in a folder called "Genomes" in the CLC_References folder of the Navigation Area. Previous downloads are listed in the Reference Data Manager dialog, and can be deleted from there if needed (but note that this requires admin rights if the files are located on a server).

When GFF3 files are imported, an output track will be issued for each feature type present in the file (see section 6.2.1), and in addition, the 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 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).

UCSC Genome Browser available data include dbsnp variants and chromosome ideograms, also called a cytogenetic ideograms, i.e., a chromosome map with numbered banding patterns that shows the relationship between the two chromosome arms and the centromere (figure 8.5). However, variants downloaded from UCSC will not be annotated on the mitochondrial genome.

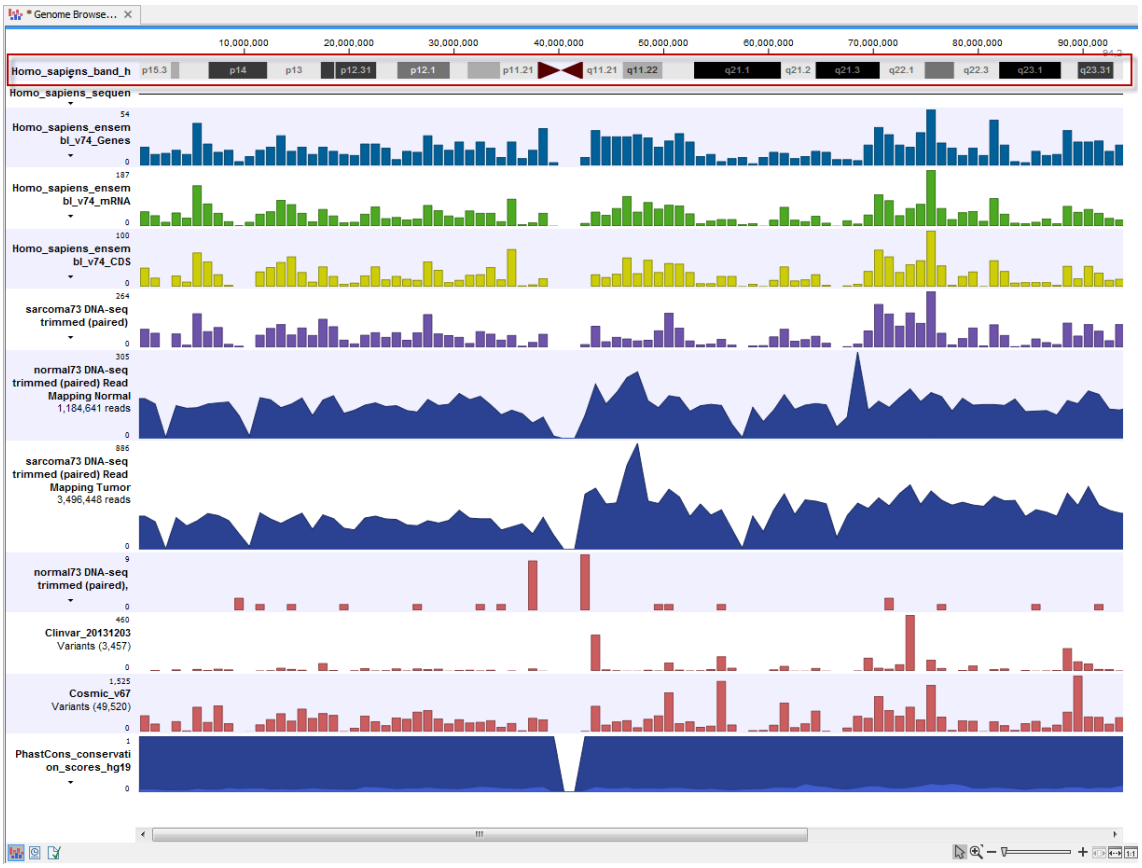


Figure 8.5: *The ideogram is particularly useful when used in combination with other tracks in a track list. In this figure the ideogram is highlighted with a red box.*

8.2 QIAGEN Sets

The **QIAGEN Sets Reference Data Library** tab gives access to all the reference data used with the Biomedical Genomics Analysis plugin ready-to-use workflows and tutorials, as well as custom workflows. From the wizard you can download and configure the reference data.

In this tab, Reference Data Sets and Elements available for download are listed to the left of the wizard under 6 headers (see figure 8.6). Each **Reference Data Set** is a collection of **Reference Data Elements**. Each Reference Data Element is assigned a **Workflow Role** when part of a

Reference Data Set. The Workflow Role of a Reference Data Element determines which inputs it will be used for when running a configured workflow with the given Reference Data Set.

Downloading sets will automatically download the elements the set is made of, but you can also download elements individually under the **Reference Data Elements** folder. **Tutorial Reference Data Sets** are made to use with some of our tutorials (https://digitalinsights.qiagen.com/support/tutorials/) (beware that some are chromosome-specific). The **Previous Reference Data Sets** folder contains older versions of the Reference Data Sets that have been replaced with updated ones in the Reference Data Sets folder.

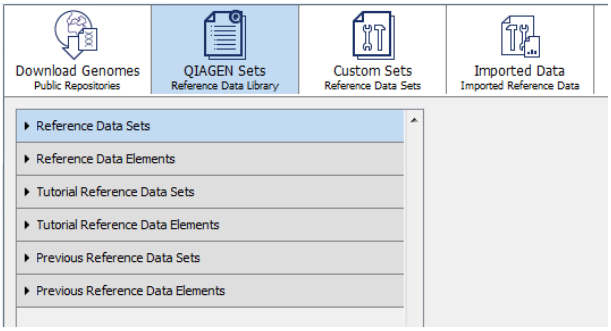


Figure 8.6: *Reference data (Data Sets and Elements) available for download are sorted in 6 different categories.*

Icons to the left of the listed names indicate whether you have already downloaded this data in your Reference folder (✓) or not (↓).

When selecting a reference set, you will see the size of the whole data set, as well as a table that recapitulates the elements included in the set with their version number and respective size.

Click on the **Download** button: Once the data is downloading, you can check the progress of the download, **Cancel**, **Pause** or **Resume** it (see figure 8.7).

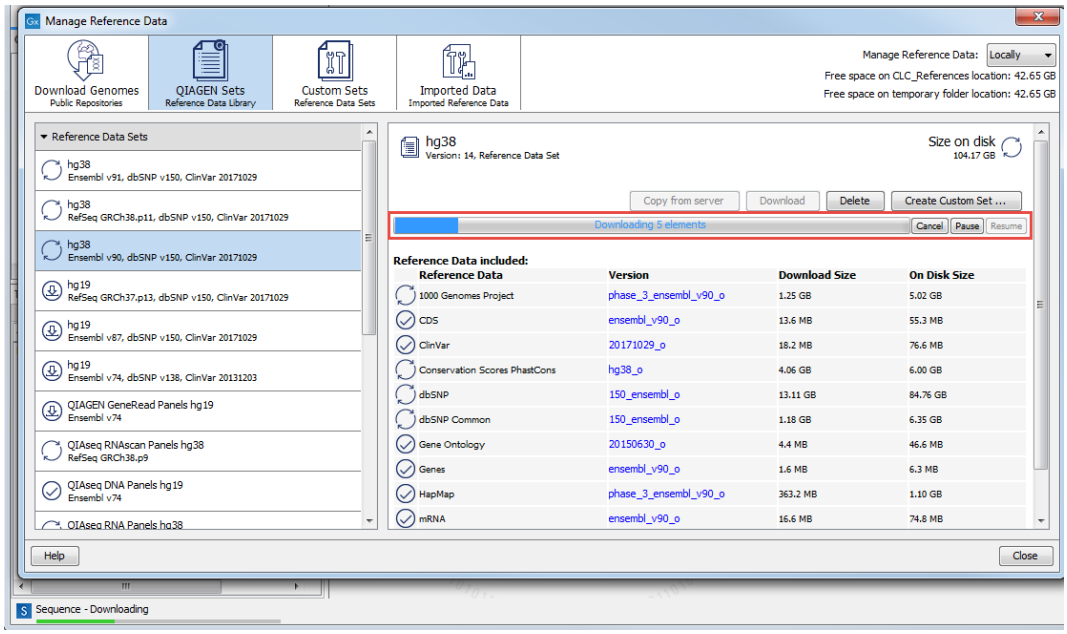


Figure 8.7: *Download of reference data is ongoing.*

The QIAGEN Sets wizard also offers a **Create Custom Set ...** button that allows you to create

your own set of reference data starting from an existing data set.

Note for the 1000 Genomes Projec (http://www.1000genomes.org/category/frequently-asked-questions/population) and HapMap (http://www.sanger.ac.uk/resources/downloads/human/hapmap3.html): These databases contain more than one reference data file, and the QIAGEN References Data Sets are initially configured with all populations available. You can specify which populations to use in the workflow wizard directly, or you can create a custom set that contains only the populations you want to work with.

The **Delete** button allows user to delete locally installed reference data. This can be used if you suspect that a downloaded reference is corrupt, and needs to be re-downloaded, or if you need to clean up space locally. Only administrators are able to delete reference data installed on the server.

The **Copy from server** / **Copy from workbench** button allows to copy locally a Reference Data Set already stored on the Server, and vice versa.

8.3 Custom Sets

The References Management offers a **Custom Sets** tab that allows you to store and generate your own sets of reference data (figure 8.8).

Custom Sets (a collection of chosen reference elements) can be used as reference data when running workflows:

- if the workflow inputs have been configured with workflow roles (see http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Configuring_input_output_elements.html).
- and the reference elements have been assigned matching workflow roles.

The Workflow Role of a Reference Data Element create the link between the workflow input and the reference data element in the custom set.

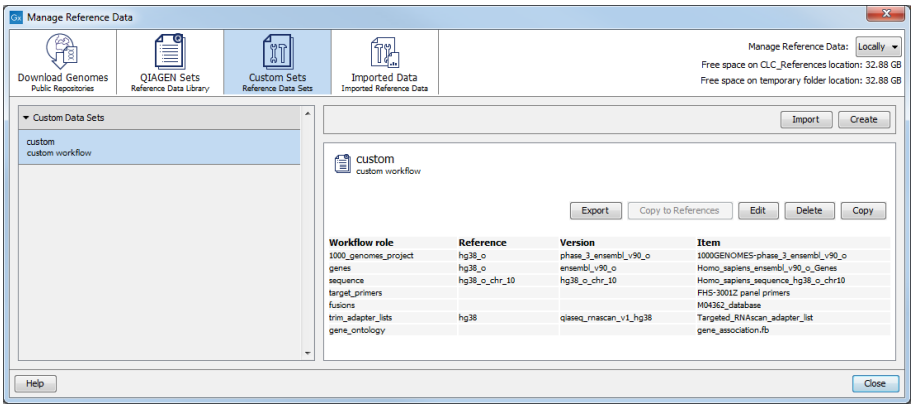


Figure 8.8: *The Custom Sets wizard.*

The principle of creating a custom Reference Data Set is to specify successively the different roles needed, followed by specifying the data fitting to the elements specified.

One way to create a custom reference data set is to click on **Create**. After naming the data set and providing a description for this data set, you can

- Select successively the roles that are needed by the workflow using the drop down menu.
- Create new roles by simply typing in the field a new role name as seen in figure 8.9. Custom roles are especially useful when working with complex workflows that include many different inputs.

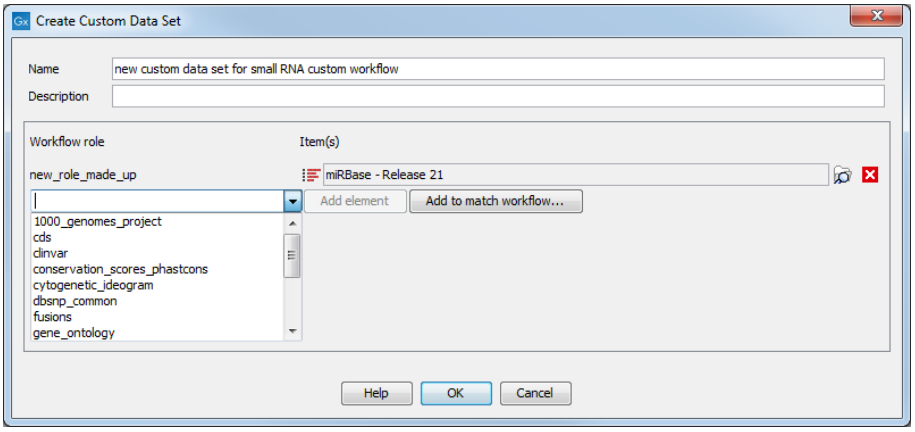


Figure 8.9: *The Create Custom Sets dialog showing a newly created role, and the drop down menu of already existing roles.*

- Use a pre-existing workflow to list roles needed: click on **Create** and then **Add to match workflow ...** (figure 8.10). You can then select an installed workflow (from the drop down list) or a workflow from the Navigation Area (to be specified in the "Custom workflow" field) for which roles have been assigned during configuration. The workflow roles specified in the workflow will then be listed here. The first line "Workflow role" can be checked/unchecked as a way to select/deselect all items in the list. Once you have picked the items you wish to include in your custom data set, click **OK** to proceed.

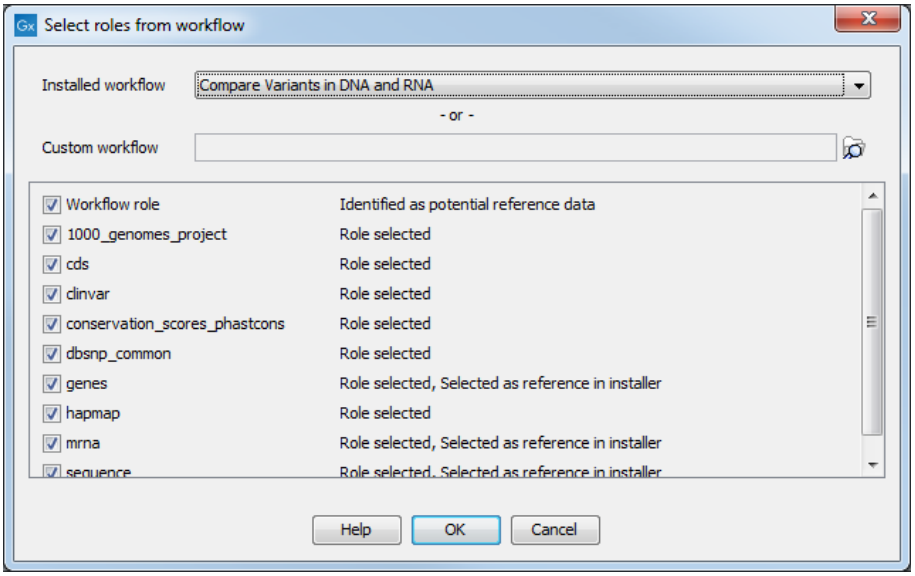


Figure 8.10: *Select the workflow for which you need a custom Reference Data Set.*

Now all the needed reference roles are listed in the main window (figure 8.11). For each role, you will specify the relevant reference element by clicking on the Browse icon to the right of the field - or delete the particular role from the Reference Data Set by clicking on the cross icon.

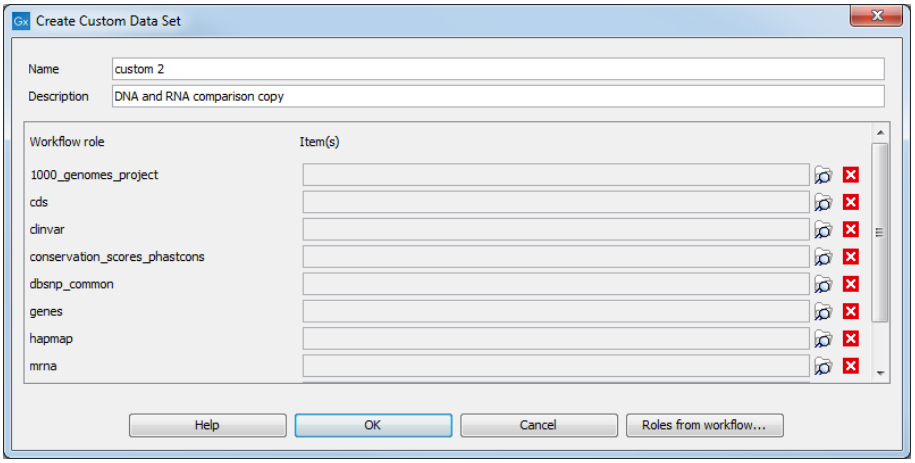


Figure 8.11: *Select the elements for each role included in your custom Reference Data Set.*

There are two ways to find a particular element: by using the Navigation Area tab or from the Reference Data tab (figure 8.12). In the Navigation Area tab, the references elements are sorted by role, in organisms specific subfolders, in the CLC_References folder (local or on server). In the Reference Data tab, the elements are sorted by role in Data Sets specific folders, including custom data sets.

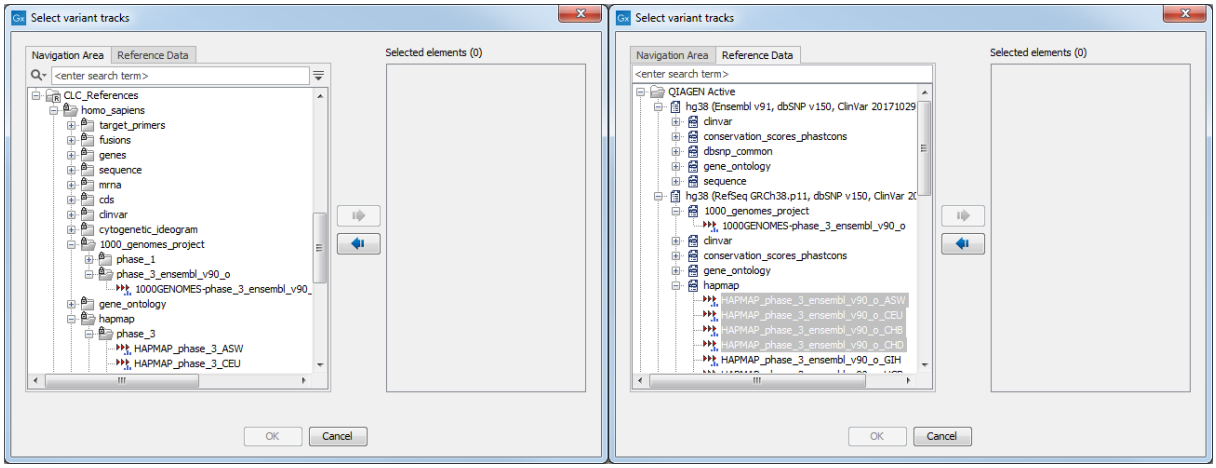


Figure 8.12: *Find the relevant reference element, either from the Navigation Area tab or from the Reference Data tab.*

Note that it is only possible to select an element whose role is the one defined by the list: for example, when browsing for 1000_genomes_project files, it will not be possible to specify a genes track. However, it is possible to create new roles as needed: just type in the name of the custom role in the field and click Add element. There is no restriction on the type of file that can be selected for custom roles.

Once you are ready to save your custom data set, give it a name, and indicate in the description the workflow it is suitable for before clicking on **OK**. The custom set is now listed to the left side of the wizard. Saving a custom data set on a server is the most efficient way to share it with all other server users. Server admins can choose to lock or unlock reference data set on the server location so that the data sets cannot be deleted or modified.

8.3.1 Copy to References

When specifying an Element, it is recommended to use files that were previously saved in the CLC_References folder, but it is also possible to choose a file from any location in the Navigation Area. When using the last option, be aware that this file is not protected as read-only and may be subject to changes. To avoid this, click on "Copy to References" to save the Navigation Area element in the CLC_References file as read-only (figure 8.13).

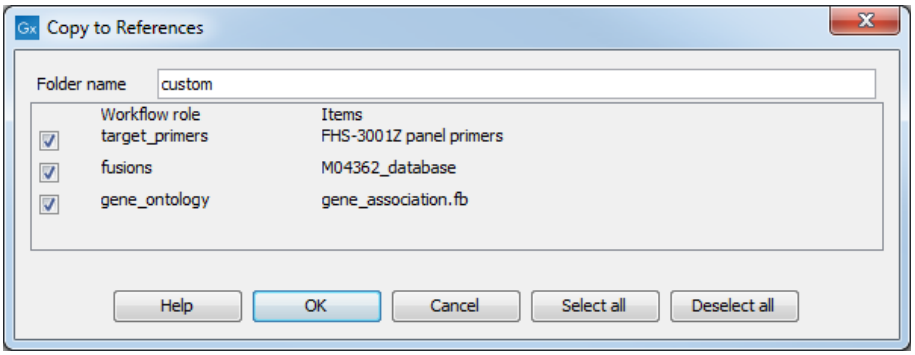


Figure 8.13: *Copying an element from the Navigation Area to the CLC_References read-only folder.*

Once the data has been copied to the CLC_References data set, the custom workflow will automatically refer to the file in the read-only folder rather than from the location in the Navigation Area that was originally specified.

For information about the additional functionalities on the Imported Data Sets, please see the section of the same name in the CLC Genomics Workbench manual.

8.3.2 Export a Custom Data Set

It is possible to export custom data set (figure 8.14). The export format is *cpc, and can be imported by anyone who has installed a workbench (even in Viewing Mode only).

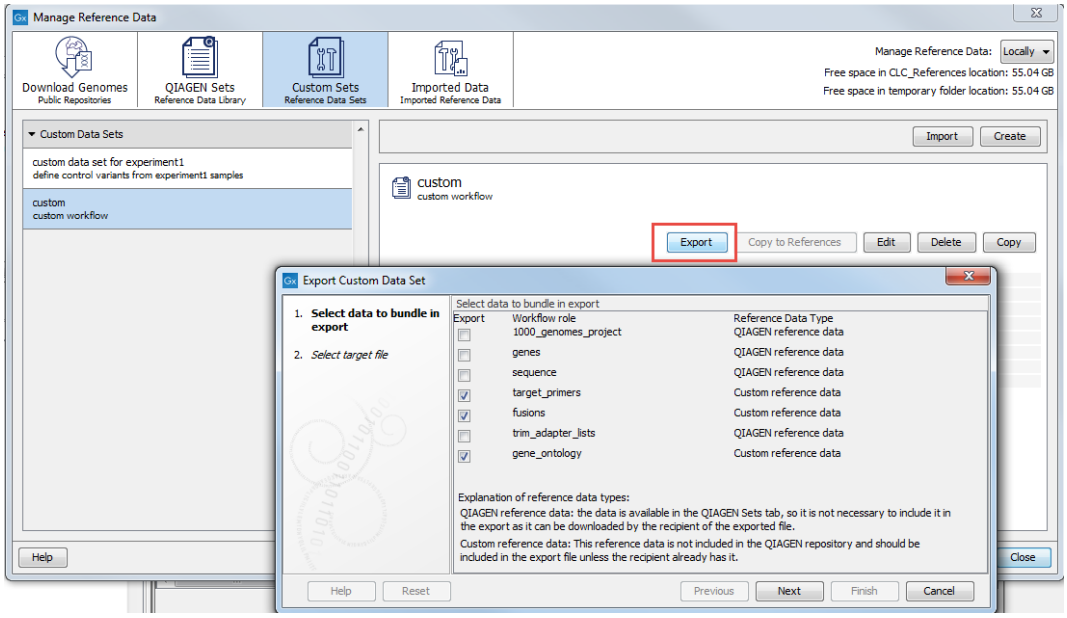


Figure 8.14: *Exporting a custom data set from the Custom Sets tab. In this example, we are exporting 7 roles, as well as data for three of them.*

Exporting a custom data set will export the different roles that are included in the data set. The export function can also export the data associated to the role, but it not always necessary. When the Reference Data Type is set to:

- QIAGEN reference data: the data is available in the QIAGEN Sets tab, so it is not necessary to include it in the export file, as it can be downloaded by the recipient of the exported file using the Reference Data Manager.
- Custom reference data: This reference data is not included in the QIAGEN repository, so it may have to be included in the export file to be shared with the recipient.
- Not reference data: A file that does not belong to the CLC_References folder has been chosen as part of the custom data set. This file needs to be imported as a reference (via the Copy to References button) and specified in the custom reference data from its location in the CLC_References folder to be exported.

When the export check box is unchecked, only the role will be exported. If the box is checked, the data associated with the role will also be exported. That way, it is possible to create small export files that only contain roles and custom elements, while the elements that are available in the QIAGEN Reference Sets list can be downloaded by the recipient of the exported file.

8.3.3 Import a Custom Data Set

It is possible to import custom data set *cpc from the Custom Sets tab (figure 8.15).

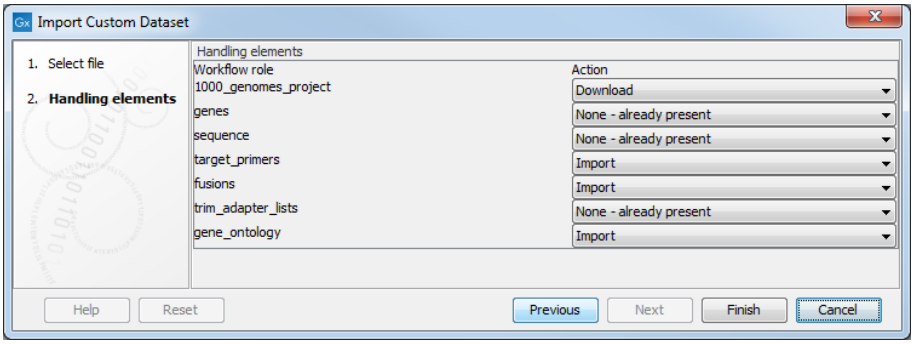


Figure 8.15: *Importing a custom data set from the Custom Sets tab. In this example, we are exporting 7 roles, as well as data for three of them.*

When importing, you are in fact importing a set of roles. In addition, you are also given options on how to handle the incoming data when present:

- Download: The custom data set contains some QIAGEN reference data that was specified, but not exported. By leaving the status to Download, you will save the data directly from the QIAGEN repository to your CLC_References folder. This may take some time for particular elements.
- None - already present: this means that the data specified in the custom data set is already saved in your own reference data.
- Import: The custom data set includes some custom reference data that was exported together with the list of roles. Importing the data set will then also import the data and save it to a newly defined folder in the CLC_References read-only folder.

You can choose not to import or download the data attached to the roles included in the data set using the drop down menu option present for each role included in the reference data set. If you include data, you can check the progress of the import in the Processes tab of the Toolbox. Once the set is imported, you will find the custom elements in the "Imported" subfolder of the CLC_References folder.

8.4 Imported Data

The Imported Data tab enables you to import Data Sets or Elements in the read-only location of the CLC_References folder from two different locations:

- **Copy from the Navigation Area** allows you to select a folder in the Navigation Area for import into the CLC_References folder (for example reference data that was imported through the traditional Import function of the workbench and saved in a specific folder in the Navigation Area)
- **Import from file** allows you to specify a *.cpd data package on your computer for import into the CLC_References folder. A *cpd file can be generated by exporting a data package as explained below.

It is possible to edit or add information to the data imported with the buttons above: Name of the dataset, Description, Author name, Author email, and Organization. For folders imported using the **Copy to References** button of the Customs Sets tab, the button **Finalize** can be used to add the above information (figure 8.16). Finalized imported data means that it is not possible to add additional elements to the imported folder.

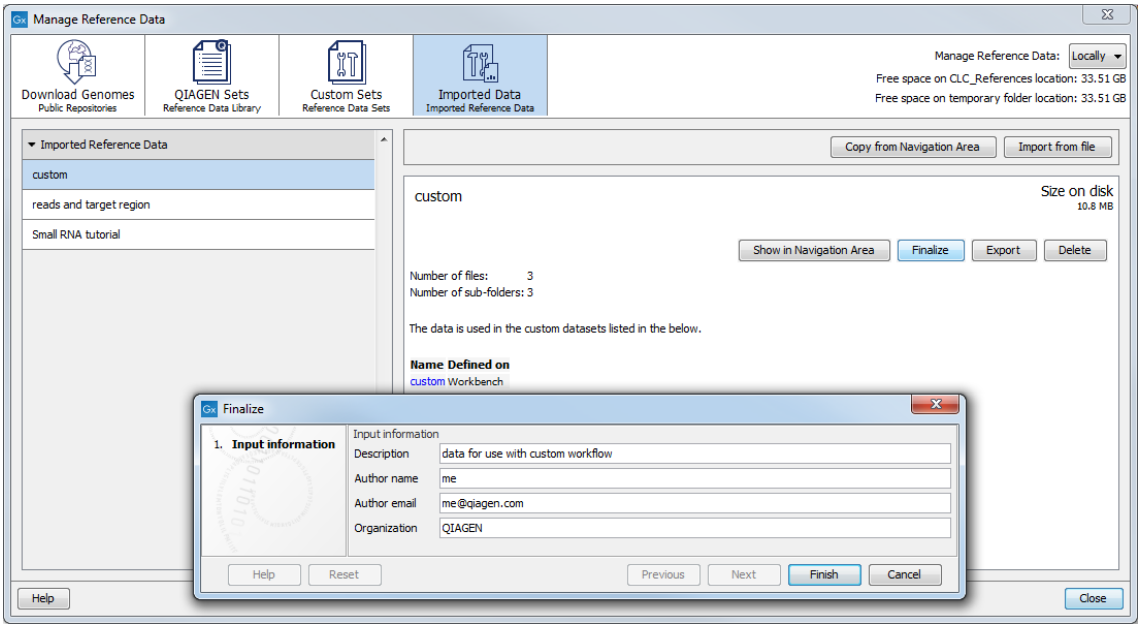


Figure 8.16: *Finalizing an imported reference set from the Imported Data tab.*

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.

8.4.1 Exporting reference data outside of the Reference Data Manager framework

This can be very useful when using the external applications, i.e., programs that are not part of the CLC workbench framework but that can be run using the same reference data.

To export a reference data set for external applications, you cannot use the **Export** button from the Reference Data Manager (that would export data set as CPD files), but the **Export** tool to export the data in the relevant format (VCF, FASTA, etc).

For example, if the Reference Data element "Clinvar" already has been exported, then there might be a folder called `/homo_sapiens/clinvar/20131203` with the file `Clinvar_20131203.vcf` If the export is invoked again, then the folder will contain two identical files with difference names: `Clinvar_20131203.vcf` and `Clinvar_20131203.1.vcf`. The second file will not be used.

No special permissions are required to export reference data, but administrator rights are required to delete reference data. If it becomes necessary to delete exported reference data, an administrator, super user, or some user with administrator rights, must do this. Deletion of exported data has to be done through the operation system, it cannot be done through the Workbench, nor the CLC server.

Chapter 9

Running tools, handling results and batching

Contents

9.1 Running tools 188
9.1.1 Running a tool on a CLC Server 191
9.2 Handling results 192
9.3 Batch processing 193
9.3.1 Standard batch processing 194
9.3.2 Batch overview 194
9.3.3 Parameters for batch runs 195
9.3.4 Running the analysis and organizing the results 196

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 11.

9.1 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 workflow:

- Double click on its name in the Toolbox tab in the bottom left side of the Workbench.
- Select it from the Toolbox menu at the top of the Workbench.
- Use the Quick Launch ( ) tool, described below.

Using the Quick Launch tool to start jobs

The Quick Launch tool, shown in figure 9.1, is started by clicking on the **Launch** button ( ) in the toolbar. It can also be launched using the keyboard shortcut Ctrl + Shift + T (⌘ + Shift + T on Mac), or by going to the Toolbox menu at the top of the Workbench and selecting the top option, Launch ( ).

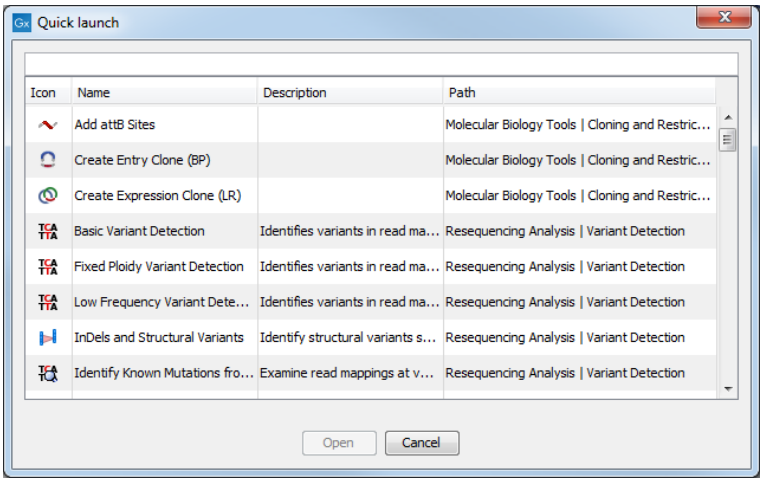


Figure 9.1: *Tools and installed workflows can be quickly found and launched using the Quick Launch tool.*

In the Quick Launch dialog, you can type terms in the text field at the top. This will filter for tools and installed workflows with matches to these terms in the name, description or Toolbox location. For tools where names have been changed between Workbench versions, searches using old names will still filter for the relevant tool. Using single or double quotes (' or ") will find a literal quote of the searched term.

In the example shown in figure 9.2, typing `create` shows a list of tools involving the word *create*. The arrow keys or mouse can be used for selecting and starting a tool from this list.

Configuring and submitting jobs

When you open a tool or installed workflow, a wizard pops up in the center of the View Area. Stepping through a succession of wizard steps, you will select the data to analyze, configure any analysis parameters, and specify how the results should be handled. You can navigate between wizard steps by clicking the buttons **Next** and **Previous** at the bottom of the window.

If you have logged into a *CLC Server* from your Workbench, you will first be asked to select whether the job should be run on the Workbench or submitted to the server. These choices, along with information about data selection and other considerations when launching tasks on a *CLC Server* are provided in section 9.1.1.

Generally, the first analysis configuration step involves selecting the data elements to be used as input. A view of your Navigation Area will be presented to you. That view will show data elements

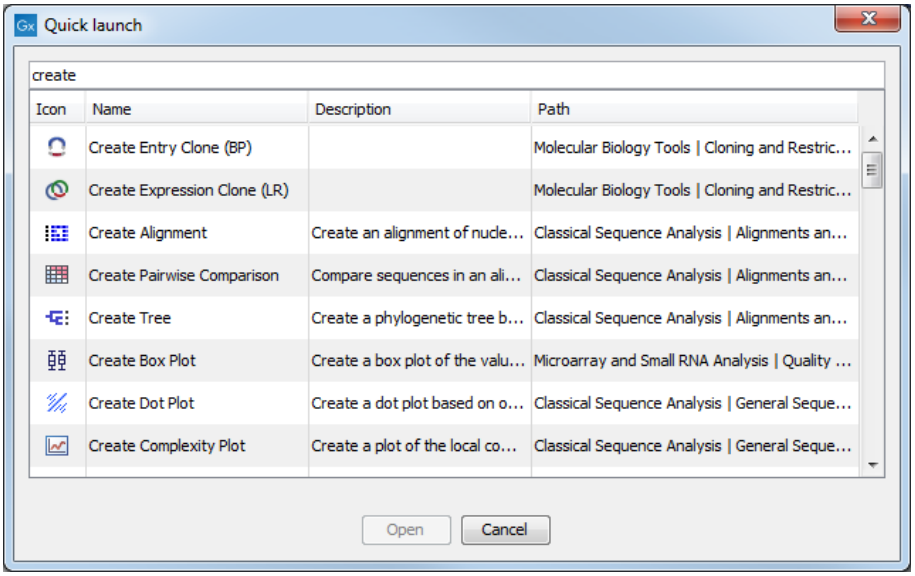


Figure 9.2: *Typing in the search field at the top will filter the list of tools to launch.*

appropriate for use as input for that tool. Folders are also shown. For example, in figure 9.3 you can see a the Workbench Navigation Area (on the left) and a view of the same Navigation Area in the wizard (on top) for the Assemble Sequences tool. This tool only accepts nucleotide sequences and nucleotide sequence lists, so data elements of other types that can be seen in the Workbench Navigation Area, such as the one called "Read mapping", and the amino acid sequence ATP8a1, are not displayed in the wizard Navigation Area.

The data types that can be used as input for a given tool are described in the manual section about that tool. This documentation can be opened directly by clicking on the **Help** button in the bottom left corner of the launch wizard.

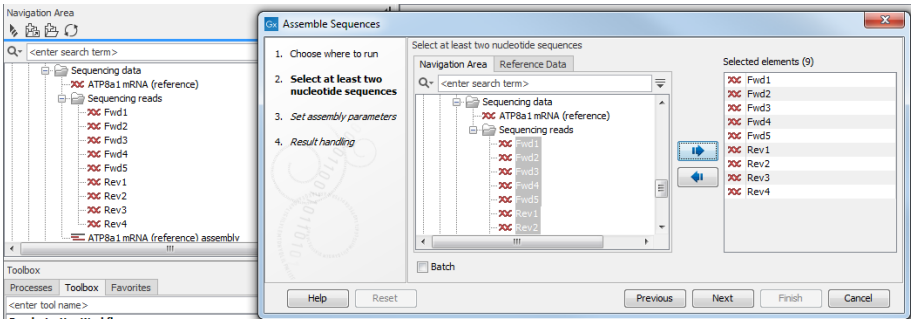


Figure 9.3: *You can select input files for the tool from the Navigation Area view presented on the left hand side of the wizard window.*

To indicate the data elements to be used in the analysis, either double click on them in the "Navigation Area" view on the left, or select them with a single click and then click on the right hand arrow. These items will then be listed in the "Selected elements" list on the right. If data elements of appropriate types were already selected in the Workbench Navigation Area before launching the tool, these will be automatically entered into the Selected elements list. To remove entries in that list, just 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 treat all those elements as a single input data set unless the "Batch" option at the bottom, has been selected. If that option

is selected, then the tool will be run multiple times, once for each "batch unit", which may be a data element, or folder containing data elements or containing folders of elements. Batch processing is described in more detail in section 9.3.

Once the data of interest has been selected, click on **Next**. Depending on the tool, there may now be one or more steps for configuring analysis parameters. An example is shown in figure 9.4. Clicking on the **Reset** button resets all parameters in that step to their default values.

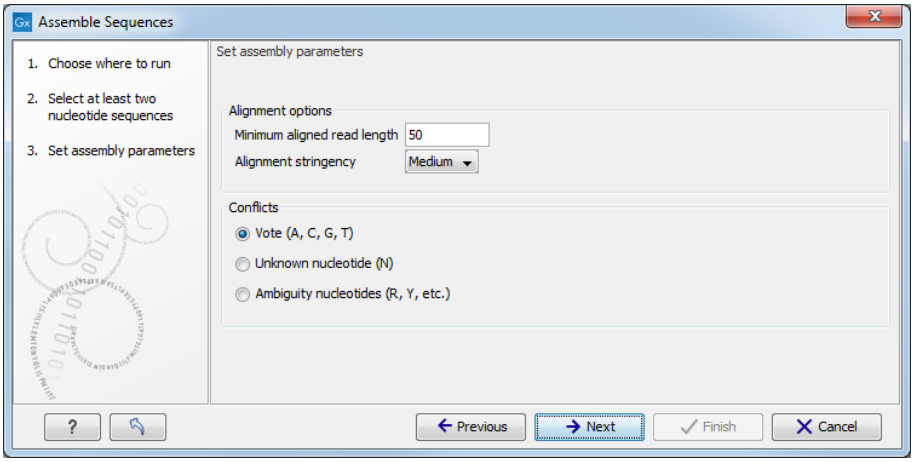


Figure 9.4: *An example of a "Set parameters" window.*

9.1.1 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, as shown in figure 9.5.

- **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.

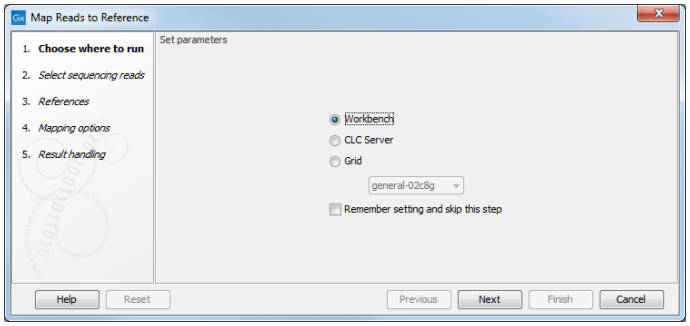


Figure 9.5: *When logged into the CLC Server, you can select where a job should be run.*

You can check the **Remember setting and skip this step** option if you wish to always use the selected option when submitting analyses that can be run on a *CLC Server*. If you select this

option but later change your mind, just start up an analysis and click on the **Previous** button. The step with the options for where to run analyses will then be shown.

The remaining wizard steps are the same whether you are launching a job on a *CLC Workbench* or a *CLC Server*, with two minor differences: when running on *CLC Server*, results are always saved, and a log of the job is always created and saved alongside the results.

When launching a task to run on a CLC Server or on grid nodes, there are a few things to be aware of:

- You can only analyze data stored in *CLC Server* data areas. Thus only these data areas will be offered to select from when configuring an analysis.
- You have to save the analysis results. For workflows, this includes creating and saving a workflow result metadata table. By contrast, for a single analysis run on the Workbench, you can normally choose whether to **Open** or **Save** results, and you can choose whether to create a workflow result metadata table.
- After you have launched an analysis, it is submitted to the *CLC Server* to be handled. You can then close the Workbench or disconnect from the *CLC Server* if you wish. If an analysis finishes while your Workbench is closed or not connected to the *CLC Server*, you will see a notification about this when you next log in from the Workbench.
- When importing data into a *CLC Server*, the location of the data being imported affects when it is safe to close the Workbench or disconnect from the server. Further information about that can be found in section 2.4.

9.2 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 9.6). 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:

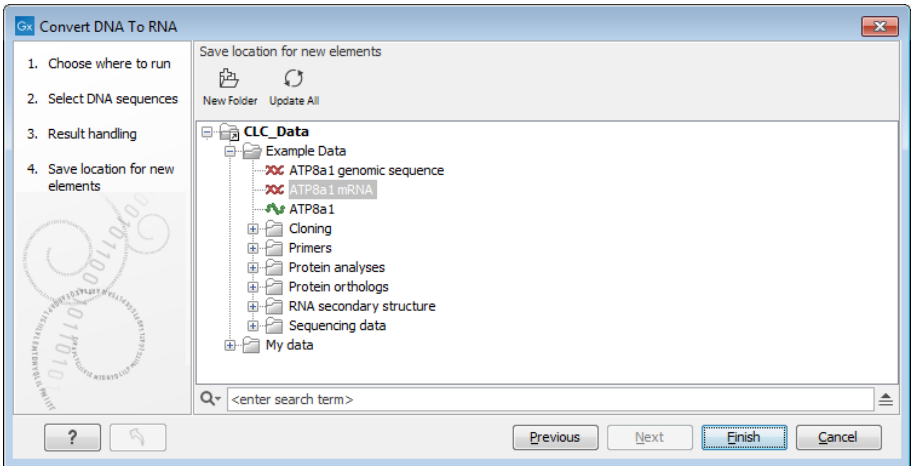


Figure 9.6: *Specify where to save the results of an analysis.*

- Drag the tab to the relevant location in the Navigation Area.
- 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 9.7.

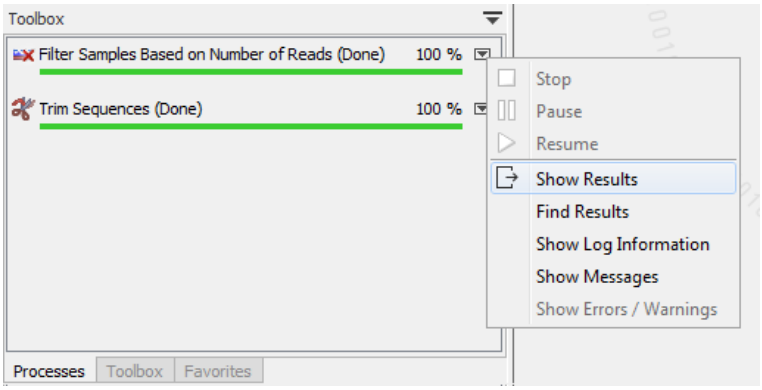


Figure 9.7: *Find or open the analysis results by clicking on the little arrow to the right of the analysis name in the Processes tab and choosing the relevant item from the menu.*

9.3 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 describes batch processing when using **individual analysis tools**. Launching workflows using batch functionality is covered in section 11.4. Running parts of workflows in batches is covered in section 11.5.

9.3.1 Standard batch processing

Batch mode is activated by clicking the **Batch** checkbox in the dialog where the input data is selected (figure 9.8).

Unlike launching a single task, you can select a folder as well as, or instead of, individual data elements for the analysis.

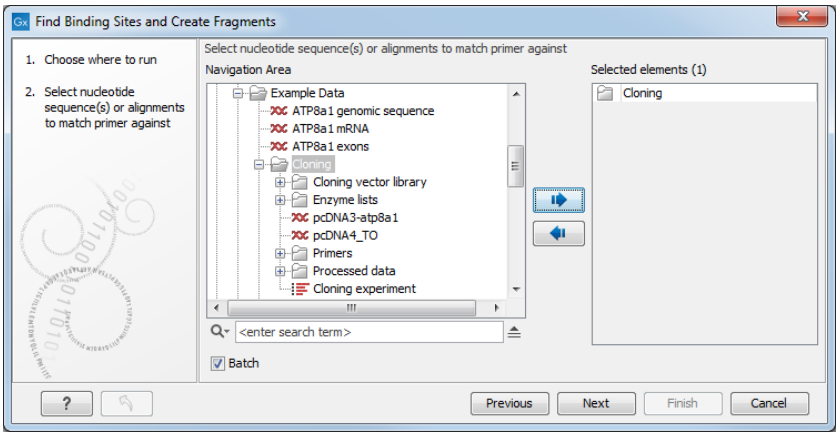


Figure 9.8: *The Cloning folder includes both folders and sequences.*

A batch unit is the set of data that will be used as a single input set for a given run of an analysis. A given batch unit can consist of one or more data elements.

If a folder is selected as input to a batch analysis, each folder or data element directly under that folder will be considered a batch unit. This means:

- Each individual data element contained directly within the folder is a batch unit.
- Each subfolder directly within this folder is a batch unit, so all elements within a given subfolder will be considered as single input for the purposes of the analysis.
- Elements in any more deeply nested subfolders (e.g. subfolders of subfolders of the originally selected folder) will not be considered for the analysis.

9.3.2 Batch overview

The next Wizard step is the batch overview where you have the opportunity to refine the list of data that will be in each batch unit. For example, you could use this step to ensure that only trimmed sequence lists - and not all sequence lists - should be used for the analysis that is being setup.

The batch overview lists the batch units on the left and the contents of the selected batch unit on the right (figure 9.9).

In this example, the two sequences (pcDNA) are defined as separate batch units because they are located at the top level of the Cloning folder. Of the four subfolders of the Cloning folder

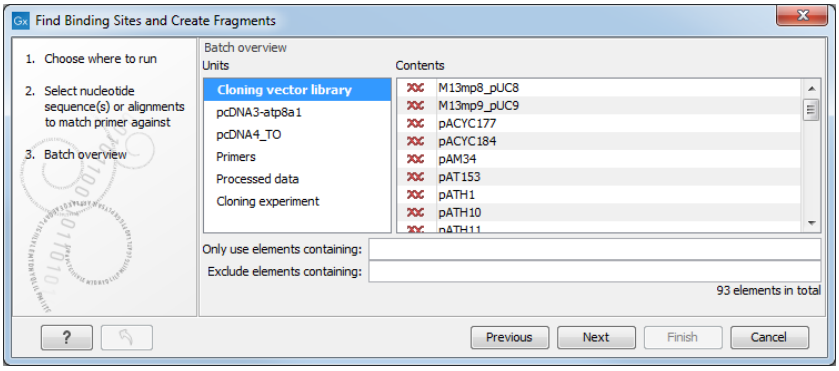


Figure 9.9: *Overview of the batch run. At the bottom right, the number of files to be analysed, summed across all batch units, is shown, 92 in this case.*

initially selected, three are listed in this view. In each of these subfolders, any data elements that the analysis could use as input will be used unless action is taken at this point to exclude some of these. So all the elements in the subfolder "Cloning vector library" and shown on the right-hand side of the dialog will be included as part of a single analysis run.

Note that folders that do not contain any data that can be used by the tool being launched will not be shown in that dialog.

Including and excluding data elements in batch units There are three ways to refine the data elements that should be included in a batch unit, and thereby get taken forward into the analysis.

- **Use the fields labeled Only use elements containing and Exclude elements containing at the bottom of the batch overview** This refinement is done based on data element names. for example, only paired reads might be desired for the analysis, in which case, putting the text "paired" into the **Only use elements containing** field might be useful.
- **Remove a whole batch unti** Right-click on the batch unit to be removed and choose the option **Remove Batch Unit**.
- **Remove a particular data element from a batch unit** Right click on the element of a batch unit to be removed and choose the option **Remove Element**. This can be useful when filtering based on name, described in the first option, cannot be used to refine the batch units specifically enough.

9.3.3 Parameters for batch runs

The subsequent dialogs depend on the analysis being run and the data being input. Generally, one of the batch units will be specified as the parameter prototype and will then be used to guide the choices in the dialogs. By default, the first batch unit (marked in bold) is used for this purpose. This can be changed by right-clicking another batch unit and choosing the option **Set as Parameter Prototype**.

When launching tools normally (non-batch runs), the Workbench does much validation of inputs and parameters. When running in batch, this validation is not performed. This means that some analyses will fail if combinations of input data and parameters are not right. Therefore we recommend that batching is used when the batch units are quite homogenous in terms of the type and size of data.

9.3.4 Running the analysis and organizing the results

The last step in setting up a batch analysis is to choose where to save the outputs (figure 9.10).

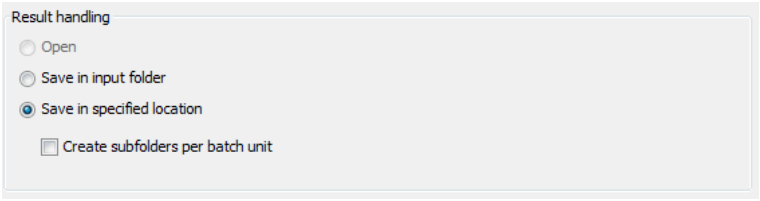


Figure 9.10: *Options for saving results when the tool was run in batch.*

The options available are:

- **Save in input folder** Save all outputs into the same folder as the input data. If the batch units consisted of folders, then the results of each analysis would be saved into the folder with the data it was generated using. If the batch units were individual data elements, then all the results will be placed into the same folder as those input data elements.
- **Save in specified location** Choose the folder where the outputs should be saved to, where when:
  - **Create subfolders per batch unit** is **unchecked**, all results for all batch units will be written to the specified folder.
  - **Create subfolders per batch unit** is **checked**, results for each batch unit will be written to a newly created subfolder of the selected folder. One subfolder is created per batch unit.

When the batch run is started, there will be one "master" process representing the overall batch job, and there will then be a separate process for each batch unit. The behavior this is different for Workbenches and Servers:

- On a Workbench, only one batch unit is run at a time. So when the first batch unit is done, the second will be started and so on. This avoids many parallel analyses that would draw on the same compute resources and slow down the computer.
- On a CLC Server, all the processes are placed in the queue, and the queue takes care of distributing the jobs. This means that if the server set-up includes multiple nodes, different batch unit analyses may be run in parallel.

To stop the whole batch run, stop the "master" process. From the Workbench, this can be done by finding the master process in the Processes tab in the bottom left hand corner. Click on the little triangle on the right hand side of the master process and choose the option **Stop**.

For some analyses, there is an extra option in the final step to create a log of the batch process. This log will be created in the beginning of the process and continually updated with information about the results. The log will either be saved with the results of the analysis or opened in a view with the results, depending on how you chose to handle the results.

Chapter 10

Metadata

Contents

10.1 Creating metadata tables 200
10.1.1 Importing metadata 200
10.1.2 Creating a metadata table directly in the Workbench 201
10.2 Associating data elements with metadata 205
10.2.1 Associate Data Automatically 206
10.2.2 Associate Data with Row 208
10.3 Working with data and metadata 209
10.3.1 Finding data elements based on metadata 209
10.3.2 Viewing metadata associations 210
10.3.3 Removing metadata associations 211
10.3.4 Identifying metadata rows without associated data 212
10.4 Moving, copying and exporting metadata 212
10.5 Editing Metadata tables 213

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 are listed below, along with a reference for where further information on that aspect can be found:

- Running tools where characteristics of the data elements are relevant. Examples are the differential expression tools, described in section 30.4.
- Defining batch units when launching workflows or tools in batch mode, and for launching workflows in batches where more than one input should be changed for each batch run, described in section 11.4.

- Distributing data to the relevant input channels in a workflow when using Collect and Distribute elements, described in section 11.5.
- Finding and selecting data elements associated with the metadata. Workflow result metadata tables are of particular use when reviewing results generated by workflows run in batch mode and are described in section 11.4.2.

Metadata tables

An example of a metadata table in the *CLC Genomics Workbench* is shown in figure 10.1. Each column represents a property of a sample (e.g., identifier, height, age, treatment) and each row contains information relevant to a sample. One column will be designated the key column. That column must contain unique entries and is used when associating data elements with a metadata row.

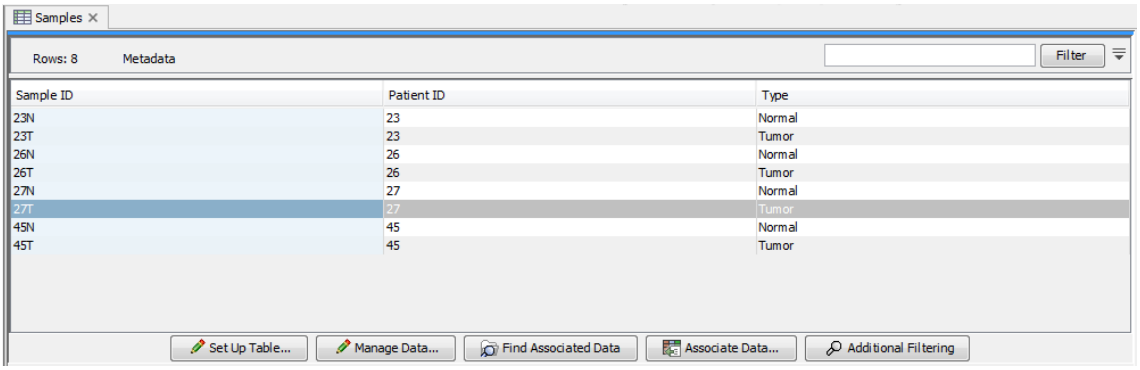


Figure 10.1: *A simple metadata table, with the key column highlighted in blue.*

A row in a metadata table can be associated 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 section 10.2

The most common way to create a metadata table is to import an Excel format file, as described in section 10.1.1. Metadata tables can also be generated by workflows as described in section 11.4.2

Searching for metadata tables based on their contents is described in section 3.4.1.

Metadata Elements table

The data elements associated particular metadata rows can be listed by selecting the metadata rows of interest and clicking on the **Find Associated Data** button. This opens the Metadata Elements table, where the associated data will be listed, as shown in figure 10.2.

When a data element is associated with a metadata row, the outputs of analyses involving that data often inherit the metadata association automatically. This means that a given row in a metadata table can be associated with several data elements. For example, at first a sample might be associated with a sequence list, but after analysis, the same metadata row could be associated with various additional elements in the Metadata Elements table, can be seen in figure 10.2.

Inheritance of metadata associations requires that a single association can be unambiguously

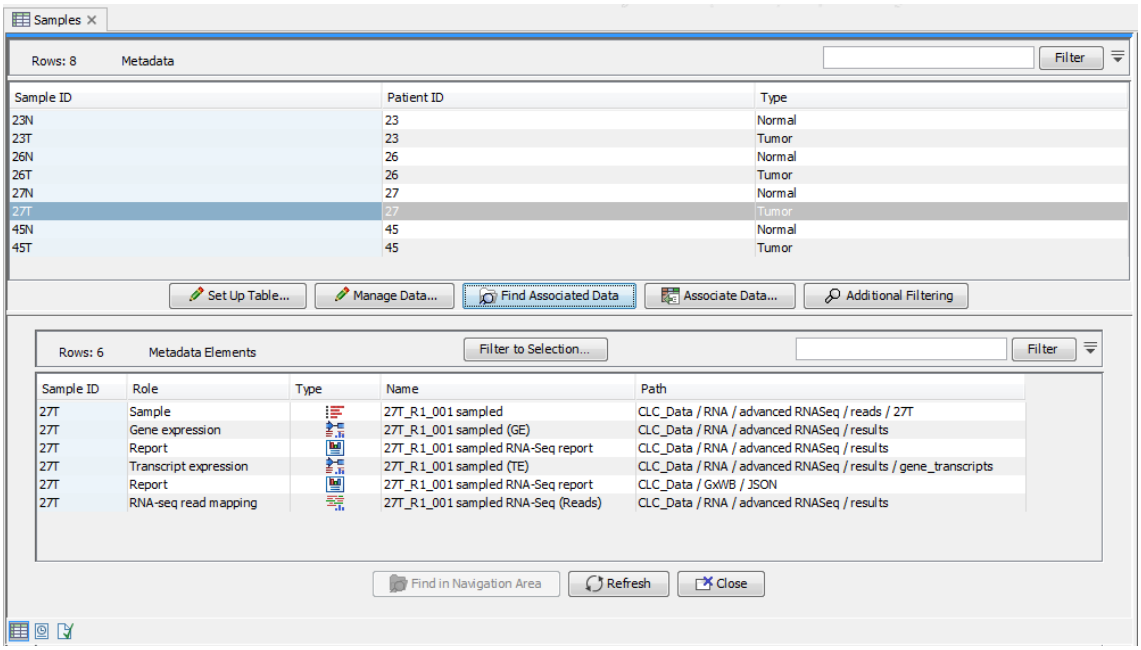


Figure 10.2: *A Metadata Table and corresponding Metadata Elements table showing elements associated with sample 27T.*

identified for an output when a tool is run. If an output is derived from two ore more inputs with different metadata associations, then no association will be inherited.

10.1 Creating metadata tables

10.1.1 Importing metadata

Metadata can be imported from an Excel format file into the *CLC Genomics Workbench* using the **Import Metadata** ( ) tool. This is described in detail in this section.

Creating a metadata table directly this way is not necessary when launching a workflow in batch mode. Here, an Excel file containing metadata can be provided directly when launching the workflow, as described in section 11.4.

The Import Metadata tool

To import an Excel (.xlsx/.xls) file as metadata in the Workbench, go to:

File | Import ( ) | **Import Metadata** ( )

The first column in the Excel file must have unique entries as that column will be designated as the key column. A different column can be specified instead later.

For the optional association with data to work, the first column of the metadata must contain entries that can be matched to data element names.

Importing the Excel file In the box labeled **Spreadsheet with sample information**, select the Excel file (.xlsx/.xls) to be imported. The rows in the spreadsheet are displayed in the Metadata preview window, as shown in figure 10.3.

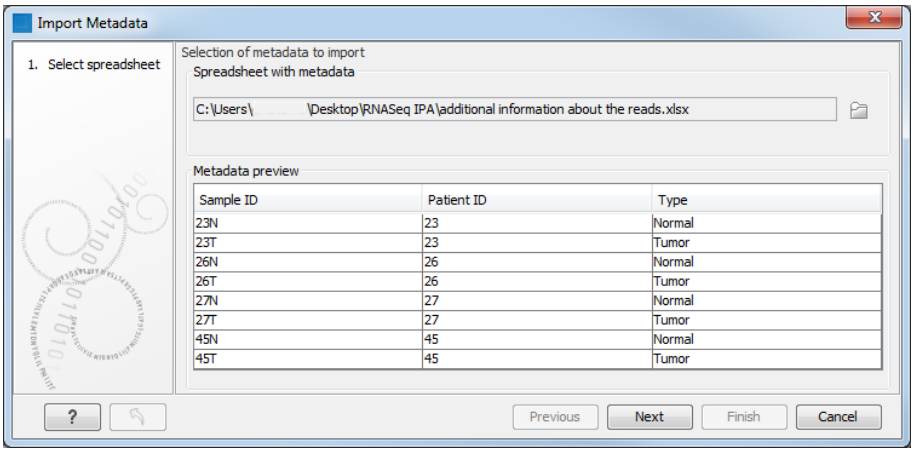


Figure 10.3: *After an Excel file is selected, its rows are visible in the Metadata preview table.*

Note that when using the Import Metadata tool, all columns are imported as text columns. This can be later changed from within the Metadata Table editor as described in section 10.5. 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 second wizard step, called "Associate with data", shown in figure 10.4, 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 10.2.

To associate data with the metadata at this step of importing metadata:

- Click on the file browser button to the right of the **Location of data** field
- Select the data elements that should be associated with the metadata.
- Select the matching scheme to use: Exact, Prefix or Suffix. Detailed explanations of these options are provided in section 10.2.1.

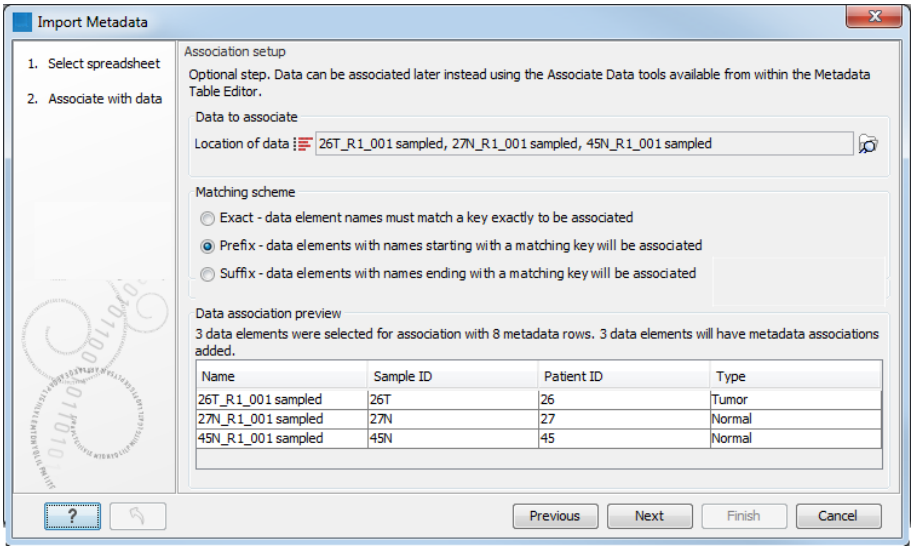


Figure 10.4: *Three data elements are selected for association. The "Prefix" partial matching scheme is selected for matching data element names with the appropriate metadata row, based on the information in the Sample ID column in this case.*

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.

You can then select where you wish the metadata table to be saved and click **Finish**.

The associated information can be viewed for a given data element in the Show Element Info view, as show in figure 10.5.

10.1.2 Creating a metadata table directly in the Workbench

Metadata tables can be created from within the *CLC Genomics Workbench*. Generally, it is much easier to enter the metadata into an Excel file and either import that, as described in section 10.1.1, or to use that file when launching a workflow in batch, and use the workflow results metadata file that is output, which is described in section 11.4.2.

To create a metadata table directly in the *CLC Genomics Workbench*, 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.

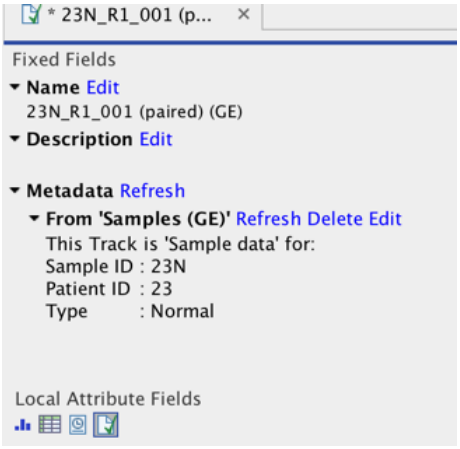


Figure 10.5: *Metadata associations can be seen, edited, refreshed or deleted via the Show Element Info view.*

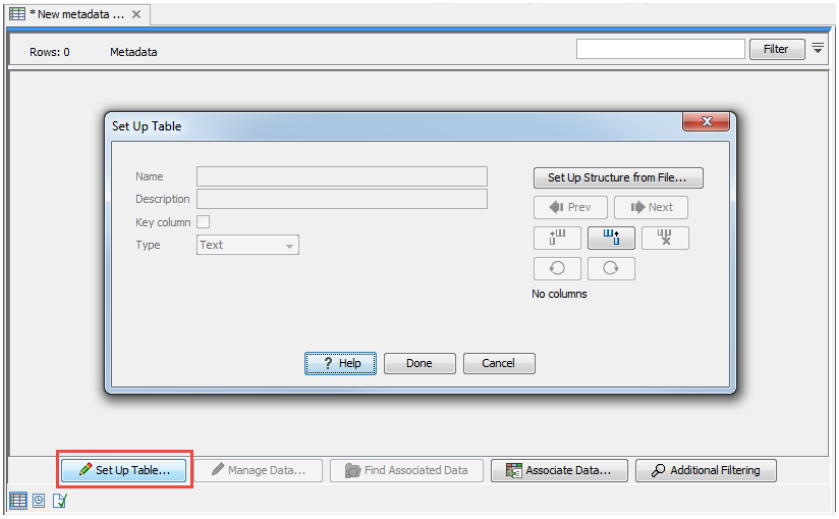


Figure 10.6: *Dialog used to add columns to an empty Metadata Table.*

Defining the table structure Click **Setup Table** at the bottom of the view (figure 10.6).

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 10.7), 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 (;).

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.

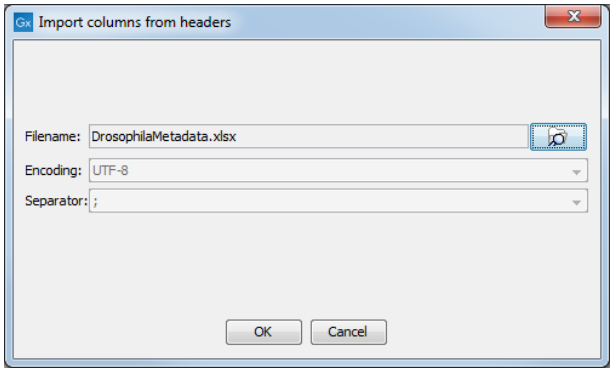


Figure 10.7: *Creating a metadata table structure based on an external file.*

Populating the table Click on **Manage Data** button at the bottom of the view (figure 10.8).

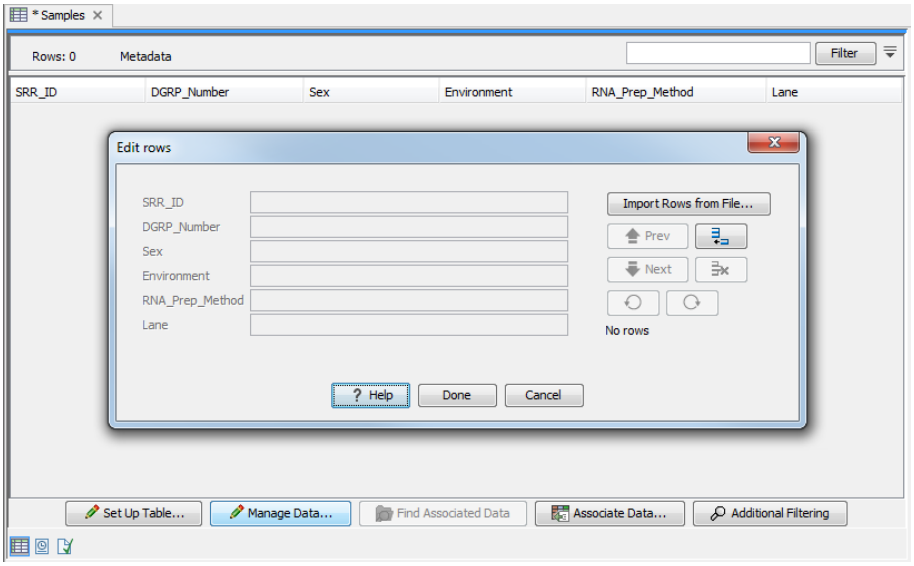


Figure 10.8: *Tool for managing the metadata itself. Notice the button labeled Import Rows from File.*

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 10.9.

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

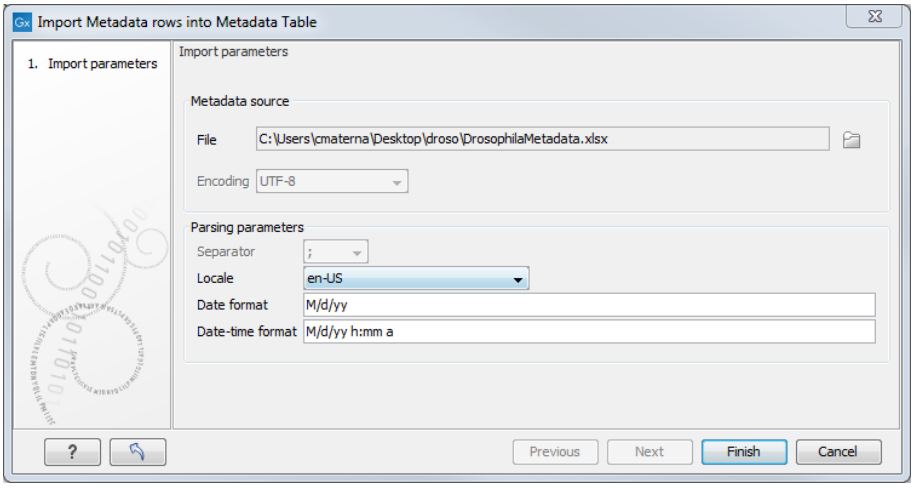


Figure 10.9: *Tool to import rows into a Metadata Table.*

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 seledcted 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 of the Toolbox. 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.

10.2 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 10.2.1.
- **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 10.2.2.

10.2.1 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 10.2.1).

Note: This option is to be used carefully when data elements already have associations with the metadata table. In addition to adding any new associations, the already existing associations will be *updated* to reflect the current information in the 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.

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. This can be done by selecting individual files, or the content of an entire folder as seen in figure 10.10

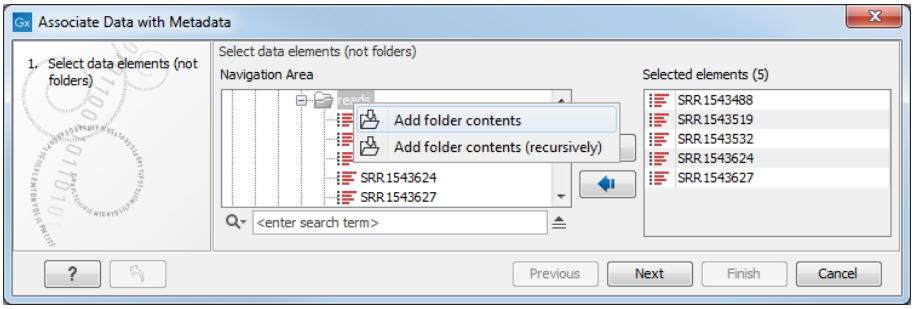


Figure 10.10: *Selecting all data elements in a folder.*

Specify a role that should be assigned to each data element that is associated to a metadata row (figure 10.11). The role can be anything that describes the data element best.

Select whether the matching of the data element names to the entries in the key column should be based on exact or partial matching.

Choose to **Save** the outputs. Data associations and roles will be saved for data elements where the name matches a key column entry according to the selected matching scheme.

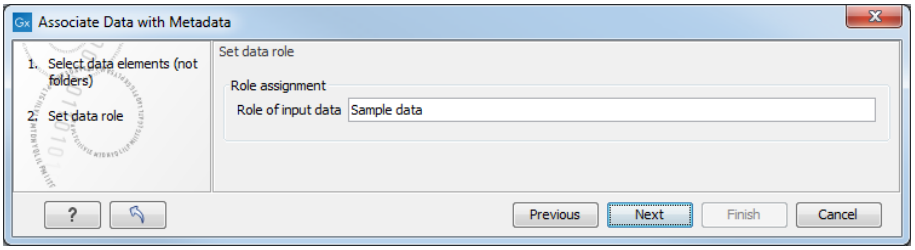


Figure 10.11: *Provide a role for the data elements. The default role provided is "Sample data".*

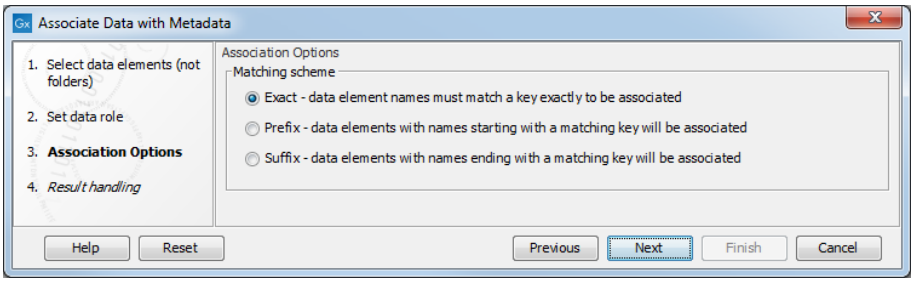


Figure 10.12: *Data element names can be matched either exactly or partially to the entries in the key column.*

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 they key entry had been `Sample23`, no match would be identified, because they 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 10.13) (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

10.2.2 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 10.13).

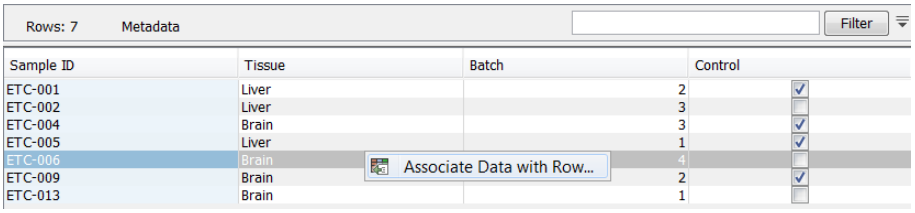


Figure 10.13: *Manual association of data elements to a metadata row.*

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.

10.3 Working with data and metadata

10.3.1 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 10.14).

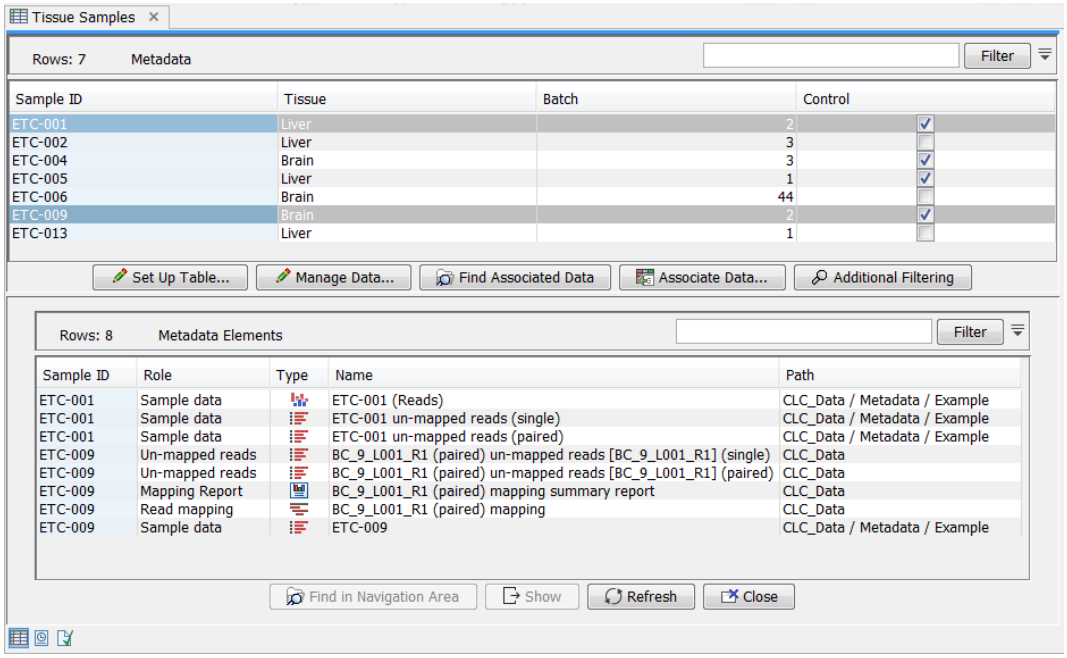


Figure 10.14: *Metadata Table with search results*

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 10.14, 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.

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 Toolbox, and navigate to the tool of interest. The data selected in the search results table will be listed as selected elements in the Wizard that appears.
- Via the Navigation area selection. Use the **Find in Navigation Area** button and then launch a tool in the Toolbox. The items that were selected in the Navigation area will be pre-selected in the Wizard that is launched.

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 http://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=Rebuilding_index.html.

10.3.2 Viewing metadata associations

Metadata associations for a data element are shown using the Element info view, as in figure 10.15.

To show Element Info,

right-click an element in the Navigation Area | Show | Element Info ( )

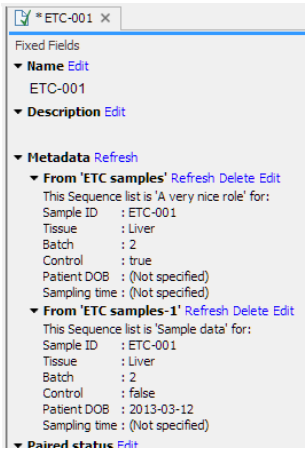


Figure 10.15: *Element Info view with a metadata association*

The Element Info view contains the details of each metadata association for the data element. The following operations are available:

- **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 12.4.

10.3.3 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 10.16). Alternatively, use the Delete key on the keyboard, or on a Mac, the fn and backspace keys at the same time.

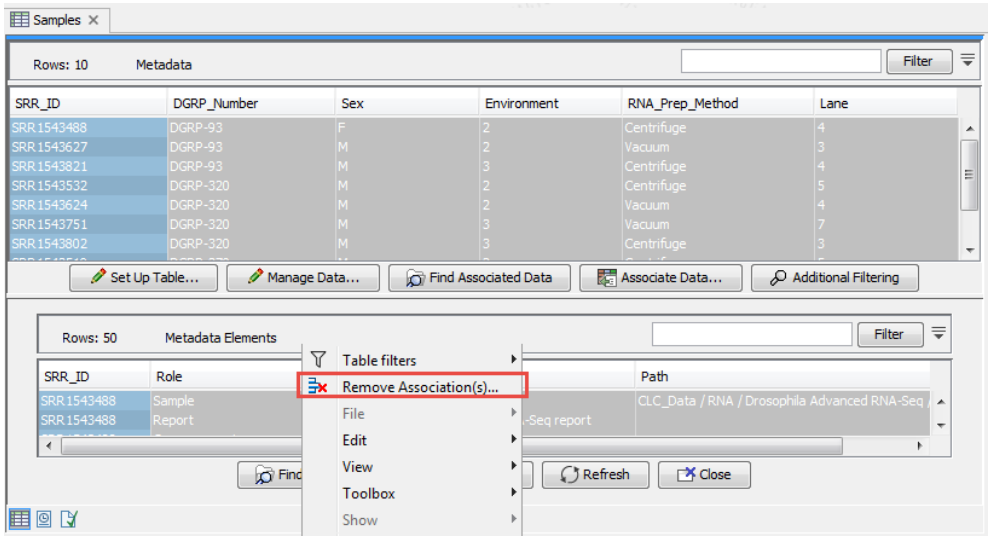


Figure 10.16: *Removing metadata associations to two data elements via the Metadata Elements table.*

Metadata associations can also be removed from within the Element info view for individual data elements, as described in section 10.3.2.

When an metadata association is removed from a data element, this update to the data element is automatically saved.

10.3.4 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 10.2.

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.

10.4 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 10.2.

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 6.6 for more information.

10.5 Editing Metadata tables

The **Metadata Table Editor** can be used to edit a metadata table. To open the Metadata Table Editor, open a metadata table already imported and saved in the Navigation Area, and click on **Set Up Table...**. For each column, it is possible to change the following:

- **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**. It is possible to designate any column as the key column, as long as entries in that column are unique.
- **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.

  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**.

The ( ) and ( ) buttons are used undo and redo changes respectively.

Columns may be deleted using the ( ) button. After metadata has been imported, additional columns can be added to the table structure. This can be done by importing the altered structure from an external file, where any columns not already in the metadata table will be added. Alternatively, individual columns can be added using the ( ) and ( ) buttons, which insert new columns before and after the current column respectively. Row information is added manually by clicking on the ( ) button and typing in the information for each column.

To edit information about samples, click on **Manage Data...**, and navigate between rows as explained above about columns.

Chapter 11

Workflows

Contents

11.1 Creating a workflow 216
11.1.1 Adding elements to a workflow 216
11.1.2 Connecting workflow elements 218
11.1.3 Ordering inputs 219
11.1.4 Validating a workflow 220
11.2 Editing existing workflows 221
11.2.1 Snippets in workflows 221
11.2.2 Workflow visualization 224
11.3 Workflow elements 229
11.3.1 Anatomy of workflow elements 229
11.3.2 Workflow element coloring 229
11.3.3 Basic configuration of workflow elements 230
11.3.4 Configuring input and output elements 231
11.3.5 Track lists as workflow outputs 234
11.3.6 Input modifying tools 235
11.4 Launching workflows individually and in batches 237
11.4.1 Importing data on the fly 238
11.4.2 Workflow outputs and workflow result metadata tables 240
11.4.3 Running workflows in batch mode 241
11.4.4 Batching workflows with more than one input changing per run 243
11.5 Batching part of a workflow 246
11.5.1 Iterate 246
11.5.2 Collect and Distribute 247
11.5.3 Running part of a workflow multiple times 248
11.6 Advanced workflow batching 250
11.6.1 Multiple levels of batching 251
11.6.2 Splitting paths in a workflow 252
11.6.3 Matching up inputs with each other and analyzing them together later in the workflow 254
11.7 Managing workflows 255

11.7.1 Updating workflows 257
11.7.2 Creating a workflow installation file 260
11.7.3 Installing a workflow 263

The *CLC Genomics Workbench* provides a framework for creating, running, distributing and installing workflows. A workflow consists of a series of connected tools where the output of one tool is used as input for another tool, making it possible to analyze many samples using a standardized pipeline. Once a workflow is created, it can be installed on your CLC Workbench or *CLC Genomics Server*, and it can be shared with colleagues to install on their CLC software.

11.1 Creating a workflow

Workflows are created and edited using the Workflow Editor, where workflow elements are added, connected and configured. In this section, we cover the basics of adding and connecting elements. Configuration of workflow elements is covered in section 11.3.

To open 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 ( )**

11.1.1 Adding elements to a workflow

Elements can be added to a workflow various ways:

- Drag tools directly from the **Toolbox** in the bottom left panel of the Workbench into the canvas area of the Workflow Editor.
- Use the **Add Element** dialog, illustrated in figure 11.1. There are multiple ways to open this dialog with all workflow elements listed:
  - 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 you wish to add to the workflow and click **OK**. Multiple elements can be selected by keeping the Ctrl key (⌘ on Mac) depressed when selecting elements.
- Use one of the relevant options offered when right-clicking on an input or output channel of a workflow element, as shown in figure 11.2 and figure 11.3.
  - **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.

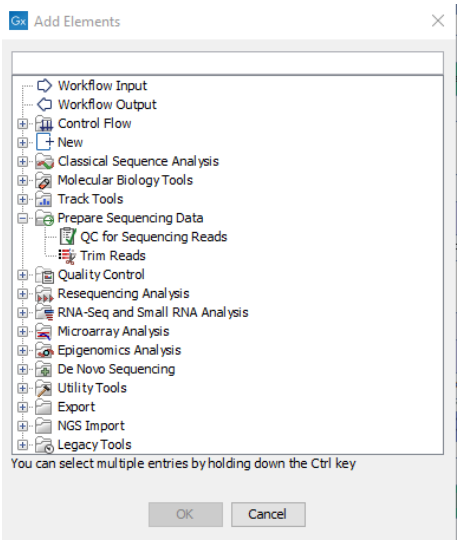


Figure 11.1: *Adding elements to a workflow.*

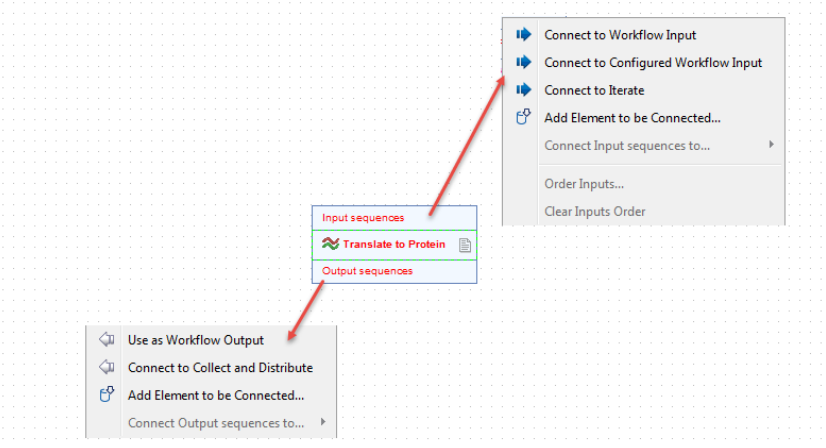


Figure 11.2: *Connection options are shown in menus when you right click on an input or output channel of a workflow element.*

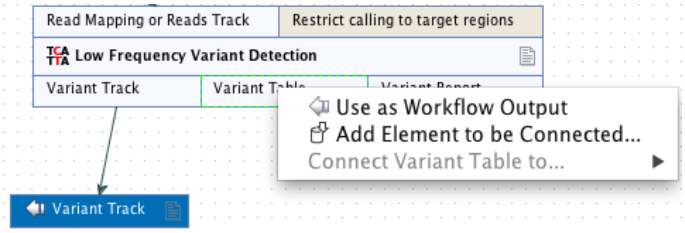


Figure 11.3: *Right clicking on an output channel brings up a menu with relevant connection options.*

– **Connect to Iterate** and **Connect to Collect and Distribute** to create an iterative process within a workflow.

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 11.4.

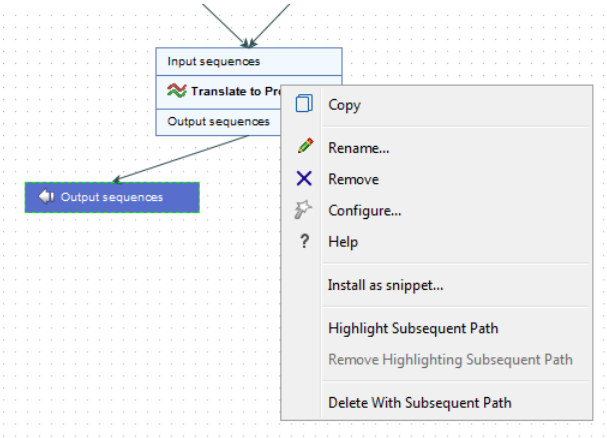


Figure 11.4: *Right clicking on an element name brings up a context specific menu that includes options for renaming or removing elements.*

11.1.2 Connecting workflow elements

Setting up connections between elements is the key to controlling where data flows. Elements are connected via their output channels and input channels. These channels are type dependent: only those with a matching type can be connected. Hovering the mouse cursor over an input or output channel will show a toolip with information like the data type required.

Compatible input and output channels can be connected in various ways:

- Click on an output channel, keep the mouse button depressed, and move 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, as shown in figure 11.5.

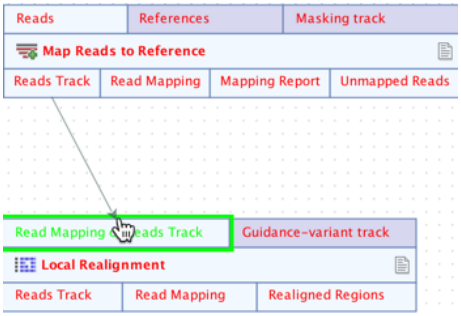


Figure 11.5: *Connecting the "Reads Track" output channel from a Map Reads to Reference element to the "Read Mapping or Reads" input channel of a Local Realignment element.*

- Use the **Connect <channel name> to...** option in the right-click menu of an input or output 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, as shown in figure 11.6.

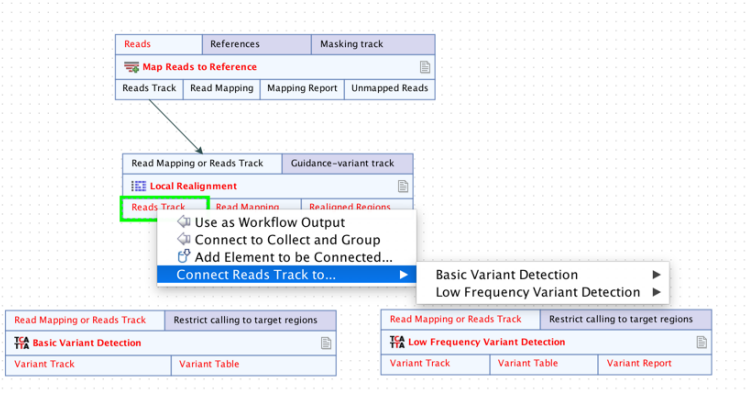


Figure 11.6: *Right-clicking on an output channel displays a context specific menu, with options supporting the connection of this channel to input channels of other workflow elements.*

A given output channel can supply its data to multiple other workflow elements, and workflow elements can accept data as input from more than one output channel. This is illustrated in figure 11.7.

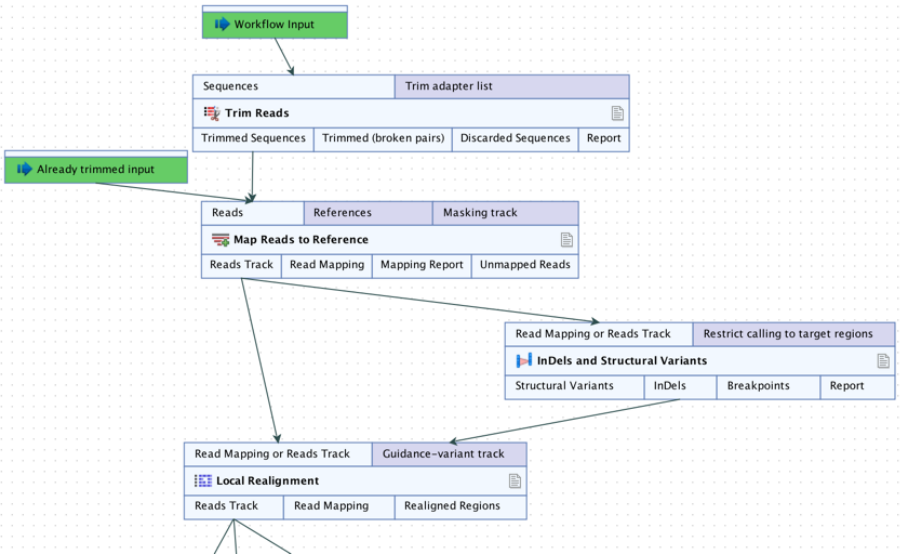


Figure 11.7: *Input channels can receive multiple connections, and multiple connections can be configured from output channels. Here, two elements are supplying data to the Reads input channel of the Map Reads to Reference element and data from the Reads Track output channel of Map Reads to Reference is being used as input to two elements.*

Input and output channels are discussed further in section 11.3

11.1.3 Ordering inputs

Two different ways to order workflow inputs are available. In both cases, an Order Inputs dialog like that shown in figure 11.8 is used. An item is moved up and down in the list by selecting it and clicking on the up or down arrow on the right.

Workflow inputs can be ordered by using:

- **Order Workflow Inputs...**. Right click on any blank area on the Workflow Editor canvas to bring up a menu with this option. This sets the order of the wizard steps prompting for the relevant input data when the workflow is launched. This ordering is reflected by a number in front of the Workflow Input element name in the workflow design. That number can also be used when configuring output element naming patterns, as described in section 11.3.4.
- **Order Inputs...**. Right click on an input channel with more than one connection to it to bring up a menu with this option enabled. This is the order that inputs to this input channel should be processed. This ordering is reflected by a number on the arrow connecting to the input channel. This is particularly useful when considering data visualization. For example, when connecting inputs to a Track List element, this is the order the constituent tracks will be in. This is described further in section 11.3.5.

See figure 11.9 for an illustration of the effects of these 2 ordering methods.

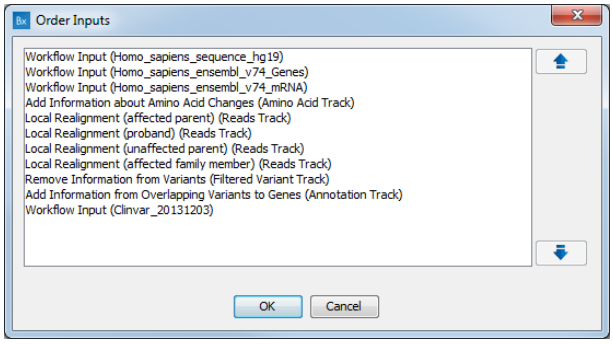


Figure 11.8: *The Order Inputs dialog is used to specify the ordering of workflow inputs.*

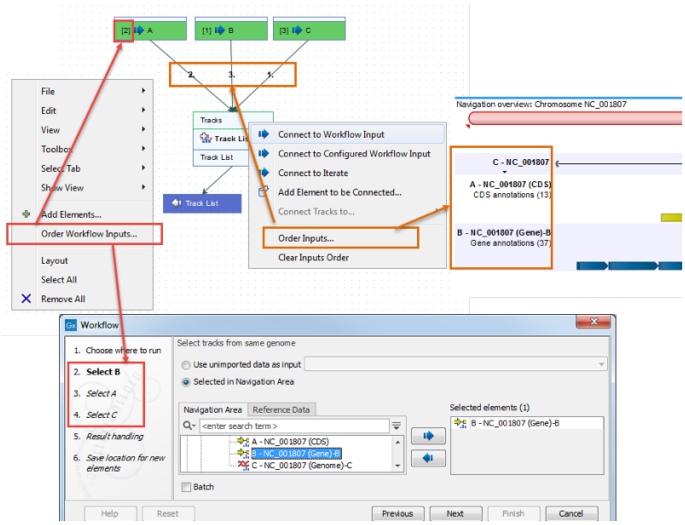


Figure 11.9: *Two levels of input ordering are available. Using Order Workflow Inputs..., the order that inputs are prompted for in the wizard was set to B, A, C. Using Order Inputs..., the order the inputs were processed was set to C, A, B. Here, this means the tracks will appear in the Track List in the order C, A, B.*

11.1.4 Validating a workflow

At the bottom of the view, there is a text with a status of the workflow (see figure 11.10). It will inform about the actions you need to take to finalize the workflow.



Figure 11.10: *A workflow is constantly validated at the bottom of the view.*

The following needs to be in place before a workflow can be executed:

- All input boxes need to be connected either to the workflow input or to the output of other tools.
- At least one output box from each tool needs to be connected to either a workflow output or to the input box of another tool.

Additional checks are done to verify that the workflow is valid, and specific warning messages will inform of eventual inconsistencies. The validation may contain several lines of text, and it is possible to scroll to see all lines. If one of the errors pertain to a specific element in the workflow, clicking the error will highlight this element.

Once these have been addressed, the status will be "Validation successful". Clicking the **Run** button will enable you to try running a data set through the workflow to test that it produces the expected results. If reference data has not been configured (see section 11.3.3), there will be a dialog asking for this in the workflow launching wizard.

11.2 Editing existing workflows

Double clicking on a workflow listed in the **Navigation Area** opens it in the Workflow Editor. Similarly, right-clicking on a workflow in the **Toolbox** and choosing the option **Open Copy of Workflow** (as in figure 11.11) will open a copy of the workflow in the Workflow Editor. Once opened, further elements can be added and configured, as desired.

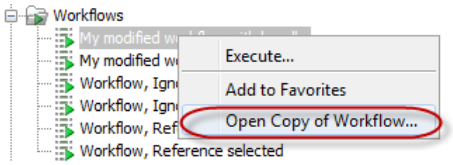


Figure 11.11: *Open a workflow in the Workflow Editor.*

Modified workflows must be saved if you wish to use them later. The workflow shown in figure 11.12 has not yet been saved, which is noted by the asterisk in the tab name and the message in red text below the workflow. If the workflow had been successfully saved, the text under the canvas would be in green text saying "Validation successful".

11.2.1 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 11.13.

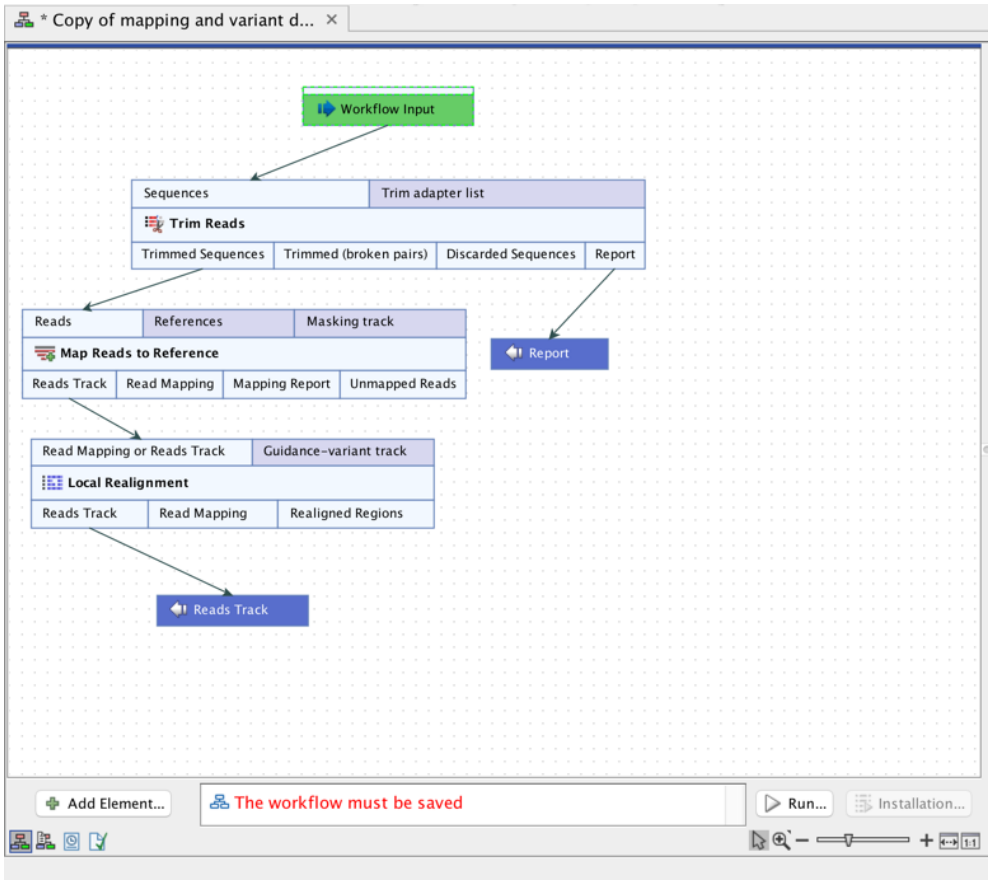


Figure 11.12: *This copy of an installed workflow is available for editing in the Workflow Editor. It has not yet been saved for later use, as indicated by the asterisk in the tab name and the warning text below the workflow.*

When you have clicked on "Install as snippet" the dialog shown in figure 11.14 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 11.15)

Right-clicking on the installed snippet in the **Side Panel** will bring up the following options (figure 11.16):

- **Add**. Adds the snippet to the current open workflow
- **View**. Opens a dialog showing the snippet, which allows you to see the structure
- **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.

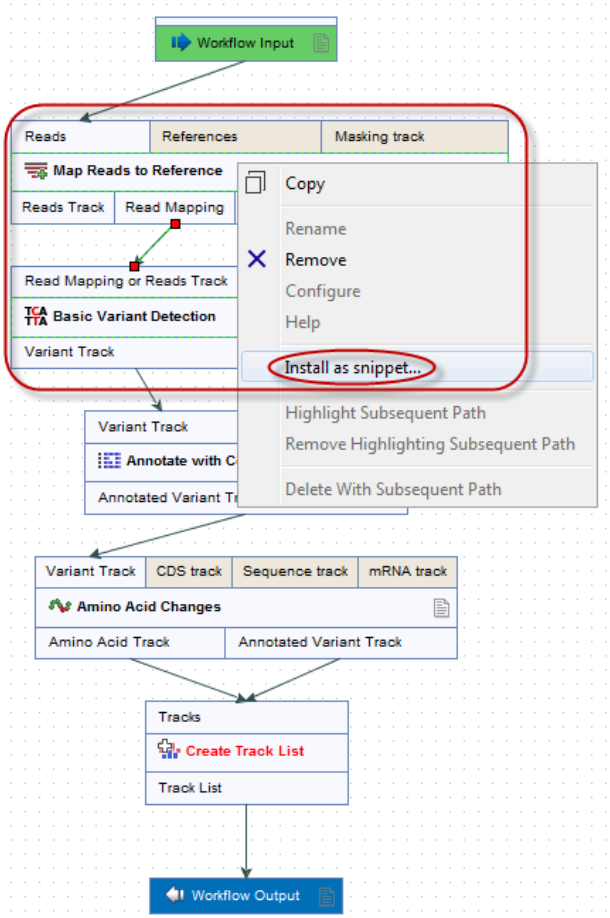


Figure 11.13: *The selected elements are highlighted with a red box in this figure. Select "Install as snippet".*

- **Migrate**. Updates the snippet (if required).

If you right-click on the top-level folder you get the options shown in figure 11.17:

- **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 11.18.

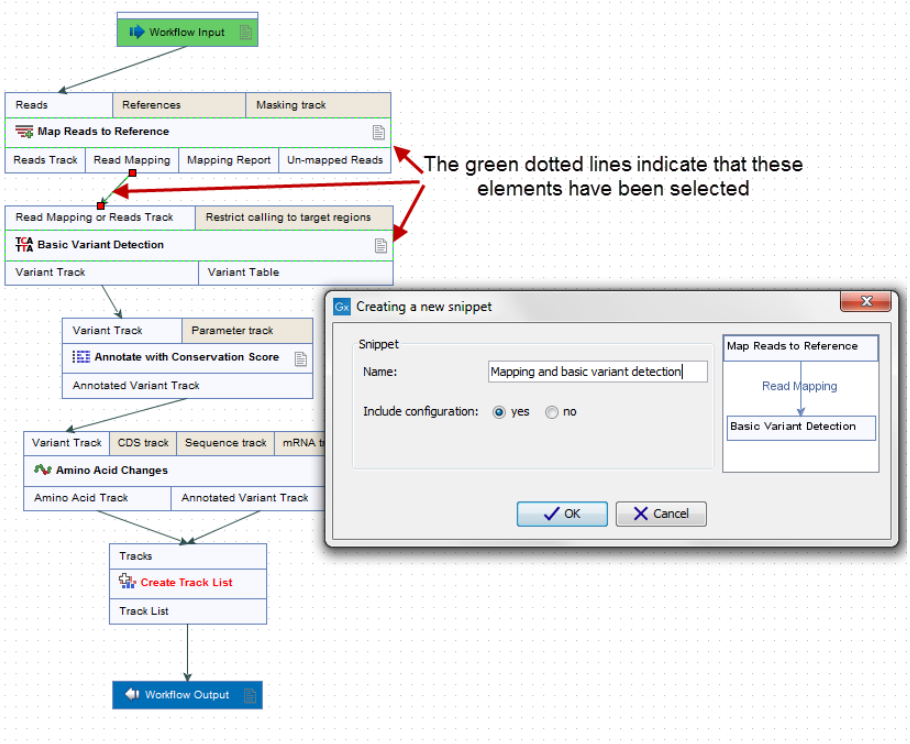


Figure 11.14: *In the "Create a new snippet" dialog you can name the snippet and select whether or not you would like to include the configuration. In the right-hand side of the dialog you can see the elements that are included in the snippet.*

11.2.2 Workflow visualization

Layout The workflow layout can be adjusted automatically, with right-clicking anywhere in the canvas and choosing the option "Layout" (figure 11.19), or with the quick command Shift + Alt + L. Note that only elements that have been connected will be adjusted.

It is very easy to make an image of the workflow. Simply select the elements in the workflow (this can be done pressing Ctrl + A, by dragging the mouse around the workflow while holding down the left mouse button, or by right clicking in the editor and then selecting "Select All"), then press the Copy button in the toolbar ( ) or CTRL + C. Press Ctrl + V to paste the image into the wanted destination, such as an email or a text or presentation program.

Configuration Editor Instead of configuring the various tools individually, the **Configuration Editor** enables the specification of all settings, references, masking parameters etc. through a single wizard window (figure 11.20). This editor is accessed through the ( ) icon located in the lower left corner.

Side Panel In the workflow editor **Side Panel**, you will find the following workflow display settings (figure **??**):

- **Minimap**. Use to navigate really large and complex workflows.
- **Find**. Counts how many times a search term is present in the workflow. Clicking on **Find** highlights the instances by changing the perimeter of the element from a full line to a dash

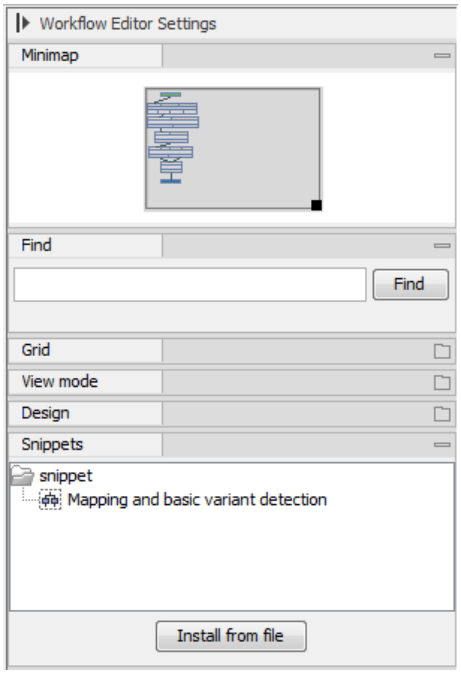


Figure 11.15: *When a snippet is installed, it appears in the Side Panel under the "Snippets" tab.*

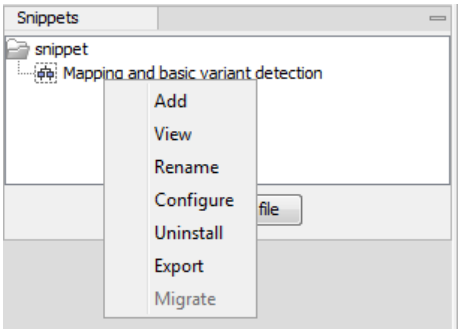


Figure 11.16: *Right-clicking on an installed snippet brings up a range of different options.*

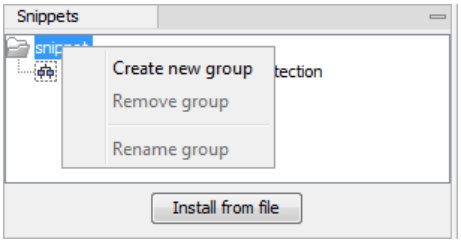


Figure 11.17: *Right-clicking on the snippet top-level folder makes it possible to manipulate the groups.*

line.

- **Grid**. Displays a grid on the canvas, and customizes the spacing, style and color of the grid. Per default, a grid is shown, and the workflow elements snap to the grid when they are moved around.
- **View mode**
  - **Collapsed**. Collapses elements of the workflow for simplified visualization of the workflow (useful for large workflows).

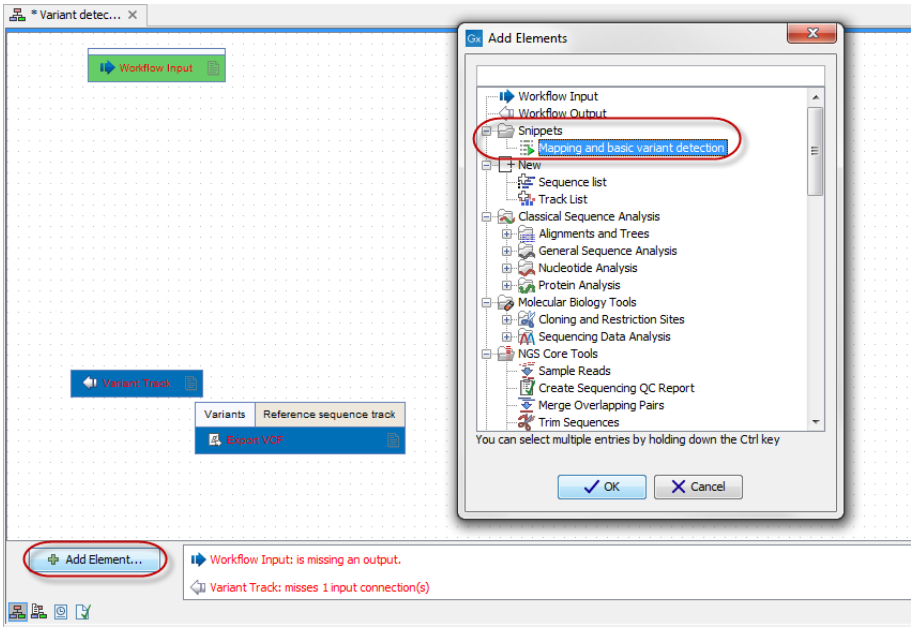


Figure 11.18: *Snippets can be added to a workflow in the workflow editor using the 'Add Element' button found in the lower left corner.*

  - **Highlight used elements**. Grays out unused elements (also possible with shortcut Alt + Shift + U).
  - **Rulers**. Adds vertical and horizontal rules on the canvas.
  - **Auto Layout**. Rearranges automatically newly added and connected elements.
  - **Connections to background**. Places connecting arrows are shown behind elements to make it easier to read elements and parameters names.

- **Design**
  - **Round elements**. Rounds corners of element boxes.
  - **Show shadow**. Adds shadows under element boxes.
  - **Coloring**. Customizes background color (see section 11.3.2) for a description of the different categories).
  - **Connection style**. Customizes connection arrows.

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 11.21). Select **Remove Highlighting Subsequent Path** to remove the highlighting.

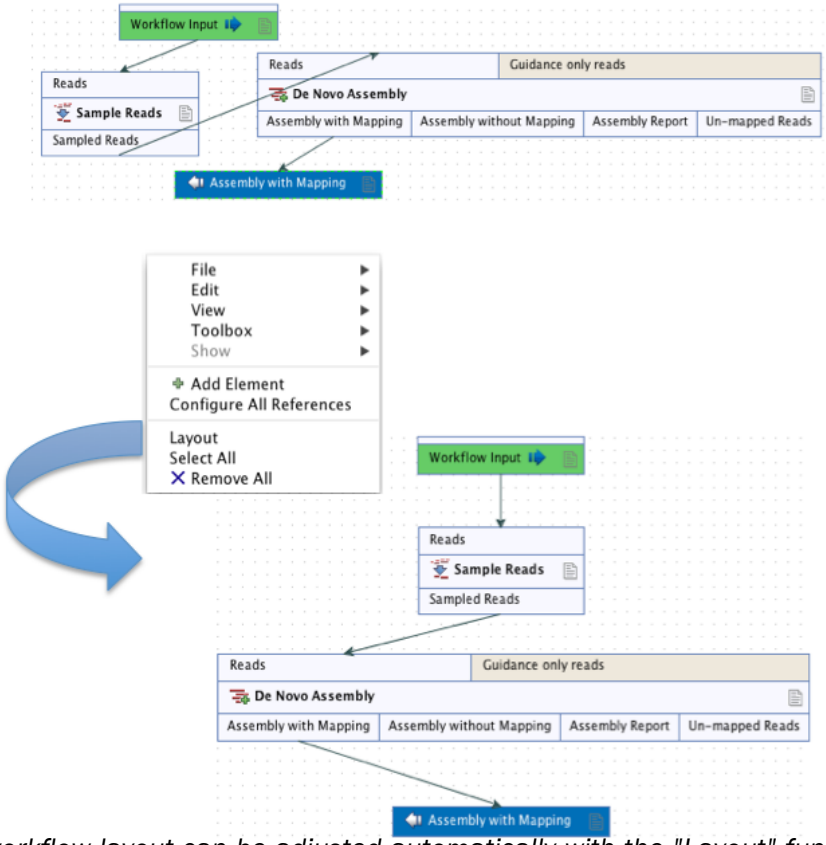


Figure 11.19: *A workflow layout can be adjusted automatically with the "Layout" function.*

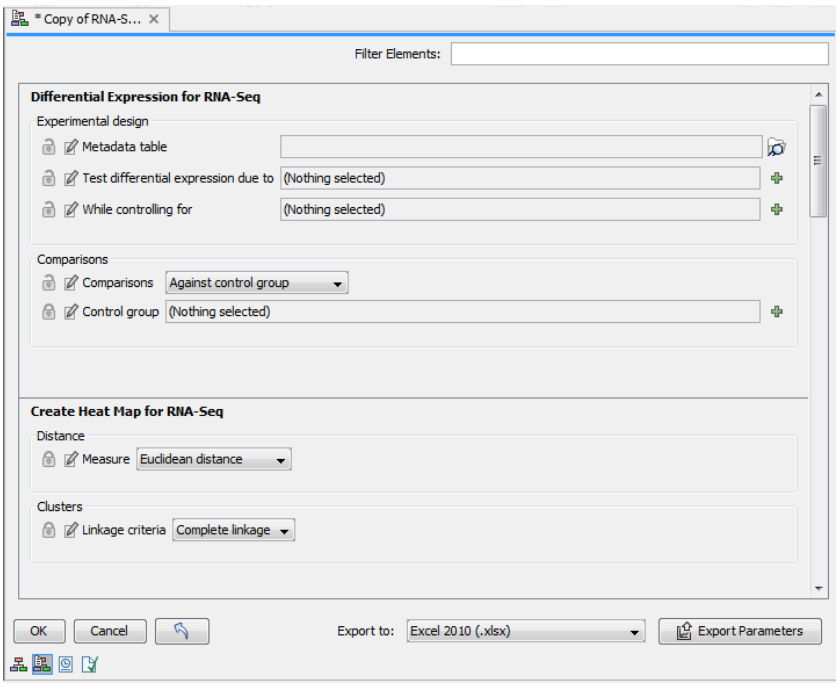


Figure 11.20: *Use the Configuration Editor to edit configurable parameters for all the tools in a given Workflow.*

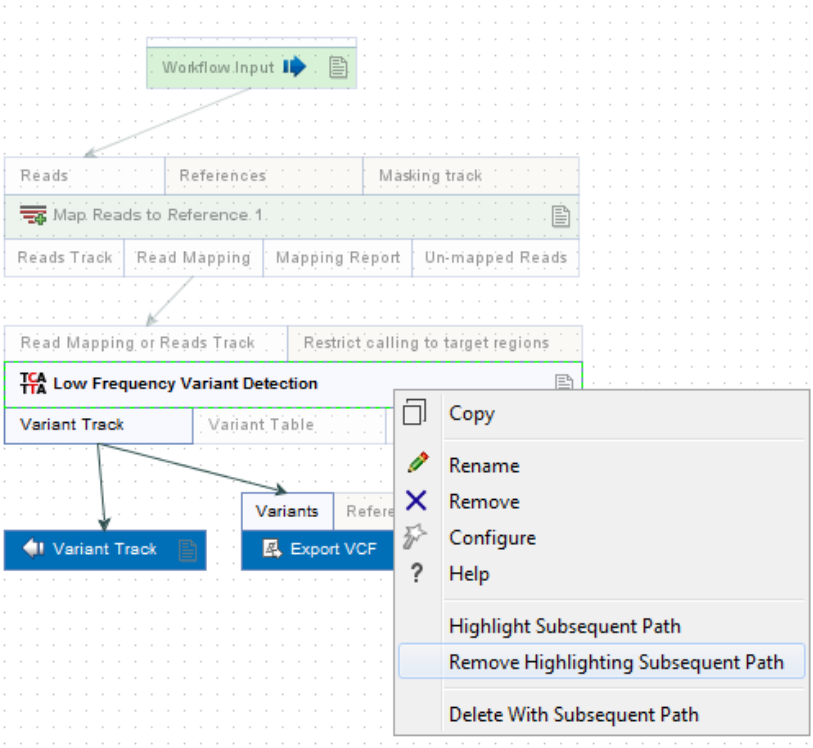


Figure 11.21: *All elements connected downstream of a selected element are highlighted after selecting the Highlight Subsequent Path menu option.*

11.3 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.

11.3.1 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 11.22). A page symbol on the right hand side of the middle section of a workflow element indicates the element can be configured.



Figure 11.22: *Anatomy of an element.*

11.3.2 Workflow element coloring

Elements background is color coded (figure 11.23) according to the following categories:

- Elements that take data into a workflow - and thus need to be specified before starting a workflow run - are bright green (for data to ba analyzed) or light purple (when input data is considered a configurable parameter, such as reference data for example).
- Elements that cause data to be saved onto a disk, either as CLC data or as data exported to another format, are dark blue.
- Input and Output channels that move data within a workflow are light grey.
- Elements are also light grey, unless they have been configured to different parameter values than the ones specified by default, in which case they become purple.
- Elements used for fine tune control of the execution of whole workflows or sections of workflows are dark green. Control flow elements are described in detail in section 11.5.

Note that the background color can be changed using the Workflow editor side panel, and that the color indicated here are the ones assigned to each category be default.

When adding a new element to a workflow, its text will appear red until it has been properly connected to other elements of the workflow.

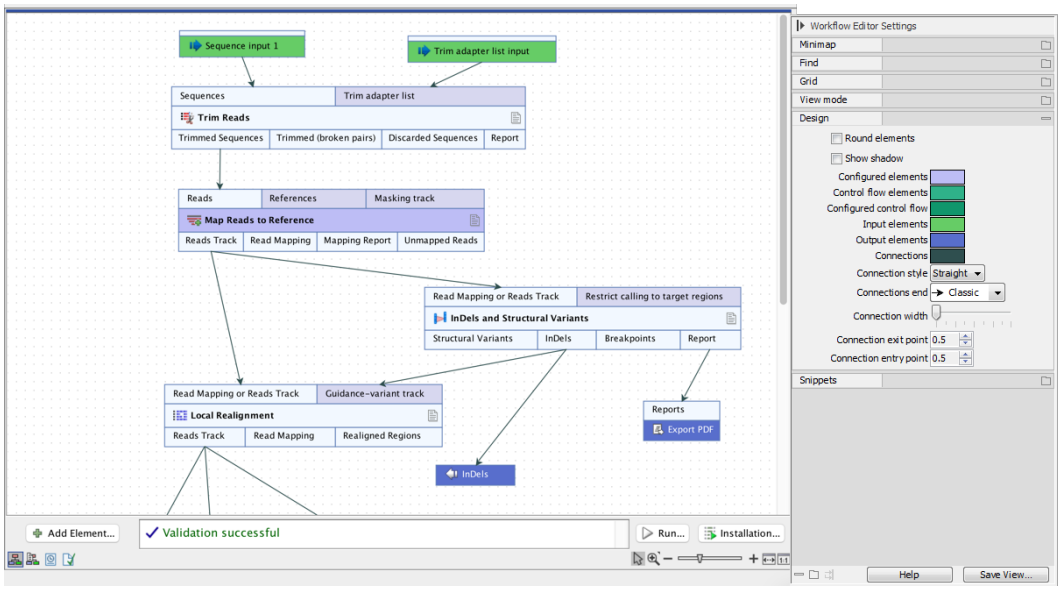


Figure 11.23: *An element's color indicates the role it plays and its state. Here, Trim Reads means is using only default parameter values, whereas the purple background for Map Reads to Reference indicates that one or more of its parameter values have been changed. The green elements are Workflow Input 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.*

11.3.3 Basic configuration of workflow elements

Four general types of changes can be made to workflow elements. These affect how the workflow wizard looks when launched or how the workflow behaves when run. Changes are accessible form the right-click menu on an element name, using the options Rename or Configure.

- **Renaming the element**. Element names are used in the wizard when the workflow is launched. Naming an element well can thus make the workflow easier to use. This is particularly the case if the original element name is very generic, or if there are multiple instances of elements of the same type. 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.

- **Locking or unlocking parameters**. A small lock icon is present before configurable parameters in the configuration wizard. Clicking on the lock icon changes it from locked ( ) to unlocked ( ) or vice versa. Parameters that are unlocked will be available for configuration in the wizard launching the workflow. Conversely, parameters that are locked will not be visible in the wizard, and thus cannot be edited, when the workflow is launched. Locking parameters can help ensure that a workflow is run with the same values or reference data inputs each time it is run.

- **Changing the name of a parameter**. Parameters that have been renamed appear with their new name in the wizard when the workflow is launched. To edit a parameter name, click on the edit icon ( ) in the configuration wizard and enter a new name. The parameter must be unlocked to be renamed, but can be locked again after configuration.

- **Changing the values of parameters**. Parameters have been assigned a default value, and unless configured otherwise, this is the value that will be used when the workflow is

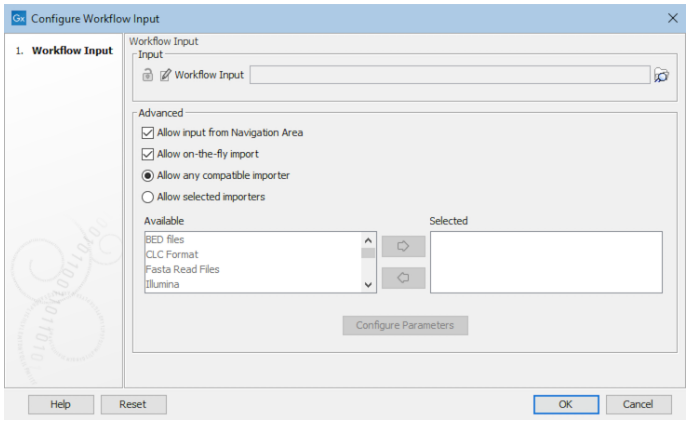


Figure 11.24: *Each parameter can be configured. An unlocked symbol beside the parameter indicates that the option will be shown and be configurable in the wizard when launching the workflow.*

run. Detailed information about default parameters for tool elements, including importers and exporters, can be found in the manual section for the specific tool. Configuration of Workflow Input and Workflow Output elements is discussed in more detail in section 11.3.4.

11.3.4 Configuring input and output elements

Configuring Workflow Input elements

Workflow Input elements and NGS Import elements are the two element types that bring data into a workflow.

At least one such element must be present in a workflow. By default, when a workflow is launched, the workflow wizard will prompt for data to be selected from the Navigation Area, or for data files to be imported on-the-fly using any compatible importer, as described in section 11.4.1.

If desired, Workflow Input elements can be configured restrict the options for data selection available when launching the workflow. To configure these elements, double click on them or right-click on an element name and select the **Configure...** option. This opens a dialog like that in figure 11.24.

Common configurations for Workflow Input elements include:

- **Configuring import options for primary inputs** Enable or disable selection of input data from the Navigation Area (already imported data) or import of raw data using on-the-fly import.

  When on-the-fly import is enabled, you can choose whether to restrict the available importers and import settings:

  - **Allow any compatible importer** When selected, all compatible importers are displayed as options when launching the workflow. All parameters of these importers will be unlocked, and thus will be available to configure when launching the workflow.
  - **Allow selected importers** When selected, one or more importers can be specified. Click on the **Configure Parameters** button to open the "Configure parameters" dialog, where the available import options can be configured for each selected importer. Select the importer to configure from the drop-down list at the top of this dialog. Each option can be locked if desired.

- **Configuring a parameter input with reference data** When a Workflow Input element is connected to a parameter input channel for reference data, a particular data element can be selected. If the parameter is then locked, the workflow will always run using this reference data. If the parameter is left unlocked, the default is to use that reference data element, but a different element could be selected when launching the workflow.
- **Configuring a parameter input with reference data from a Reference Data Set** Specify a workflow role in the **Workflow role** field (figure 11.25) to use the data element with that role assigned to it within a Reference Data Set. You can either type in a role name or choose one from the drop down list, as long as there is a Reference Data Set with a matching role. The **Workflow role** field is available when a Workflow Input element is connected to a parameter input channel. Reference Data Sets are described in more detail in sections 8.2 and 8.3.

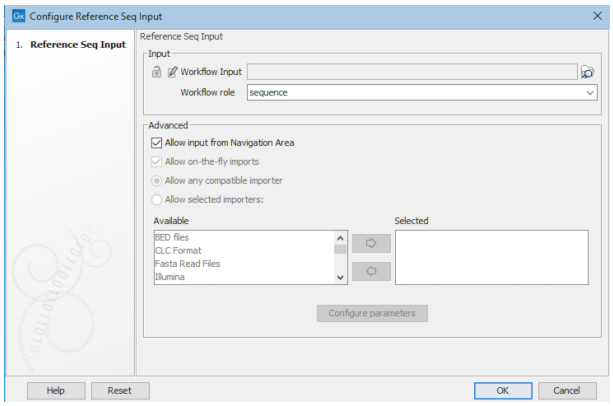


Figure 11.25: *A workflow role has been configured in this Workflow Input element. When launching this workflow, a Reference Data Set would be prompted for by the wizard. The data element with the specified role in that Reference Data Set would then be used as input.*

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.

The naming pattern for workflow outputs and exports can be specified by configuring Workflow Output elements and Export elements respectively. To do this, double click on a Workflow Output or Export element, or right-click and select the option **Configure...**. Naming patterns can be configured in the **Custom output name** field in the configuration dialog.

The rest of this section is about configuring the **Custom output name** field, with a focus on the use of placeholders. This information applies to both Workflow Output elements and Export elements. Other configuration settings for Export elements are the same as for export tools, described in section 6.6.2. Placeholders available for export tools, run directly (not via a workflow) are different and are described in section 6.6.3.

Configuring custom output names

By default, a placeholders is used to specify the name of an output or exported file, as seen in figure 11.26. Placeholders specify a type of information to include in the output name, and are a convenient way to apply a consistent naming pattern. They are replaced by the relevant information when the output is created.

The placeholders available are listed below. Hover the mouse cursor over the **Custom output name** field in the configuration dialog to see a tooltip containing this list. Text-based forms of the placeholders are not case specific.

- **{name}** or **{1}** - default name for the tool's output
- **{input}** or **{2}** - the name of the first workflow input (and not the input to a particular tool within a workflow). To include additional input names, use for example **{2:1}-{2:2}**. In this case, the workflow output will be a concatenation of the two first inputs. (See section 11.1.3 to learn more about input order).
- **{metadata}** or **{3}** - the batch unit identifier for workflows executed in batch mode or workflows that contain Iterate elements. Depending on how the workflow was configured at launch, this value may be 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}**.
- **{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}** - name of the user who launched the job
- **{host}** - 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 filenames.

You can choose any combination of the placeholders and text, including punctuation, when configuring output or export names. For example, `{input}({day}-{month}-{year})`, or `{2} variant track` as shown in figure 11.27. In the latter case, if the first workflow input was named `Sample 1`, the name of the output generated would be "Sample 1 variant track".

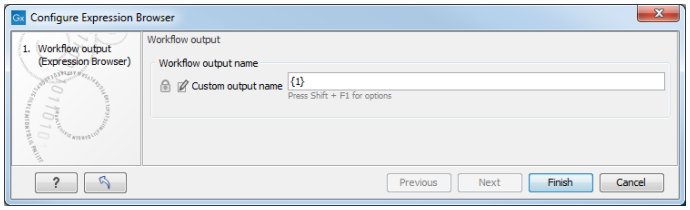


Figure 11.26: *The names that outputs are given can configured. The default naming uses the placeholder {1}, which is a synonym for the placeholder {name}.*

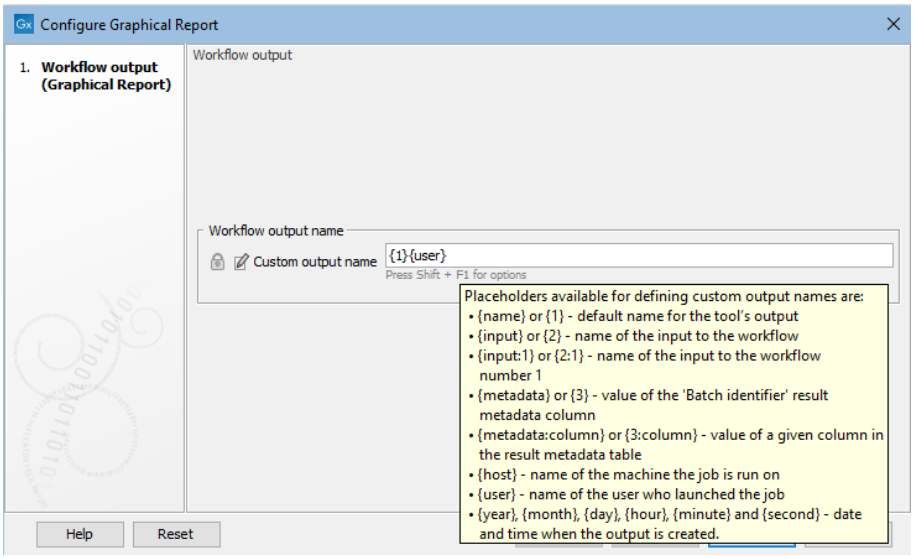


Figure 11.27: *Providing a custom name for the result.*

It is also possible to save workflow outputs and exports into subfolders by using a forward slash character `/` at the start of the output name definition. For example the custom output name `/variants/{name}` refers to a folder "variants" that would lie under the location selected for storing the workflow outputs. When defining subfolders for outputs or exports, all later forward slash characters in the configuration, except the last one, will be interpreted as further levels of 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. If the folders specified in the configuration do not already exist, they are created.

Note: In some circumstances, outputs from workflow element output channels without a Workflow Output element or an Export element connected may be generated during a workflow run. 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.

11.3.5 Track lists as workflow outputs

Track lists can be made and saved in workflows by doing the following (see figure 11.28:

- 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

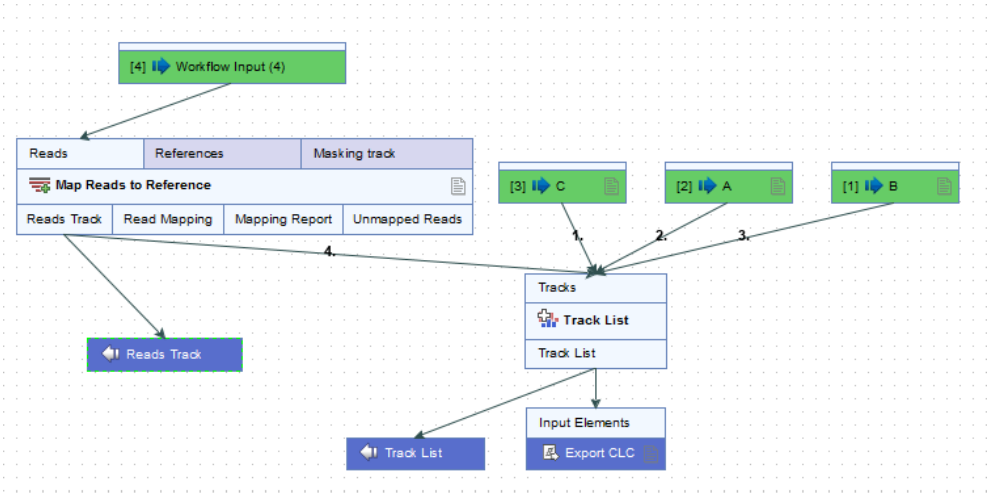


Figure 11.28: *This workflow will save and export a track list containing pre-existing data stored in a CLC data area, as well as data generated by the workflow. For workflows containing the Track List element to work, it is mandatory that the data generated by the workflow and included in the Track list is also saved as an independent track.*

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 11.1.2). 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.

11.3.6 Input modifying tools

An input modifying tool is a tool that manipulates its input objects (for example by adding annotations) without producing a new object. In a workflow, an input modifying tool is marked with the symbol (Ⓜ) (figure 11.29).



Figure 11.29: *Input modifying tools are marked with the letter M.*

There are 2 cases where it is not possible to save the Output of such a tool:

- When the input modifying tool is used in a branch(see figure 11.30), it cannot be guaranteed which workflow branch will be executed first, which in turn means that different runs can result in production of different objects. A message in red letters will appear saying "Branching before a modifying tool can lead to non-deterministic behavior", and the **Run** and **Create Installer** buttons are disabled.

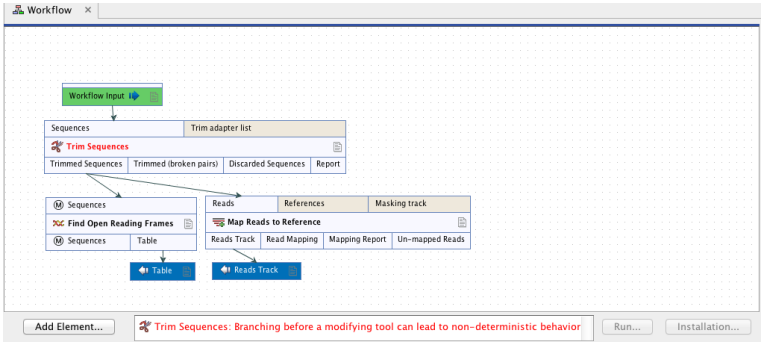


Figure 11.30: *A branch containing an input modifying tool is not allowed in a workflow.*

The problem can be solved by adding elements (with respect to order of execution) so that the branch disappears. This is shown in figure 11.31.

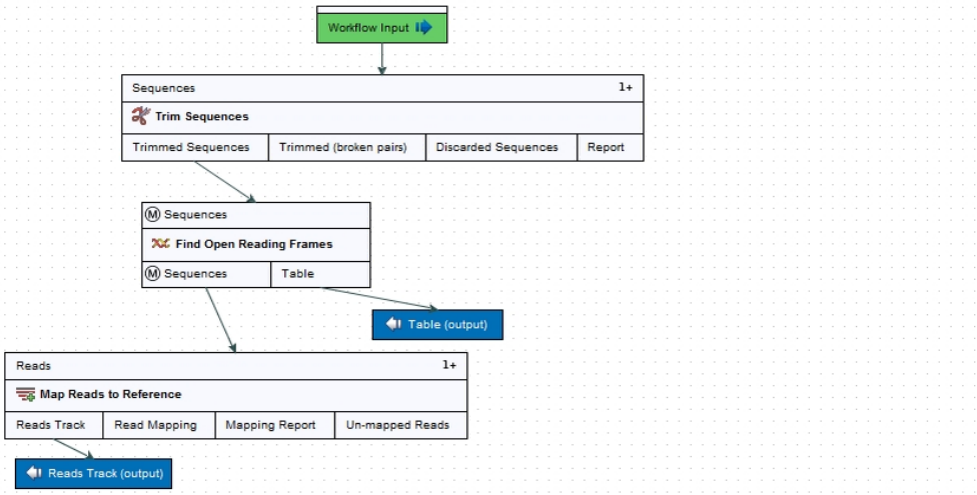


Figure 11.31: *A branch containing an input modifying tool has been resolved and the workflow can now be run or installed.*

- When a workflow is made of only one of these tools. To be able to save the output of such a tool, simply add an element before or after the first one (see figure 11.32).

  The output arrow is marked with "copy" to indicate that this is a copy of the result that is used as input at the next level in the chain (see figure 11.33).

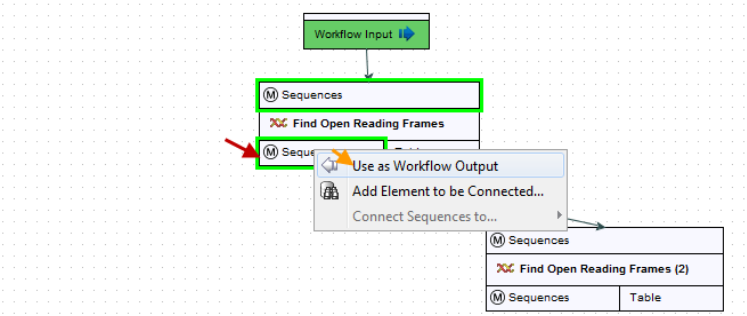


Figure 11.32: *A workflow output element can be added to an input modifying element when adding an element before or after the input modifying tool.*

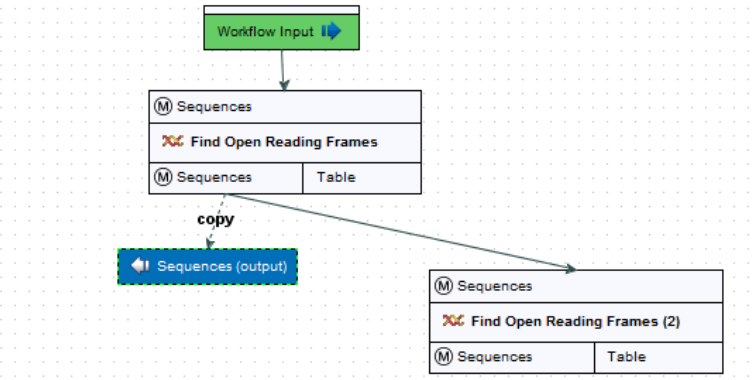


Figure 11.33: *This output is marked with "copy" to indicate that this is a copy of the result that is used as input at the next level in the chain.*

11.4 Launching workflows individually and in batches

Workflows can be launched:

- **From under the Toolbox menu** by double clicking on an installed workflow.
- **From a Workflow Editor** by clicking on the **Run Workflow** button at the bottom of the Workflow Editor.

  Double-clicking on a saved workflow design in the Navigation Area opens it in the Workflow Editor. Installed workflows can be opened in the Workflow Editor by selecting them in the Toolbox in the bottom, left side of the *CLC Genomics Workbench*, right-clicking and selecting the option **Open Copy of Workflow**.

When a workflow is launched, a wizard is opened. Stepping through this wizard, configurable parameters can be reviewed and changed as needed, and the locations that outputs should be saved under can be specified. Where input elements are configured with a role, the workflow launch wizard will require you to select (and potentially download) the relevant reference data.

If you are connected to a *CLC Genomics Server*, the wizard will include the option to run the workflow locally on the Workbench or on the server.

If the workflow is not properly configured, you will see that in the dialog when the workflow is launched[1].

[1] If the workflow uses a tool that is part of a plugin, a missing plugin can also be the reason why the workflow is not enabled. A workflow can also become outdated because the underlying tools have changed since the workflow

One of the outputs that can be generated by a workflow run is a workflow result metadata table. This metadata table contains a row for each of the outputs generated by the workflow, and each row is associated with the relevant data objects. Workflow result metadata tables are particularly useful for navigating to results of workflows run in batch mode, and are described further in section 11.4.2.

Some aspects of launching workflows are different to launching tools, and can be more efficient and flexible, including:

- Data can be imported on the fly, as the first action taken when launching a workflow, saving doing this as a separate activity.
- Batch units can be defined using metadata, or, for simple workflows, based on the organization of the data in the Navigation Area. For individual tools, batch units can only be defined based on data organization.
- Metadata can be read from an Excel file as an integrated part of launching workflows in batch mode. This can save much hands-on work.
- Data used for multiple workflow inputs can be changed for each batch run. For example, a single workflow could be launched in batch mode, where reads from some samples would be mapped to a particular reference, but reads from other samples would be mapped to other references.
- As well as running whole workflows in batches, workflows can be designed such that just parts will run in batches. Results from those batched sections can then follow different downstream paths in the workflow, if desired.

The following aspects relating to launching workflows are covered in this section:

- Importing data on the fly, as part of a workflow run: section 11.4.1
- Workflow result metadata tables: section 11.4.2
- Launching workflows in batch mode: section 11.4.3
- Launching workflows with more than one input that should change per batch run: section 11.4.4.
- Running sections of a workflow in batches: section 11.5

11.4.1 Importing data on the fly

NGS sequence data can be imported on the fly, as an initial action when a workflow is run, avoiding the need import the data prior to launching the workflow. There are two ways that this can be done:

- Choose the option "Select files for import" when launching a workflow, as shown in figure 11.34.

was created (see section 11.7.1)

- Include an NGS import element in the workflow.

The two choices are illustrated in figure 11.35.

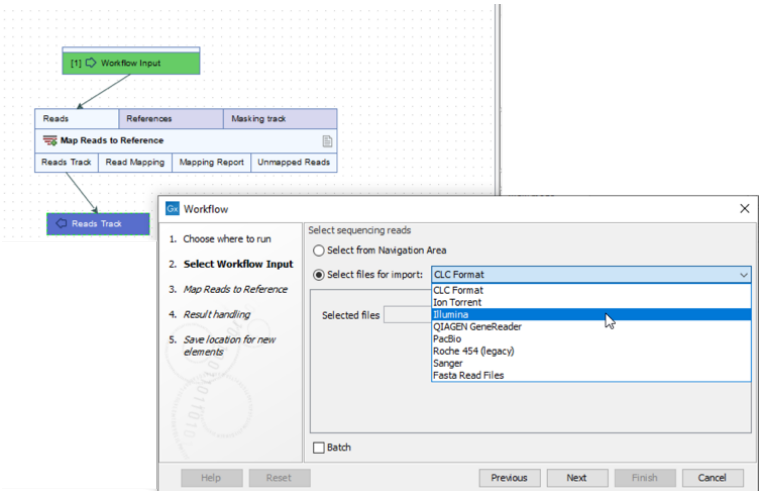


Figure 11.34: *Files containing NGS sequence data can be selected as workflow inputs. Such data is imported on the fly when the workflow starts running.*

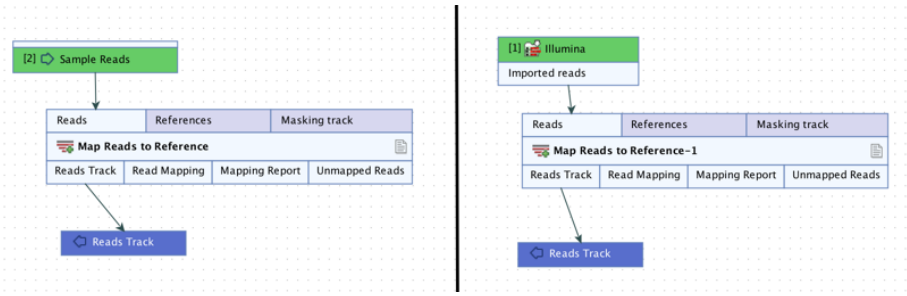


Figure 11.35: *NGS sequence data can be imported as the intial action in a workflow using either of the workflows pictured. The choice of using on-the-fly import, supported by the workflow on the left, or a dedicated NGS import element, as on the right, should take into account the general differences described in the text.*

Considerations when deciding between these two styles include:

- Workflows using NGS import elements cannot be launched in batch mode. Workflows using the on-the-fly import functionality can be run in batch mode, as described in section 11.4.3 and section 11.4.4.
- If an output element is connected to an NGS import element in the workflow design, the imported data will be saved. The imported data is *not* saved when using the on-the-fly import functionality.
- NGS import workflow elements can be configured with new defaults, and the workflow author can decide which values should configurable when launching the workflow. By contrast, all options for a given NGS importer are configurable by workflow users when using the on-the-fly importer functionality.
- A particular NGS importer must be selected when adding a dedicated worklow import element. Using the on-the-fly import functionality, the person launching the workflow can choose from any of the NGS importers.

Elements downstream of the initial import behave the same way however the import is done. So, for example, if you include a Trim Reads element in the workflow, you could add an output to its Trimmed Sequences output channel to save the sequence lists containing the trimmed data in both cases.

The configuration options available when importing NGS data are described in the Import chapter, specifically, section 6.3. These options are displayed slightly differently in the workflow wizard compared to the import tool wizard, but how the settings affect the import is the same.

In addition to NGS reads, Sanger trace files and CLC format files can also be imported as an initial action when a workflow is run. CLC format import is most likely to be of interest when receiving data from other CLC systems to use for an analysis.

When launching a workflow where the input elements involve data types other than sequence data, e.g. reads tracks, variant tracks, etc., the option to import from disk is still available, but here, only CLC format files are accepted at this time.

11.4.2 Workflow outputs and workflow result metadata tables

When launching a workflow, you can select where results should be saved in the same way as when running a single tool, as described in section 9.3.4. The results to be saved are those that have been sent to output elements (dark blue boxes), as described in section 11.3.4.

A workflow result metadata table, which contains information about the workflow outputs, can also be generated. This option is selected by default, as shown in figure 11.36. The workflow result metadata table is useful when navigating and investigating results generated by workflows.

Workflow result metadata tables

When a workflow is launched, a metadata table with information about the results can be generated, as shown in figure 11.36. In most circumstances, one workflow result metadata table is generated per workflow launch, with a batch run of a workflow generating one workflow result metadata table containing information about all the results. An exception occurs 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. In this case, one workflow result metadata table is generated per batch run and is saved in the relevant input folder. See section 11.4.3 for more information on launching workflows in batch mode.

Workflow result metadata tables contain one row for each output generated by the workflow, with associations in place between the workflow outputs and the relevant rows of the metadata table. A *Batch identifier* column will be included for workflows run in batch, as shown in figure 11.37. If the batch units were defined using metadata, the workflow result metadata table would include original metadata relevant to any of the outputs.

When an entire workflow is run in batch mode, every row in the metadata table will have a an entry in the *Batch identifier* column. Where only parts of the workflow are run in batches, only outputs generated in the batched sections will have an entry in that column. This is illustrated in figure 11.37. Running parts of workflows in batches is described further in section 11.5.

A Metadata Elements table can be opened beneath the metadata table by selecting rows of interest and clicking on the **Find Associated Data** button, as shown in figure 11.37. Finding and working with data associated to metadata rows is described further in section 10.3.1.

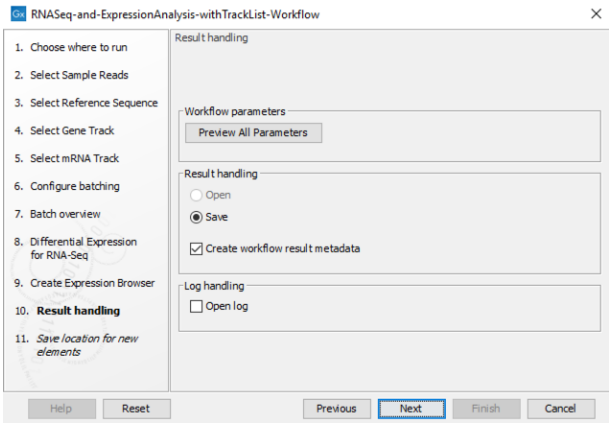


Figure 11.36: *The final step when launching a workflow includes an option to create a workflow result metdata table. This is enabled by default.*

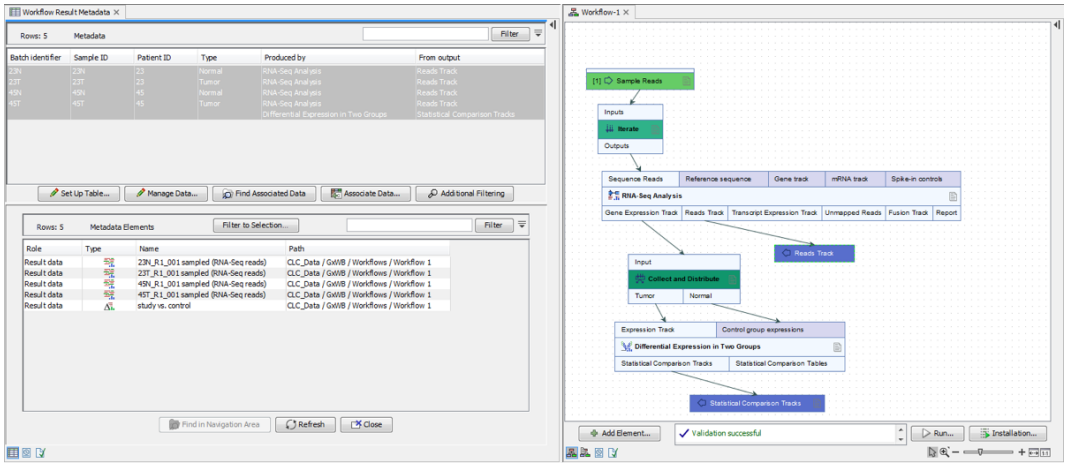


Figure 11.37: *The workflow result metadata table, top left, was generated from a run of the workflow on the right. Here, the RNA-Seq Anaylysis had been run in 4 batches. The results were then gathered and used as input to the Differential Expression in Two Groups tools, which was run once. There are thus 5 rows in the metadata result table in the top left. The RNA-Seq Analysis results each have a batch identifier, while the statistical comparison output does not.*

11.4.3 Running workflows in batch mode

Workflows can be run in batch mode by:

- Clicking in the Batch checkbox shown at the bottom of relevant workflow launch configuration steps.
- Running a workflow that contains one or more **Iterate** control flow elements.

In both cases, after selecting all of the inputs to be used for all batches, the grouping of the data into batch units must be defined.

In the simplest case, where just one workflow input uses different data in each batch run, batch units can be defined based on metadata, or they can be derived from the organization of the data in a CLC software area, as described for launching analysis tools in section 9.3.4.

For more complex scenarios, batch units are defined based on metadata, for example, when more than one workflow input uses different data in each batch run, as described in section 11.4.4, or

where just parts of the workflow should run in batches, as described in section 11.6,

Defining batch units based on metadata

When launching a workflow to run in batch mode, there are two formats metadata can be provided in: an Excel spreadsheet or a CLC metadata table.

1. Information in an Excel spreadsheet can be used to define batch units for any workflow launched in batch mode. The metadata in that file is imported into the CLC software at the start of the workflow run.

   The data is matched with the metadata based on the contents of the first column of the Excel file. That column must contain either the exact names of the data files or a unique prefix, that is, at least enough of the first part of the file name to identify it uniquely. If data is selected from the Navigation Area, the column contents are matched against the data element names. If data is imported on the fly, the column contents are matched against the names of the files being imported. The full file name can include file extensions, but not the path to the data.

   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`.

   Providing metadata in an Excel format file is often the most convenient route, and it is the only option available if you are importing the data using the on-the-fly functionality when the workflow is started.

2. A CLC metadata table with relevant data elements associated to it can be used when those data elements have been selected from the Navigation Area as inputs. How to create a metadata table is described in section 10.1.1.

Defining batch units based on the contents of an Excel format file is illustrated in figure 11.38. There, a workflow with a single input is being launched in batch mode. Eight files containing Illumina reads had been selected as input and the "Batch" checkbox ticked. In the step shown, the option to define batch units based on metadata is the only one available, as the data will be imported using the on-the-fly import functionality. An Excel format file containing metadata has been selected, and then the column SRR_ID from that file has been selected as the basis of the batch units.

In the next step, a preview of the batch units is shown. The workflow will be run once for each row shown in the left side of the preview, with the input data grouped as shown in the right hand column. See figure 11.39.

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. These are identical to those described in section 9.3.4.

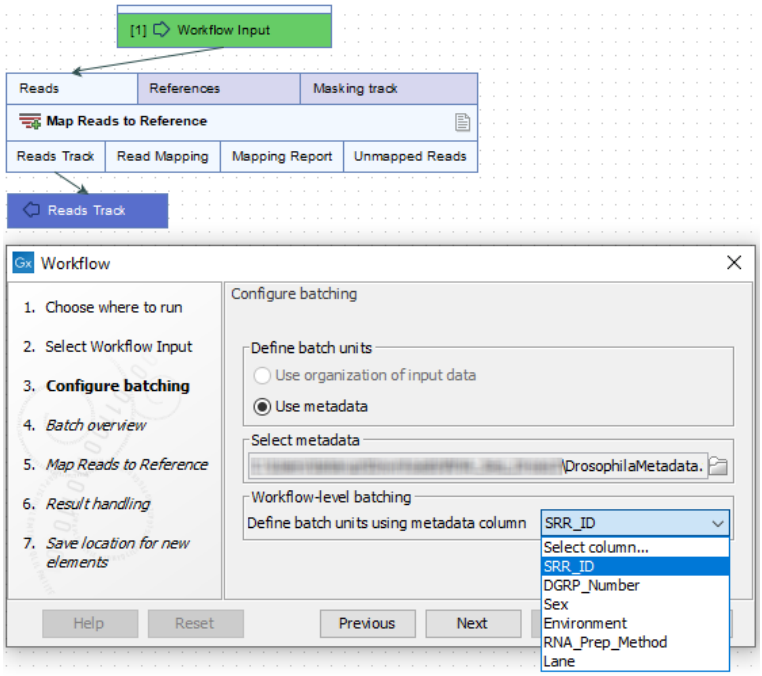


Figure 11.38: *Configuring batch units using metadata. As the data will be imported from external files, the metadata defining the batch units must also be imported from an external file, in this case, an Excel file with the names of the data files in the first column. A column with information defining the grouping of the samples for analysis, the batch units, is then selected. Here, that is the SRR_ID column.*

If the workflow contains Export elements, then an additional option, **Export to separate directories per batch unit**, is presented (figure 11.40). When this option is checked, the files exported from each batch run will be placed in separate subfolders under the export folder selected for each export step.

11.4.4 Batching workflows with more than one input changing per run

When a workflow contains multiple input elements (multiple bright green boxes), a Batch checkbox will be available in each of the wizard steps for selecting input data. Checking that box for a given input step indicates that the data for that input should change in each batch run. Data selected for inputs where the Batch checkbox is not checked are considered as a single set that should be used for that workflow input for all of the batch runs.

Where more than one input will change per batch run, batch units are defined using metadata. This is most easily explained using an example. Figure 11.41 shows a workflow with a Map Reads to Contigs element and two workflow input elements, Sample Reads and Reference Sequences. This workflow can be used to map particular sets of reads to particular references. In this example, the metadata is provided by two Excel files, one containing the information for the Sample Reads input data and one with information about the Reference Sequences input data.

The contents of Excel files that would work in this circumstance are shown in figure 11.42. Of particular note are:

- The first column of each of the Excel files contains the exact data file names for all the data that should be used for that input across all of the batch runs.

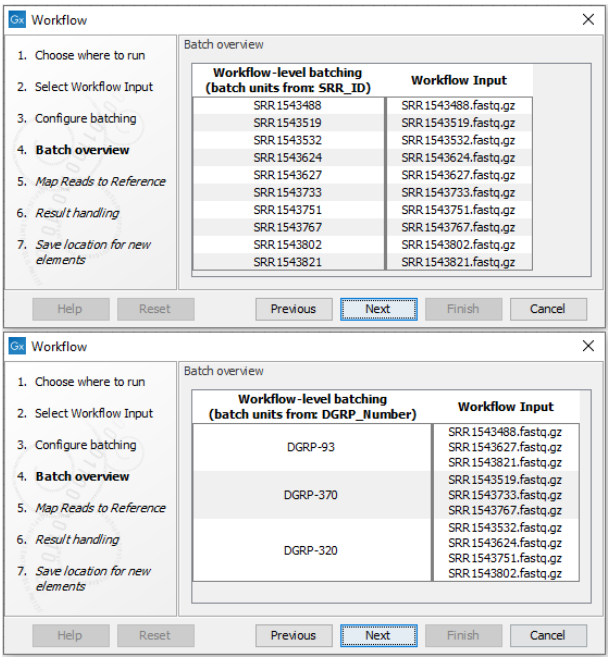


Figure 11.39: *The Batch overview step of the wizard allows you to review the batch units configured. In the top image, a column called SRR_ID had been selected, resulting in 8 batch units, so 8 workflow runs, with the data from one input file to be used in each batch. In the lower image, a column defining different batch units was selected. There, the workflow would be run 3 times with the input data grouped into 3 batches.*

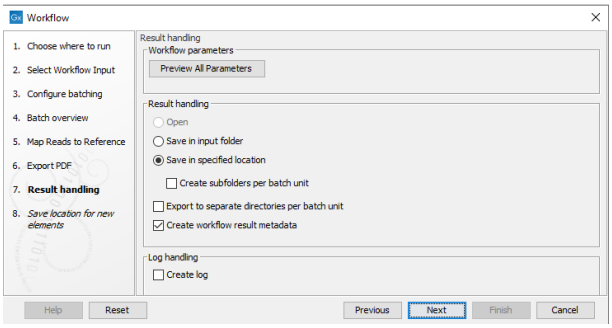


Figure 11.40: *Options are presented in the final wizard step for configuring where outputs and exported files from each batch run should be saved.*

- At least one column in each file has the same name as a column in the other file. That column should contain the information needed to match the Sample Reads input data with the relevant Reference Sequences input data for each batch run.

In the Workflow-level batch configuration area, the following are specified:

- The primary input. The input that determines the number of times the workflow should be run.
- The column in the metadata for that primary input specifying 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

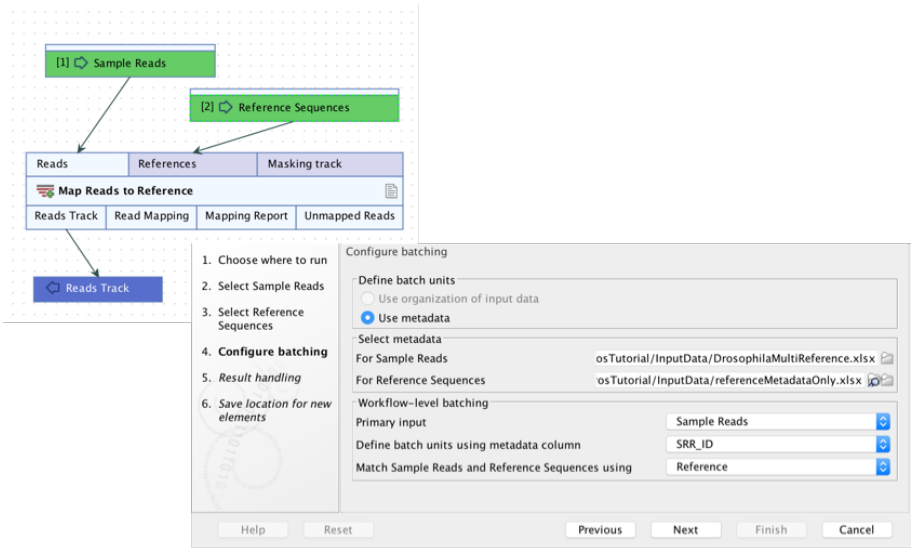


Figure 11.41: *A workflow with 2 inputs, where the Batch checkbox had been checked for both in the initial launch steps. Metadata is used to define the batch units since the correct inputs must be matched together for each run. Clicking on the plain folder icon brings up the option to import an external file, like an Excel file. The folder icon with the magnifying glass on it indicates that you can select an item from the Navigation Area, like a metadata table.*

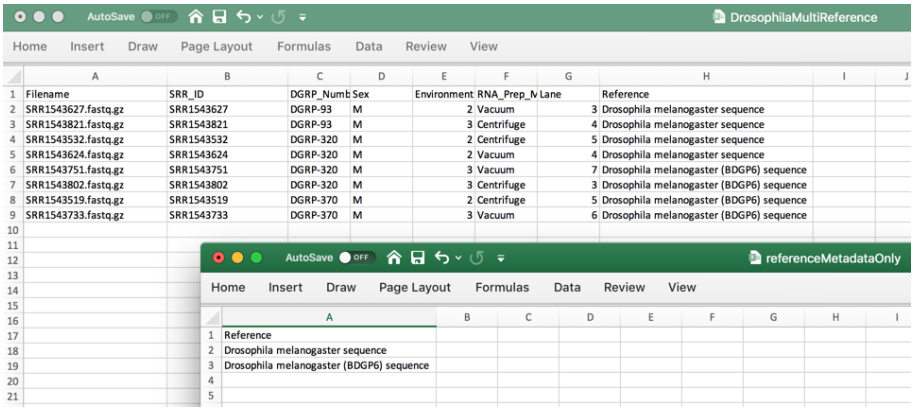


Figure 11.42: *Two Excel files containing information about the data to be used in each batch run for the workflow shown in figure 11.41. With the settings selected there, the number of batch runs will be based on the Sample Reads input, and will equal the number of unique SRR_ID entries in the DrosophilaMultiReference.xlsx file. The correct reference sequence to map to is determined by matching information in the Reference column of each Excel file.*

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 the example in figure 11.41, Sample Reads is the primary input: We wish to run the workflow once for each sample. We wish to run the workflow once for each SRR_ID entry, and the Reference sequence to use for each of these batch runs is defined in a column called Reference, which is present in both the Excel file containing information about the samples and the Excel file containing information about the references.

11.5 Batching part of a workflow

In the cases when only a part of the workflow should be batched, or when different inputs should follow different paths in the workflow, **control flow elements** can be added to the workflow.

Control flow elements control the flow of data through a workflow. Two control flow elements are available, **Iterate** and **Collect and Distribute**, and they can be found in the Control Flow folder of the Add Element wizard as shown in figure 11.43.

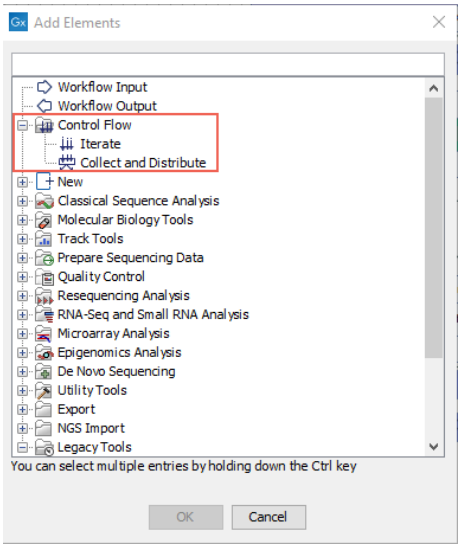


Figure 11.43: *The control flow elements Iterate and Collect and Distribute are found in the Control Flow folder of the Add Element wizard.*

The Iterate element is used to define a branch of a workflow that should be run multiple times, by splitting its inputs into groups (batch units). The Collect and Distribute can be used downstream of an Iterate element to collect all results of an iteration, and group them for collective analysis by further tools. Examples of workflows making use of these elements to address common bioinformatics analysis needs are provided in section 11.6.

Note: It is always possible to rename control flow elements so that the information presented in the workflow wizard, when launching the workflow, is easier to understand (see section 11.3.3).

For a minority of workflow designs containing control flow elements, it is possible that multiple copies of a particular output will be generated. If you are concerned about a particular workflow, please try running a test using a small dataset and investigating the outputs, either directly in the Navigation Area or via the workflow result metadata table.

11.5.1 Iterate

Adding an Iterate element to the top of a workflow causes the workflow branch below it to be run once for each batch unit. A given batch unit iteration runs until the end of the workflow, or until a Collect and Distribute element is encountered.

If desired, sections downstream of a Collect and Distribute element can also be run once per batch unit by adding another Iterate element, just after the Collect and Distribute element. The composition of batch units at this point in the workflow can be adjusted as desired.

Note: A single Iterate element at the top of a workflow without any downstream Collect and

Distribute element is equivalent to checking the Batch button when launching the workflow. In such cases, having a simple workflow design, without control flow elements, and checking the Batch button when launching is preferable.

For workflows with a single workflow input element (green box) that contain a single Iterate element, batch units can be defined based on the location of the input data or based on a information in a metadata table. For any workflow containing more than one **Iterate** element, or where there is a single **Iterate** element *and* the Batch button is checked when starting the workflow, batch units must be defined using information in a metadata table that the input data has been associated with.

For information on creating metadata tables and associating data with them, see section **??**.

Configuring an Iterate element The configuration options available for an Iterate element are shown in figure 11.44. They are:

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.
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 worfklow 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.

11.5.2 Collect and Distribute

When a Collect and Distribute element is encountered in a workflow, the intermediate results from all the iteration units prior to that point are gathered. Based upon metadata and configuration done when launching the workflow, these are distributed as required for downstream steps.

Note: Collect and Distribute elements are only relevant in workflows with upstream Iterate elements.

Configuring a Collect and Distribute element

The configuration of Collect and Distribute elements is shown in figure 11.45.

In the **Outputs** field of a Collect and Distribute element, terms are entered in a comma separated list. The number of terms determines the number of output channels from the Collect and

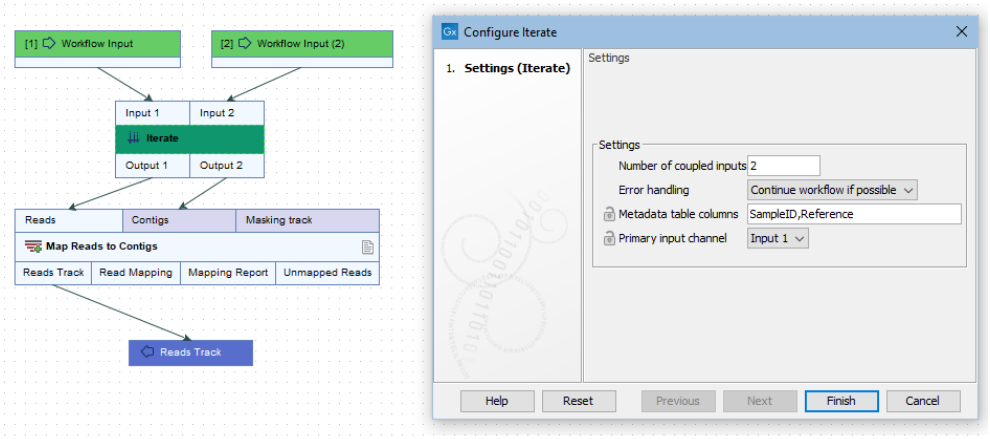


Figure 11.44: *The number of coupled inputs in this simple example is 2, allowing each set of sample reads to be mapped to a paticular reference, rather than using the same reference for all iterations.*

Distribute element. The connections made between output channels of Collect and Distribute element and input channels of downstream elements specify how those groups of inputs are distributed in the following stage of the workflow.

If the Collect and Distribute element has more than one output, then the path taken by each output is determined by the value in a particular column of the metadata provided when launching the workflow. This column can be pre-configured in the **Group by metadata column** setting.

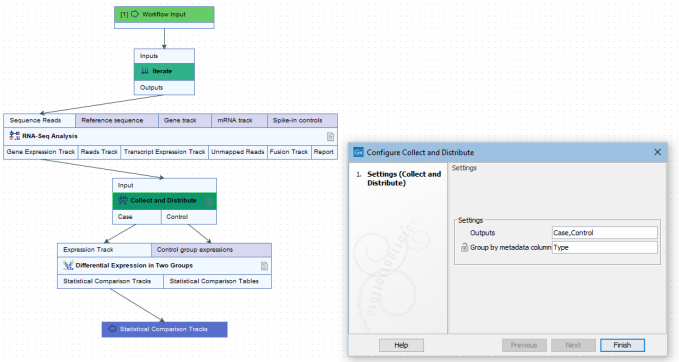


Figure 11.45: *A comma separated list of terms in the Outputs field of the Collect and Distribute element defines the number of output channels and their names.*

11.5.3 Running part of a workflow multiple times

To run a part of the workflow multiple times, enclose that part of the workflow between an *Iterate* and a *Collect and Distribute* element. For example, the workflow in figure 11.46 will allow you to run the RNA-Seq Analysis tool once per sample, and create a single combined report for the whole batch. All parts of the workflow that are downstream of an Iterate element will run multiple times, until a Collect and Distribute element is encountered. The parts of the workflow downstream of the Collect and Distribute element will be run only once.

When running on the server, the iterating parts of the workflow will run as separate jobs. This requires that the server is configured accordingly and job nodes or grid nodes are available.

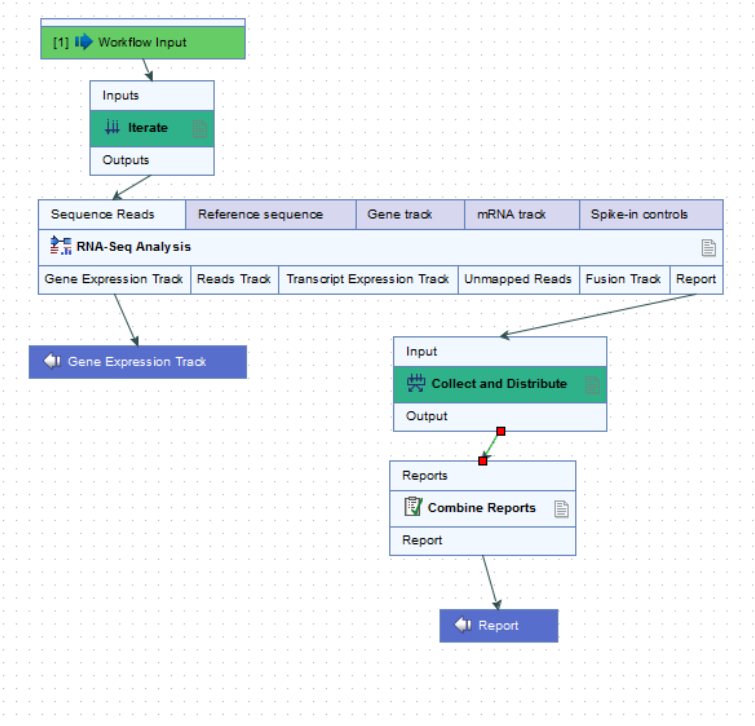


Figure 11.46: *In this workflow, the RNA-Seq analysis tool can be run once per sample, at the same time as creating a single combined report for the whole batch of samples.*

When running the workflow, you can use a metadata table to specify the iteration (batch) units. After selecting all samples in the first step (with the "batch" checkbox NOT selected), you can specify which column in the metadata table defines how the samples should be grouped. In the example in figure 11.47, grouping by the column "ID" will result in the RNA-Seq Analysis tool being run 8 times, once for each sample. Selecting the "Gender" column instead would result in the RNA-Seq Analysis tool being run 2 times, once for each value in that column (male and female). In both cases, the Combine Reports tool will run only once for all samples.

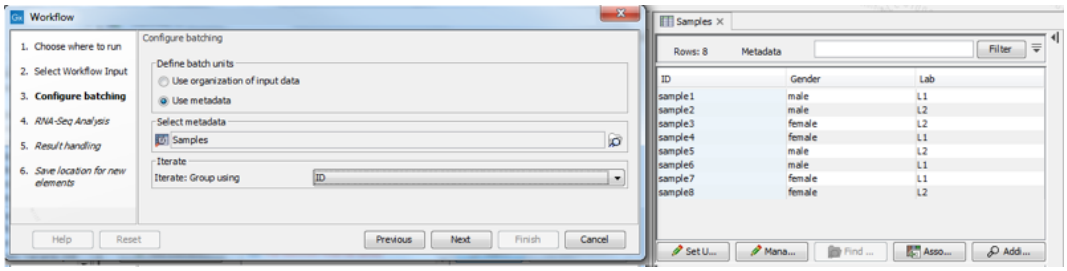


Figure 11.47: *With the current selection in the wizard, the RNA-Seq Analysis tool will run 8 times, once for each sample. The Combine Reports tool will only run once for all samples.*

It is possible to rename the Iterate element, which will also change the text displayed in the wizard when the workflow is run. To do this, right-click the Iterate element, and choose Rename. The new name of the element will now be displayed in the wizard (figure 11.48).

Control flow elements are described in more detail in section 11.5.

Defining batch units when using Demultiplex Reads

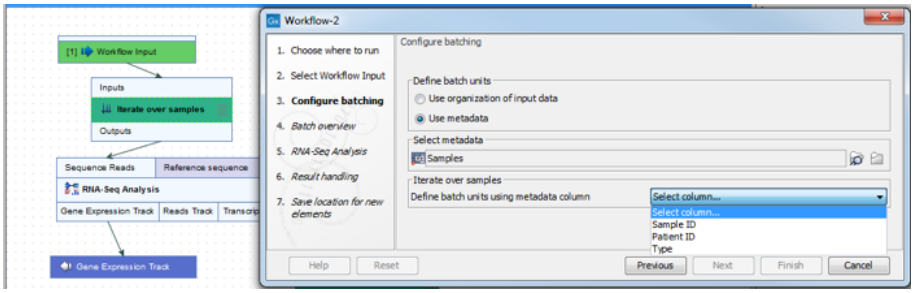


Figure 11.48: *The Iterate element can be renamed to change the text that is displayed in the wizard when running the workflow.*

When Demultiplex Reads is used in a workflow, the Group Sequences output channel is connected to an Iterate element (figure 11.49). Batch units for the iterating section of the workflow that follows, (Trim Reads, in figure 11.49), can be defined based on information provided in the barcode file imported to Demultiplex Reads, rather than a separate metadata table. For this, the CSV or Excel format file needs to contain a column with the barcodes, a column with the sample names, and further columns, containing the relevant metadata.

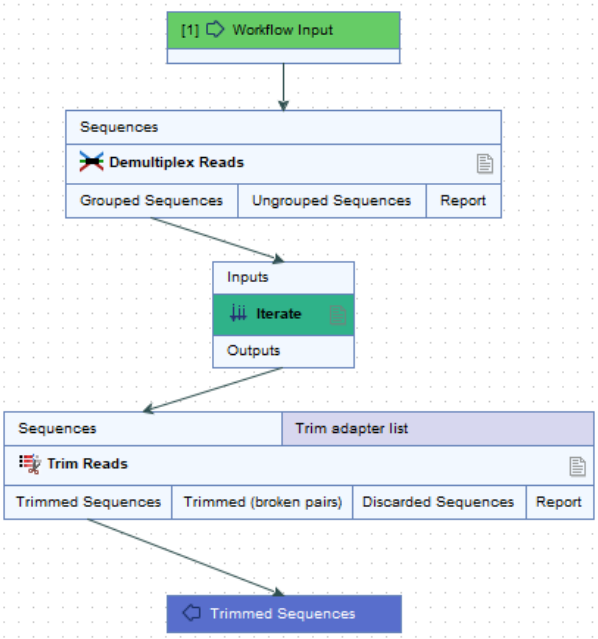


Figure 11.49: *The Group Sequences output channel of Demultiplex Reads connects to an Iterate element. The data to be analyzed together in the next workflow section, i.e. the batch units, can be defined using information from the barcode file or from a separate metadata table.*

11.6 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 11.5 and section 11.6.1.
  - **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. Design details are described in section 11.6.2. Running such workflows is described in section 11.5.3.

- **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 11.4.4. 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, as described in section 11.6.3.

11.6.1 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 11.50 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.

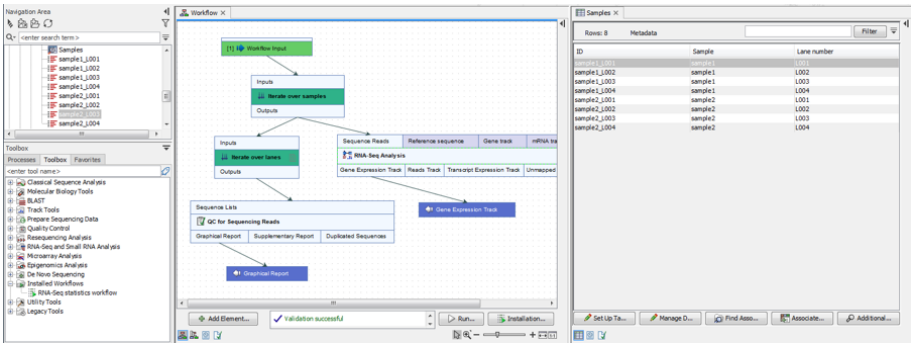


Figure 11.50: *The top-level Iterate element results in a subdivision (grouping) of the data, and the innermost Iterate results in a further subdivision (grouping) of each of those groups.*

When running the workflow, only metadata can be used to define the groups, because the

workflow contains multiple levels of iterations (figure 11.51).

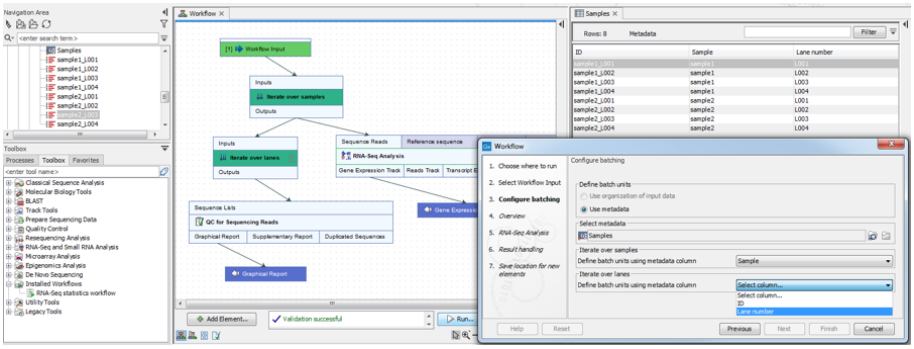


Figure 11.51: *When the workflow contains multiple levels of iterations, only metadata can be used to define the groups.*

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 11.5.

11.6.2 Splitting paths in a workflow

Different samples can be processed differently, either by following a different path, or playing a particular role in a downstream element of the same workflow. This can be achieved by configuring the Collect and Distribute element with multiple outputs.

To configure a Collect and Distribute element, double-click on it in the workflow, and enter the names of the outputs, separated by commas (figure 11.52). Inputs to the Collect and Distribute element are split into as many groups as there are output channels, and each of those groups can then be sent to different input channels of downstream elements. These can be different input channels of the same workflow element, or to different workflow elements.

Important note: the Collect and Distribute element always ends the ongoing iteration. To continue an iteration on a path after the data have been split, place a new Iterate element directly after the Collect and Distribute element.

In figure 11.52, a configured Collect and Distribute element splits the samples into cases and controls. Subsequently, each case sample is individually analyzed against all of the control samples. Note that one path from the Collect and Distribute element (the cases) continues the iteration, whereas the other path (the controls) does not.

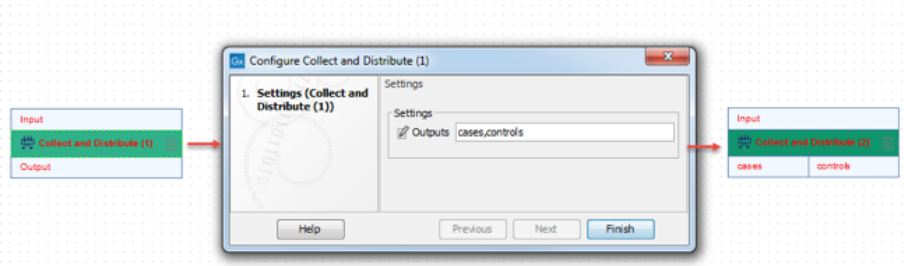


Figure 11.52: *Double-click the Collect and Distribute element to configure its outputs.*

For this workflow, the groupings for the Collect and Distribute element must be specified in

addition to specifying the iteration units. The groupings are coupled to the metadata table chosen for the corresponding Iterate element when defining the batch units. It is possible to group using a different column of the metadata table than that specified for the iteration units, as long as the groupings are compatible. For example, when running the workflow in figure 11.53, it is possible to select a metadata column called "Type" to split the samples into cases and controls, even though the iteration was over the "ID" column in the same metadata table (figure 11.54).

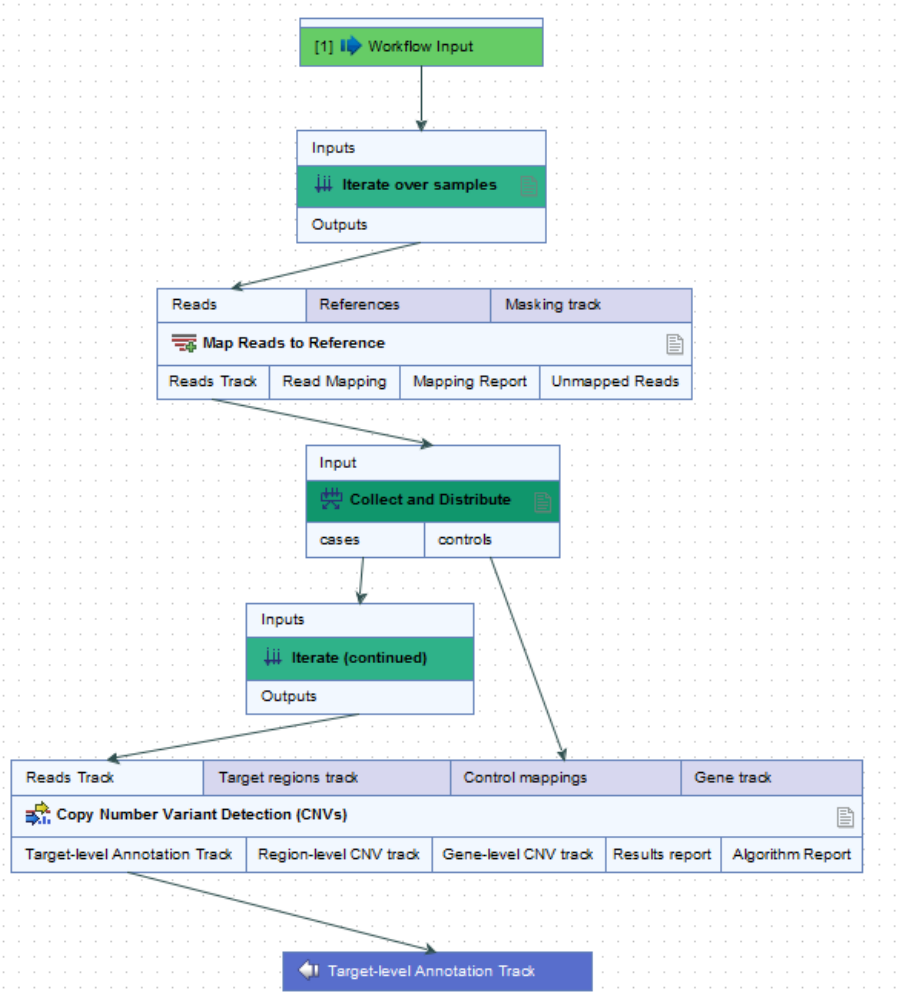


Figure 11.53: *In this workflow, each case sample is analyzed against all of the control samples.*

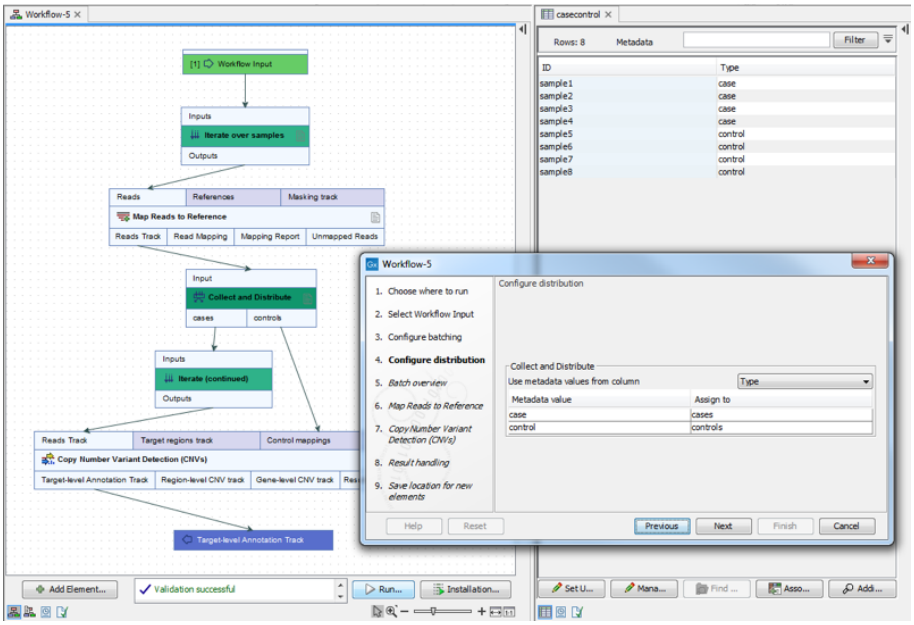


Figure 11.54: *It is possible to select the metadata column "Type" to split the samples into cases and controls, even though the iteration was over the "ID" column in the same metadata table.*

Control flow elements are described in more detail in section 11.5.

11.6.3 Matching up inputs with each other and analyzing them together later in the workflow

In some applications, we want to analyze pairs, triplets or even larger groups of data, such that they are matched up with each other in each iteration. This may be achieved without control flow elements if the results from each batch do not need to be merged later, and only one level of batching is needed, as described in section 11.4.4. If there is a need to merge the results from different batches, or multiple levels of batching, then this can be achieved using a configured Iterate element.

To configure an Iterate element, double-click on it in the workflow, and enter the number of inputs and outputs, separated by commas (figure 11.55). The effect of this will be that the Iterate element will take as many inputs as specified, and will group them into batch units while matching them up with each other.

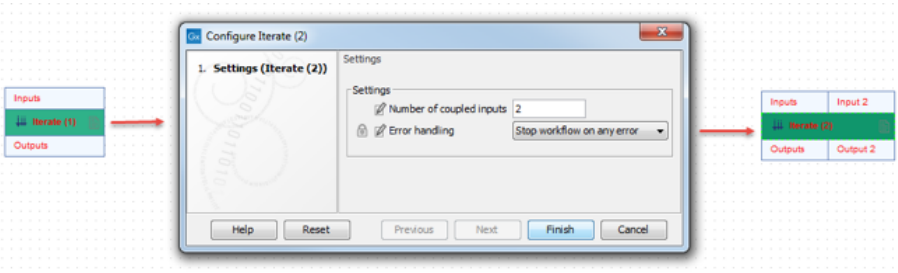


Figure 11.55: *Double-click the Iterate element to configure its inputs and outputs.*

Important note: the Collect and Distribute element always ends the ongoing iteration. To continue an iteration on a path after the data have been split, place a new Iterate element directly after the Collect and Distribute element.

In the workflow in figure 11.56, the reads and contigs will be matched for the Map Reads to Contigs tool, in the same way as described in section 11.4.4. However, the configured Iterate element makes it possible to collect the unmapped reads into a single list for the entire batch.

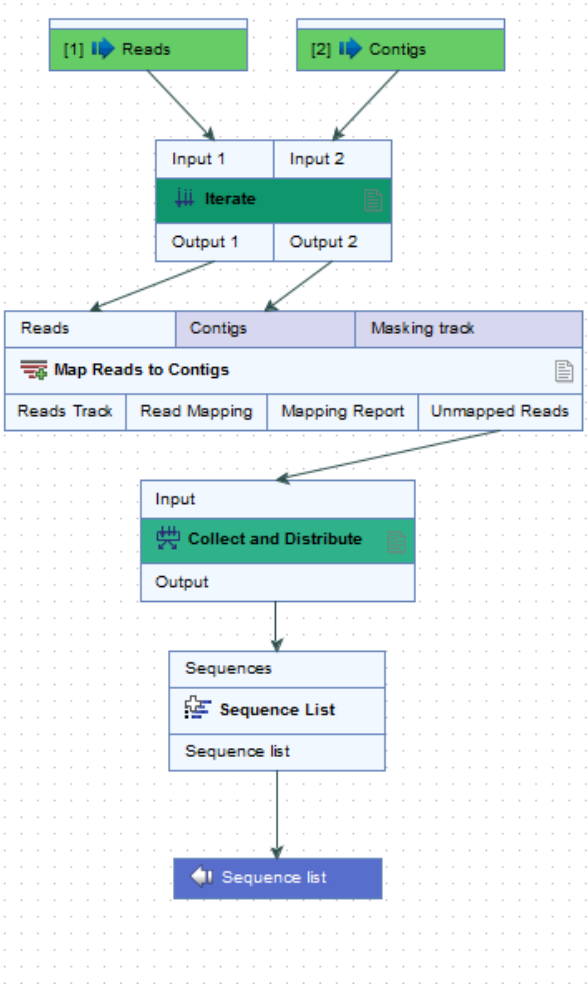


Figure 11.56: *In this partial workflow, the Collect and Distribute element is being used so that a single sequence list is created that will contain all the unmapped reads from all the initial inputs.*

Control flow elements are described in more detail in section 11.5.

11.7 Managing workflows

Workflows can be managed from the Workflow Manager:

Help | Manage Workflows ( )

or using the "Workflows" button ( ) in the toolbar and then select "Manage Workflow..." ( ).

The Workflow Manager (figure 11.57) lists Installed workflows and Ready-to-Use workflows, but the functionalities described below (Configure, Rename, and Uninstall) are only available to custom workflows. You can always create a copy of a Ready-to-Use workflow (by opening the Ready-to-Use workflow and saving a copy in your Navigation Area) to enable the options described below.

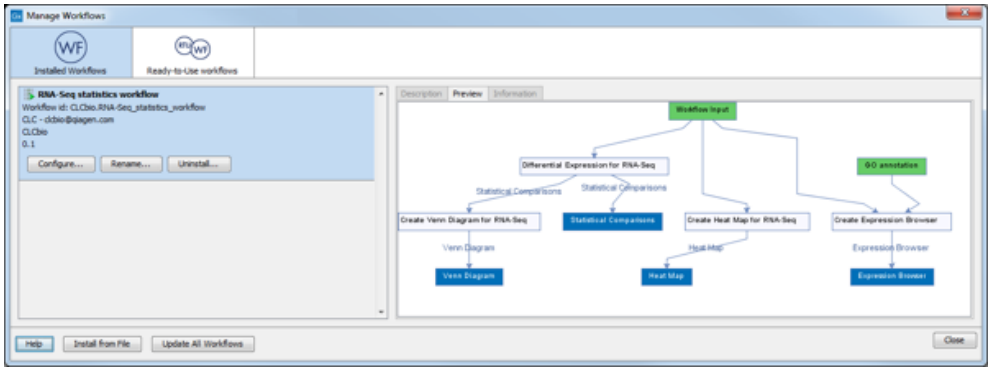


Figure 11.57: *Preview of the workflow.*

Configure Select the workflow of interest and click on the button labeled Configure. You will be presented with a dialog listing all the reference data that need to be selected. An example is shown in figure 11.58.

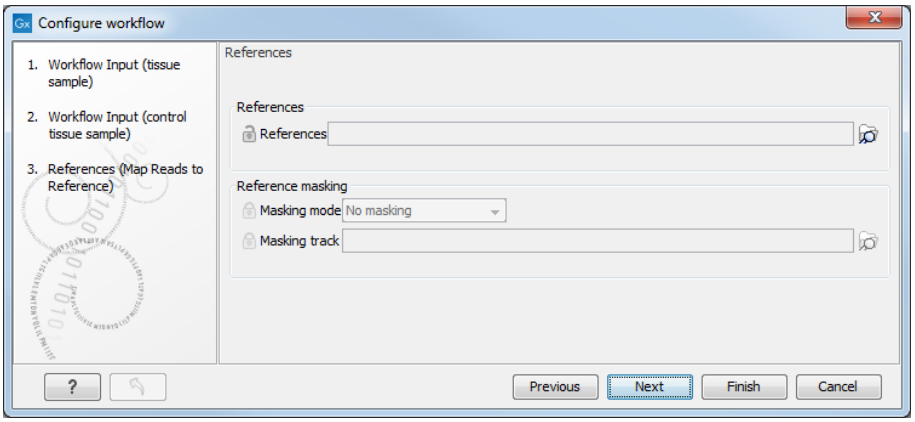


Figure 11.58: *Configuring parameters for the workflow.*

This dialog also allows you to lock parameters of the workflow (see more about locking in section 11.3.3). Note that data parameters should only be locked if they should not be set, or 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. In addition, if the workflow is intended to be executed on a server, it is important to select reference data that is located on the server.

Rename In addition to the configuration option, it is also possible to rename the workflow. This will change the name of the workflow in the **Toolbox**. The workflow id (see below) remains the same. To rename an element right click on the element name in the Navigation Area and select "Rename" or click on the F2 button.

Uninstall Use this button to install a workflow.

Description, Preview and Information In the right side of the window, you will find three tabs. **Description** contains the description that was entered when creating the workflow installer (see section 11.7.2), the **Preview** shows a graphical representation of the workflow , and finally you can get **Information** about the workflow (figure 11.59).

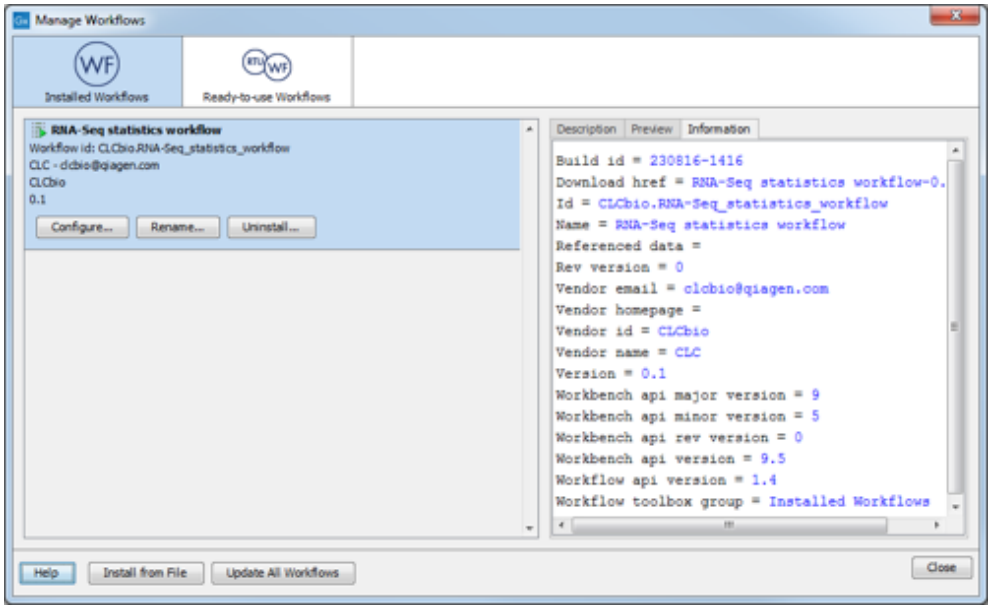


Figure 11.59: *Workflow identification and versioning.*

The Information field contains the following:

- Build id. The date (day month year) followed by the time (based on a 24 hour time) when the workflow was exported to a file through the Installation button at the bottom of the workflow window. If the workflow was installed locally without going through a file, the build ID will reflect the time of installation.
- Download href. The name of the workflow .cpw file
- Id. The unique id of a workflow, by which the workflow is identified
- Major version. The major version of the workflow
- Minor version. The minor version of the workflow
- Name. Name of workflow
- Rev version. Revision version. The functionality is activated but currently not in use
- Vendor id. ID of vendor that has created the workflow
- Version. <Major version>.<Minor version>
- Workbench api version. Workbench version
- Workflow api version. Workflow system version (a technical number that can be used for troubleshooting)

11.7.1 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 Ready-to-Use 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 10.0.1 and 10.5 are part of the same major release line (10). Version 9.x is part of a different major version line (9).

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 11.60).

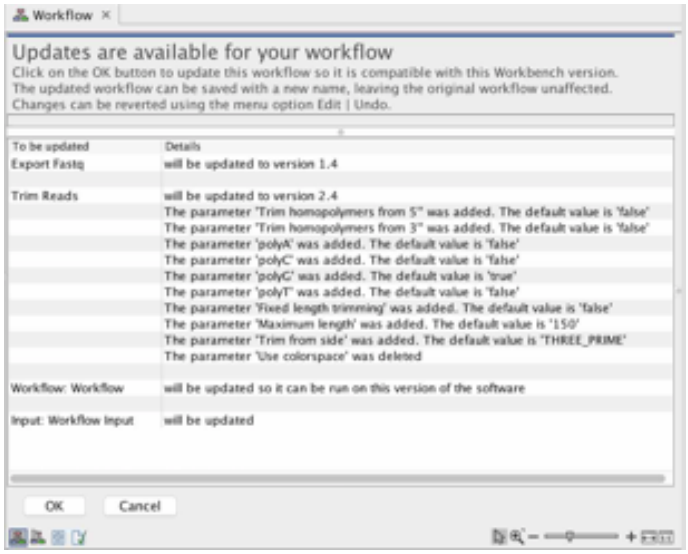


Figure 11.60: *The workflow update editor lists tools and parameters that will be updated.*

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 Ready-to-Use workflows when using a upgraded Workbench in the *same* major version line

When working on an upgraded *CLC Workbench* in the same major version line, installed and Ready-to-Use workflows are updated using the Workflow Manager.

To start the Workflow Manager, go to:

Help | **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 11.61).

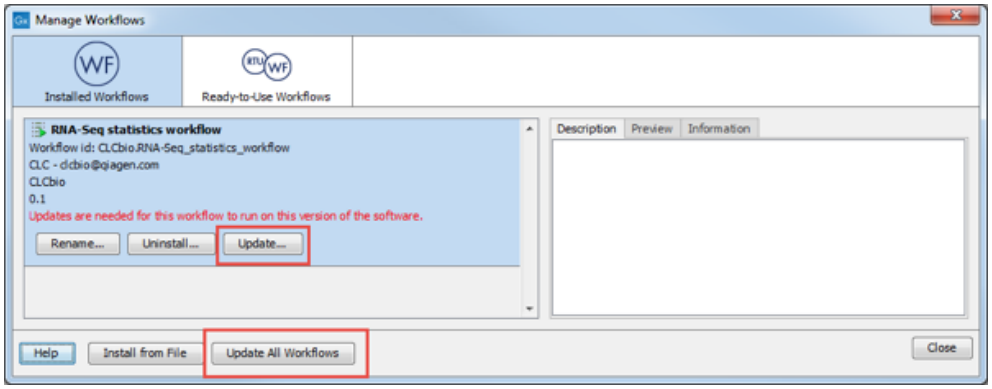


Figure 11.61: *A message in red text indicates a workflow needs to be updated. The Update button can be used to update an individual workflow. Alternatively, update all workflows that need updating by clicking on the Update All Workflows button.*

When you update a workflow through the Workflow Manager, the old version is overwritten.

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 Ready-to-Use workflows, distributed via plugins, the *CLC Workbench* will usually need to be run as an administrative user.

When one or more installed workflows or Ready-to-Use 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 11.62).

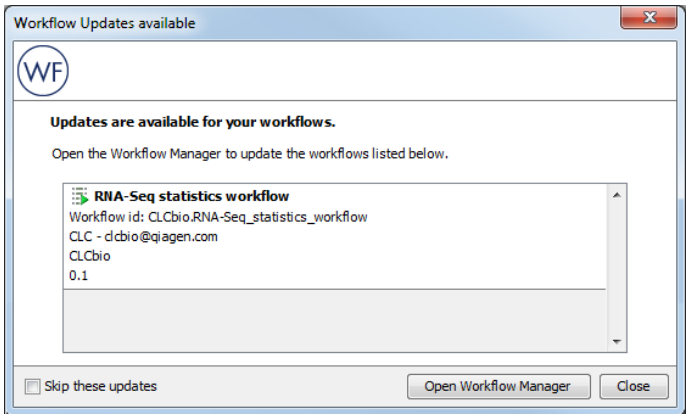


Figure 11.62: *A dialog reporting that an installed workflow needs to be updated to be used on this version of the Workbench.*

Updating installed workflows when using software in a *higher* major version line

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.

This is done by selecting the workflow in the **Installed Workflows** folder of the **Toolbox** in the bottom left side of the Workbench, then right-clicking on the workflow name and choosing the option "Open Copy of Workflow" (figure 11.63).

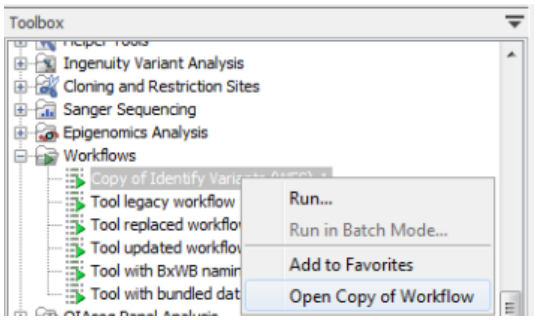


Figure 11.63: *Open a copy of an installed workflow by right-clicking on its name in the Workbench Toolbox.*

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 worfklow, 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.

11.7.2 Creating a workflow installation file

At the bottom of the workflow editor, click the **Create Installer** button (or use the shortcut Shift + Alt + I) to bring up a dialog where you provide information about the workflow to be distributed (see an example in figure 11.64).

The information entered in this dialog will be visible for users installing the workflow and will enable them to look up the source of the workflow any time.

Author name Provide the name of the author of the workflow.

Author email Provide the email of the author of the workflow.

Author homepage Provide the homepage of the author of the workflow.

Organization The organization name is important because it is part of the workflow id (see more in section 11.7.1).

Workflow name The workflow name is based on the name used when saving the workflow in the **Navigation Area**. The workflow name is essential because it is used as part of the workflow id (see more in section 11.7.1). The workflow name can be changed during the installation of the workflow. This is useful whenever you have a workflow that you would like to use e.g.

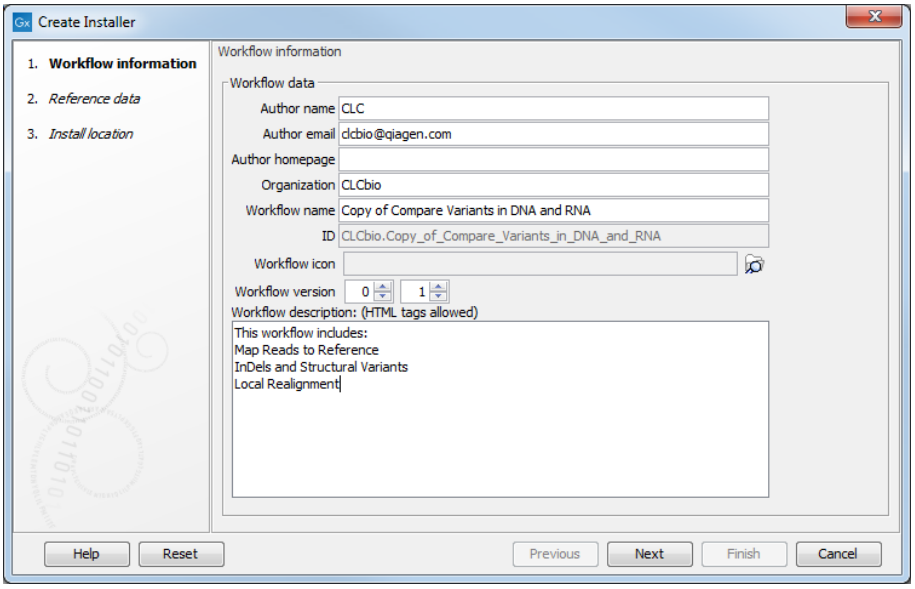


Figure 11.64: *Workflow information for the installer.*

with small variations. The original workflow name will remain the same in the **Navigation Area** - only the installed workflow will receive the customized name.

ID The final id of the workflow.

Workflow icon An icon can be provided. This will show up in the installation overview and in the **Toolbox** once the workflow is installed. The icon should be a 16 x 16 pixels gif or png file. If the icon is larger, it will automatically be resized to fit 16 x 16 pixels.

Workflow version A major and minor version can be provided.

Workflow description Provide a textual description of the workflow. This information will be displayed when a user mouses-over the name of the installed Workflow in the Workbench Toolbox, and is also presented in the Description tab for that Workflow in the Manage Workflows tool, described in section 11.7. Simple HTML tags are allowed (should be HTML 3.1 compatible, see `http://www.w3.org/TR/REC-html32`).

If you configured any of the workflow elements with data, clicking **Next** will give you the following options for the reference data (see figure 11.65). You can choose to

- **Ignore**. This is the recommended setting when the workflow inputs have been configured with workflow roles. The user installing the workflow on their local system will have to apply their own references.
- **Reference**. This is the recommended setting when the workflow inputs have been configured by selecting elements in a shared CLC_References directory. The data will not be bundled with the workflow, but the reference data is included in the workflow by pointing to the shared data in the CLC_References directory. This is particularly useful when working with large reference data.
- **Bundle**. This is the recommended setting when you cannot, or do not wish, to share the data through a CLC_References folder (see section 8.4 for how to transfer data to a

CLC_References folder.) This option will include the data in the workflow by directly bundling the reference data with the workflow. **Note!** Bundling data should only be used to bundle small data sets with the workflow installer.

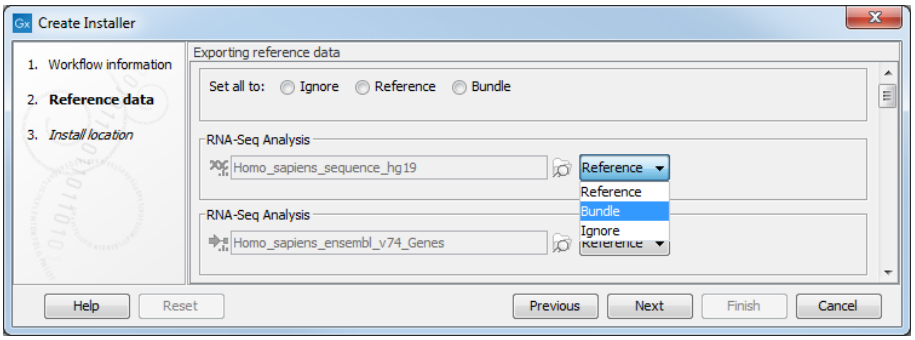


Figure 11.65: *Bundling data with the workflow installer.*

Click **Next** and you will be asked to specify where to install the workflow (figure 11.66). You can install your workflow directly on your local computer. If you are logged on a server and are the administrator, the option "Install the workflow on the current server" will be enabled. Finally, you can select to save the workflow as a .cpw file that can be installed on another computer. Click **Finish**. This will install the workflow directly on the selected destination. If you have selected to save the workflow for installation on another computer, you will be asked where to save the file after clicking **Finish**. If you chose to bundle data with your workflow installation, you will be asked for a location to put the bundled data on the workbench.

Installing a workflow with bundled data on a server, the data will be put in a folder created in the first writable persistence location. Should this location not suit your needs, you can always move it afterwards, using the normal persistence operations.

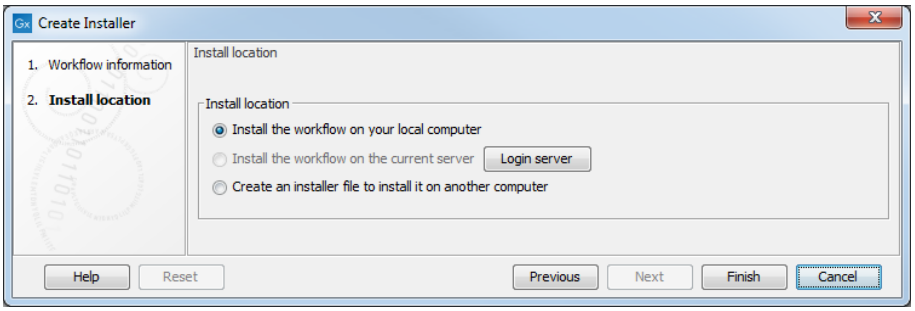


Figure 11.66: *Select whether the workflow should be installed on your local computer or on the current server. A third option is to create an installer file (.cpw) that can be installed on another computer.*

In cases where an existing workflow that has already been installed is modified, the workflow must be reinstalled. This can be done by first saving the workflow after it has been modified and then pressing the **Create Installer** button. Click through the wizard and select whether you wish to install the modified workflow on your local computer or on a server. Press **Finish**. This will open a pop-up dialog "Workflow is already installed" (figure 11.67) with the option that you can force the installation. This will uninstall the existing workflow and install the modified version of the workflow. **Note!** When forcing installation of the modified workflow, the configuration of the original workflow will be lost.

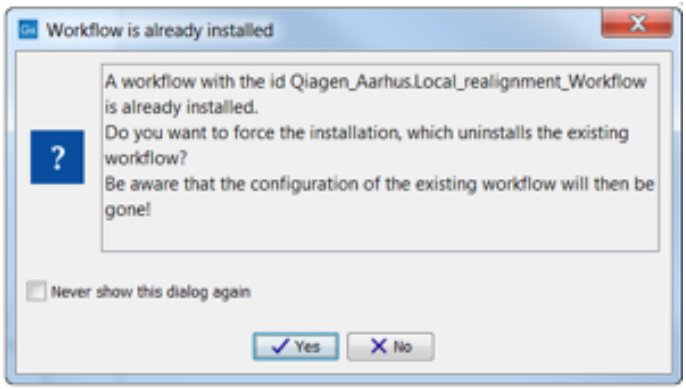


Figure 11.67: *Select whether you wish to force the installation of the workflow or keep the original workflow.*

11.7.3 Installing a workflow

Workflow .cpw files can be installed on a Workbench using the workflow manager:

Help | Manage Workflows ( )

or press the "Workflows" button ( ) in the toolbar and then select "Manage Workflow..." ( ).

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 11.68).

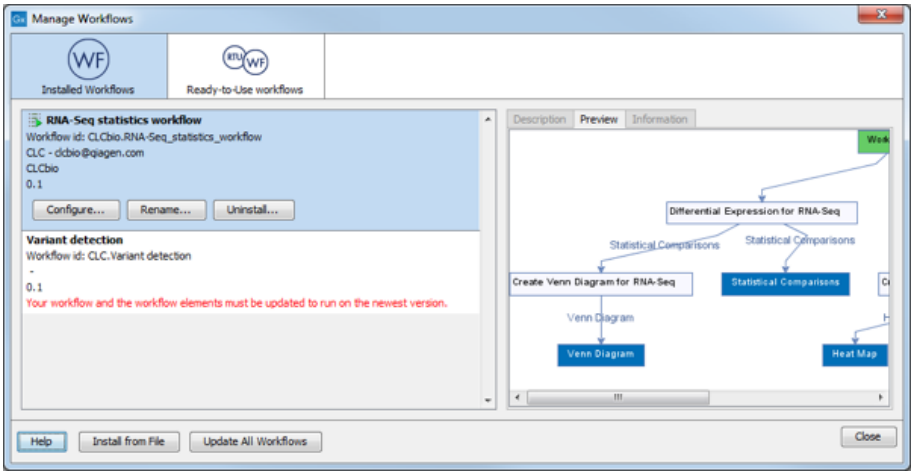


Figure 11.68: *Workflows available in the workflow manager. The alert on the "Variant detection" workflow means that this workflow needs to be updated.*

When installing the workflow on a different system to the one where it was created, the connection to the reference data (not the bundle data) needs to be re-established. This is only possible when the parameter is unlocked (which it usually is by default).

Part III

Basic sequence analysis

Chapter 12

Viewing and editing sequences

Contents

12.1 View sequence . . . 265
12.1.1 Sequence settings in Side Panel . . . 266
12.1.2 Selecting parts of the sequence . . . 272
12.1.3 Editing the sequence . . . 273
12.1.4 Sequence region types . . . 274
12.2 Circular DNA . . . 275
12.2.1 Using split views to see details of the circular molecule . . . 276
12.2.2 Mark molecule as circular and specify starting point . . . 277
12.3 Working with annotations . . . 278
12.3.1 Viewing annotations . . . 279
12.3.2 Adding annotations . . . 283
12.3.3 Edit annotations . . . 285
12.3.4 Removing annotations . . . 286
12.4 Element information . . . 287
12.5 View as text . . . 288
12.6 Sequence Lists . . . 288

CLC Genomics Workbench offers five different ways of viewing and editing single sequences as described in the first five sections of this chapter. Furthermore, this chapter also explains how to create a new sequence and how to gather several sequences in a sequence list.

12.1 View sequence

When you double-click a sequence in the **Navigation Area**, the sequence will open automatically, and you will see the nucleotides or amino acids. The zoom options described in section 2.2 allow you to e.g. zoom out in order to see more of the sequence in one view. There are a number of options for viewing and editing the sequence which are all described in this section.

All the options described in this section also apply to alignments (further described in section 21.2).

12.1.1 Sequence settings in Side Panel

Each view of a sequence has a **Side Panel** located at the right side of the view (see figure 12.1.

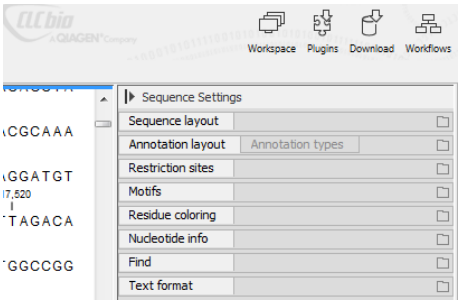


Figure 12.1: *Overview of the Side Panel which is always shown to the right of a view.*

When you make changes in the **Side Panel** the view of the sequence is instantly updated. To show or hide the **Side Panel**:

select the View | Ctrl + U

or **Click the ( ) at the top right corner of the Side Panel to hide | Click the ( ) to the right to show**

Below, each group of settings will be explained. Some of the preferences are not the same for nucleotide and protein sequences, but the differences will be explained for each group of settings.

Note! When you make changes to the settings in the **Side Panel**, they are not automatically saved when you save the sequence. Click **Save/restore Settings** ( ) to save the settings (see section 4.6 for more information).

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 12.3.1.

Restriction sites

See section 12.1.1.

Motifs

See section 15.8.1.

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 only apply to nucleotide sequences.

- **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 12.1.1.

    - **+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 16.4.
  - **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.

- **Trace data.** See section 19.1.

- **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 score is displayed on a graph (Learn how to export the data behind the graph in section 6.8).
    - **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.

- **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 level is displayed on a graph (Learn how to export the data behind the graph in section 6.8).
    * **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.

- **Secondary structure.** Allows you to choose how to display a symbolic representation of the secondary structure along the sequence.

  See section 23.2.3 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 17.3.1.

- **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.** Searches 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 enabled using the "Nucleotide info" side panel.

- **Position search.** Finds 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 an 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.** Searches 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.** Five different sizes.
- **Font.** Shows a list of Fonts available on your computer.
- **Bold residues.** Makes the residues bold.

Restriction sites in the Side Panel

Please see section 20.1.1.

12.1.2 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 several parts at the same time (multiselect) You can select several 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 12.1.1.)

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 | **Toolbox** | **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.

12.1.3 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 20.3.4.

Note When editing annotated nucleotide sequences, the annotation content is not updated automatically (but its position is). Please refer to section 12.3.3 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.

12.1.4 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 12.2 is an example of three regions with separate colors.

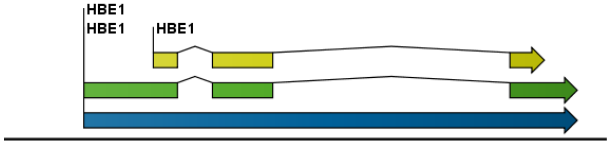


Figure 12.2: *Three regions on a human beta globin DNA sequence (HUMHBB).*

Figure 12.3 shows an artificial sequence with all the different kinds of regions.

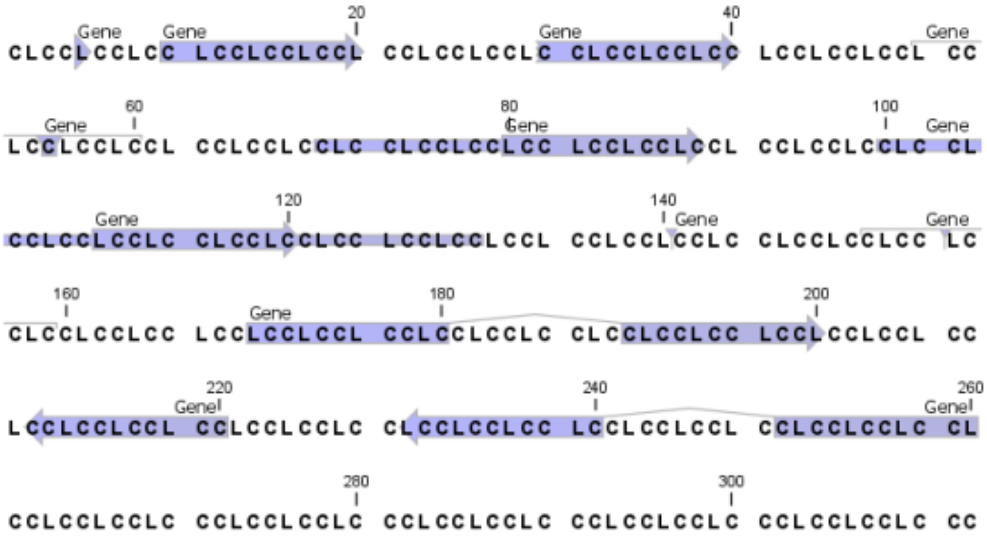


Figure 12.3: *Region #1: A single residue, Region #2: A range of residues including both endpoints, Region #3: A range of residues starting somewhere before 30 and continuing up to and including 40, Region #4: A single residue somewhere between 50 and 60 inclusive, Region #5: A range of residues beginning somewhere between 70 and 80 inclusive and ending at 90 inclusive, Region #6: A range of residues beginning somewhere between 100 and 110 inclusive and ending somewhere between 120 and 130 inclusive, Region #7: A site between residues 140 and 141, Region #8: A site between two residues somewhere between 150 and 160 inclusive, Region #9: A region that covers ranges from 170 to 180 inclusive and 190 to 200 inclusive, Region #10: A region on negative strand that covers ranges from 210 to 220 inclusive, Region #11: A region on negative strand that covers ranges from 230 to 240 inclusive and 250 to 260 inclusive.*

12.2 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 12.4.

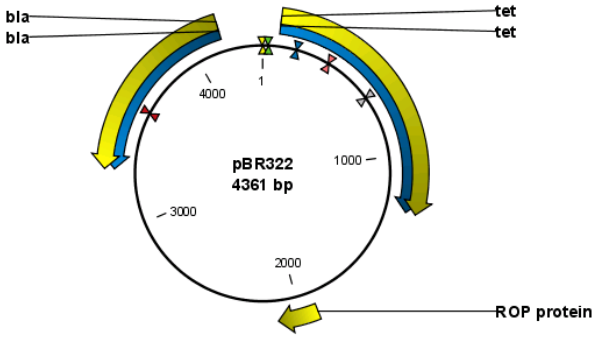


Figure 12.4: *A molecule shown in a circular view.*

This view of the sequence shares some of the properties of the linear view of sequences as

described in section 12, but there are some differences. The similarities and differences are listed below:

- **Similarities**:
  - The editing options.
  - Options for adding, editing and removing annotations.
  - **Restriction Sites**, **Annotation Types**, **Find** and **Text Format** preferences groups.
- **Differences**:
  - 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.

12.2.1 Using split views to see details of the circular molecule

In order 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 (ACT) 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 12.5.

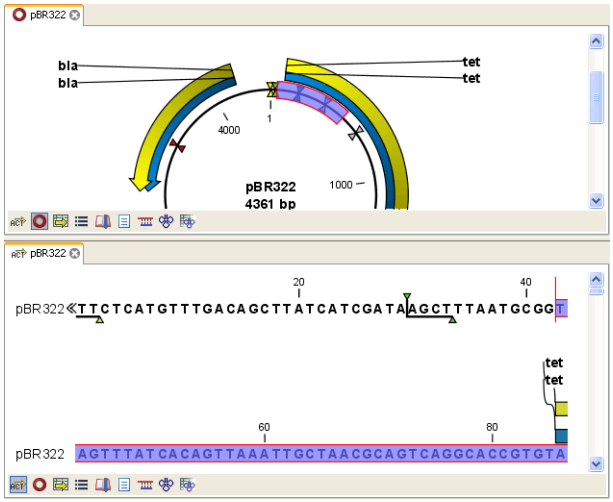


Figure 12.5: *Two views showing the same sequence. The bottom view is zoomed in.*

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.

12.2.2 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 12.6).

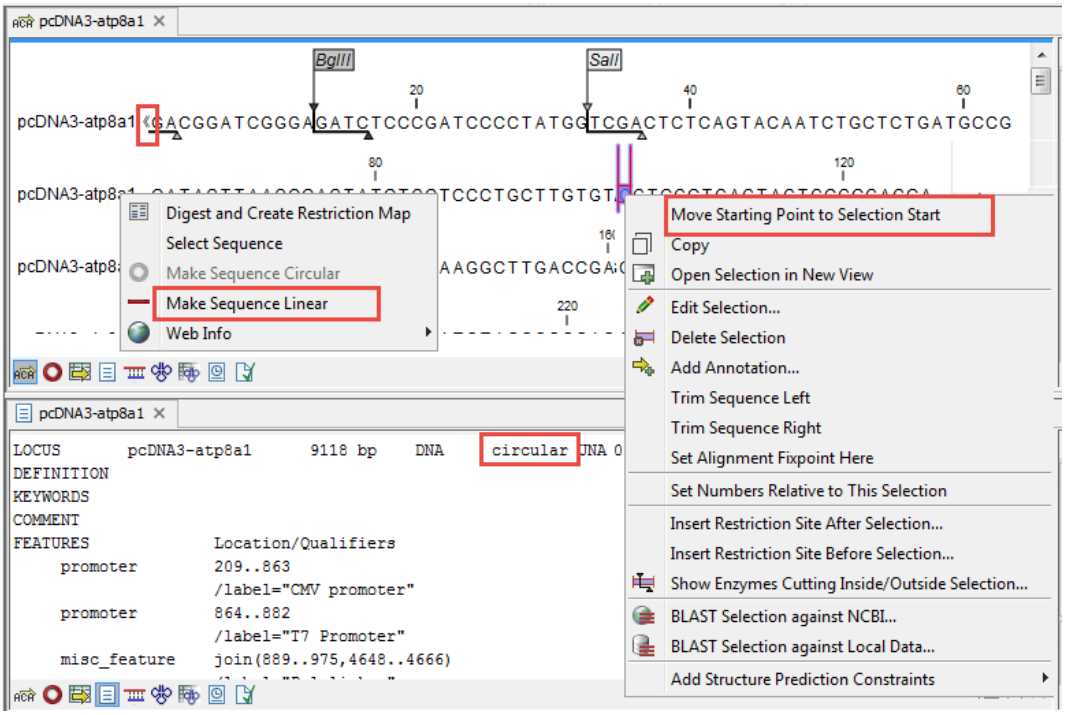


Figure 12.6: *Double angle brackets marks the start and end of a circular sequence seen in linear view. Below, the Text view of the same sequence shows the mention circular in the first line.*

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 Selection Start** (figure 12.7).

.

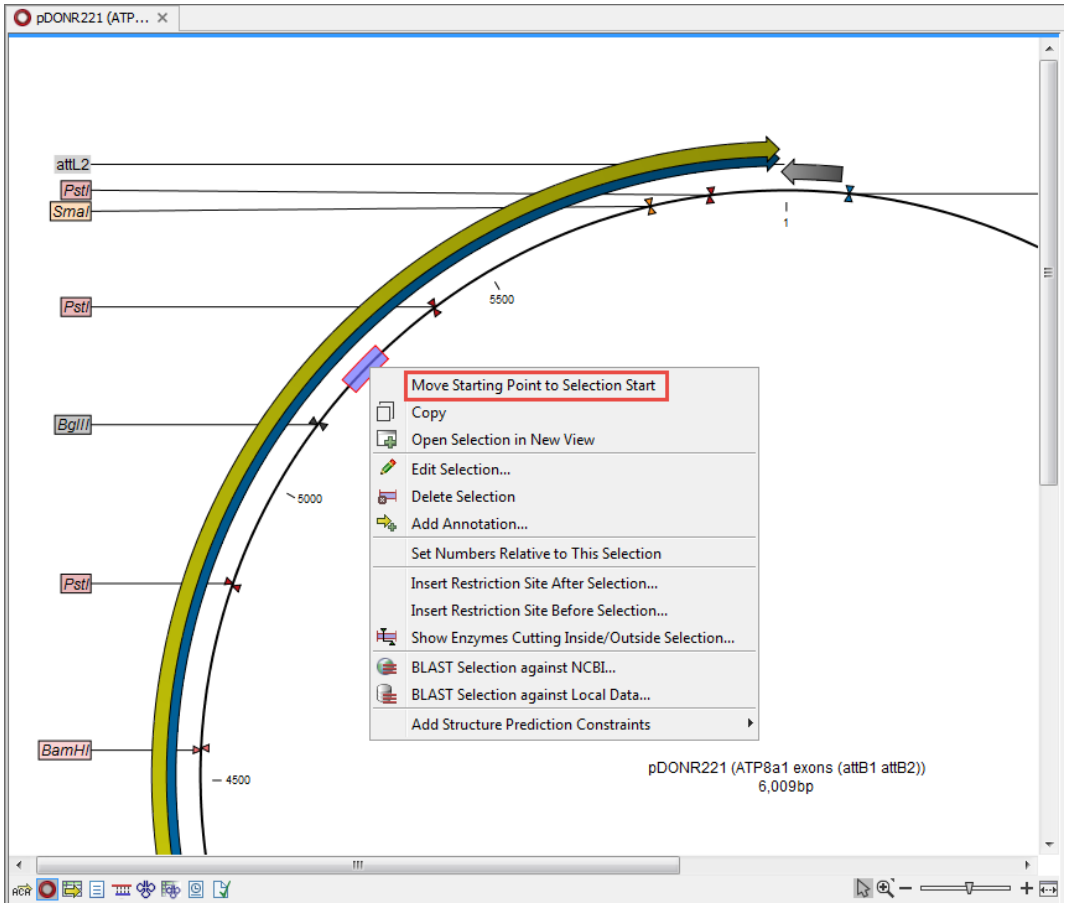


Figure 12.7: *Right-click on a circular sequence to move the starting point to the selected position.*

12.3 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 14.4.2), and atom groups defined on the structure transferred to sequence annotations or vica versa (section 14.4.3).
- You can manually add annotations to a sequence (described in the section 12.3.2).

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 34.2.

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.

12.3.1 Viewing annotations

Annotations can be viewed in a number of different ways:

- As arrows or boxes in all views displaying sequences (sequence lists, alignments etc)
- In the table of annotations ( ).
- In the text view of sequences ( )

In the following sections, these view options will be described in more detail.

In all the views except the text view ( ), annotations can be added, modified and deleted. This is described in the following sections.

View Annotations in sequence views

Figure 12.8 shows an annotation displayed on a sequence.

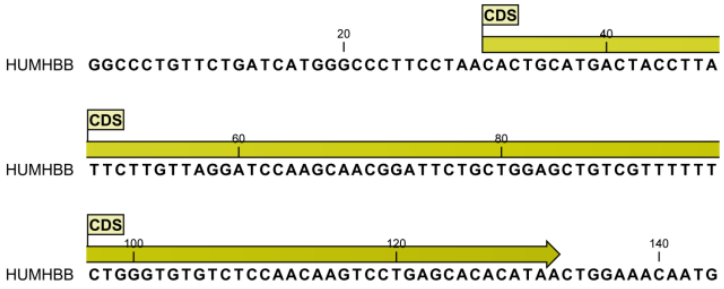


Figure 12.8: *An annotation showing a coding region on a genomic dna sequence.*

The various sequence views listed in section 12.3.1 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 12.9.

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).

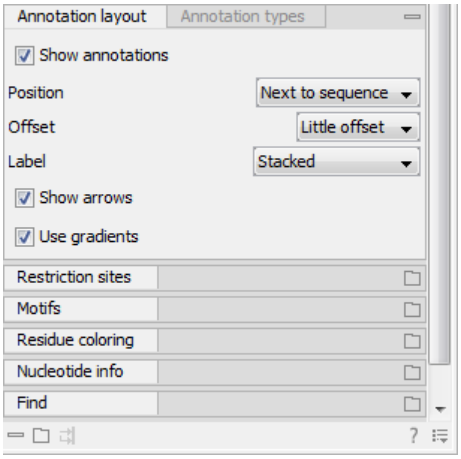


Figure 12.9: *The annotation layout in the Side Panel. The annotation types can be shown by clicking on the "Annotation types" tab.*

- **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 them 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 12.10).

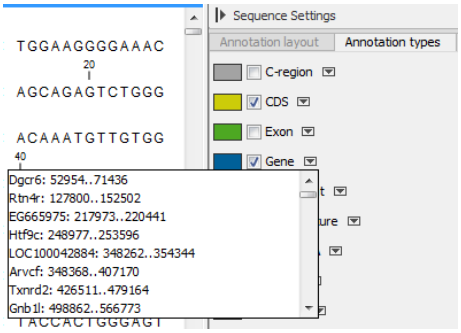


Figure 12.10: *Browsing the gene annotations on a sequence.*

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 12.11) 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.

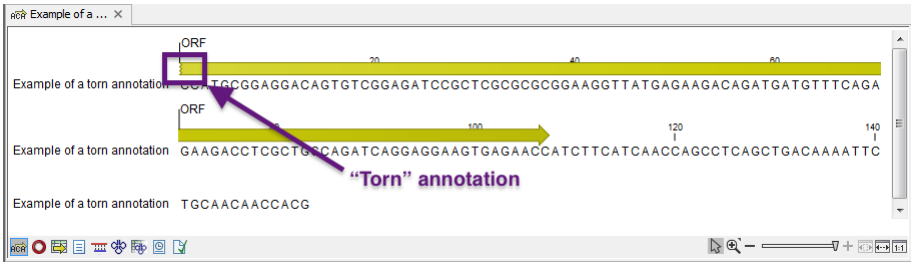


Figure 12.11: *Example of a torn annotation on a sequence.*

View Annotations in a table

Annotations can also be viewed in a table:

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 12.12).

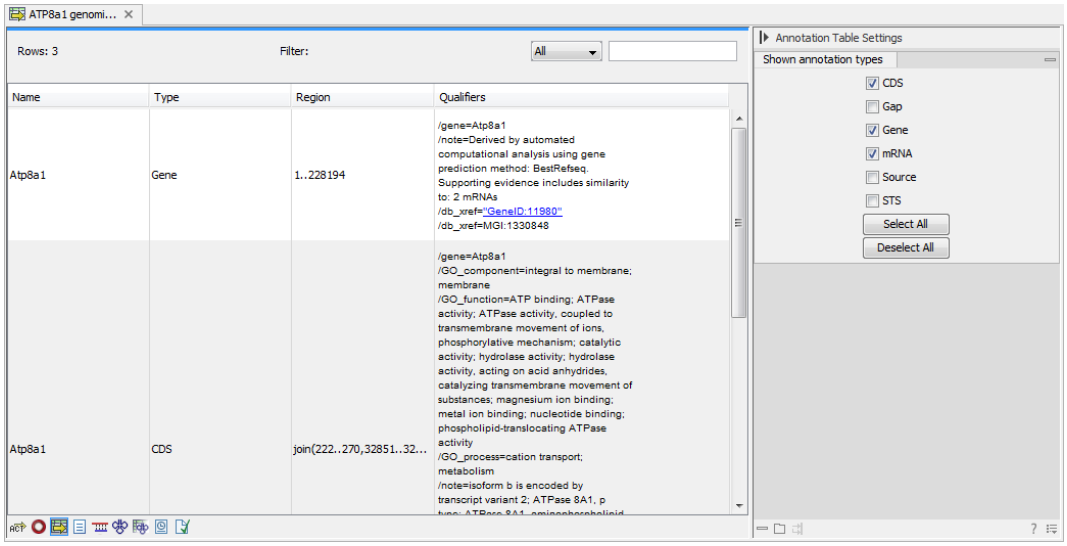


Figure 12.12: *A table showing annotations on the sequence.*

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.**

The Name, Type and Region for each annotation can be edited simply by double-clicking, typing the change directly, and pressing **Enter**.

This information corresponds to the information in the dialog when you edit and add annotations (see section 12.3.2).

You can benefit from this table in several ways:

- It provides an intelligible overview of all the annotations on the sequence.
- You can use the filter at the top to search the annotations. Type e.g. "UCP" into the filter and you will find all annotations which have "UCP" in either the name, the type, the region or the qualifiers. Combined with showing or hiding the annotation types in the **Side Panel**, this makes it easy to find annotations or a subset of annotations.
- You can copy and paste annotations, e.g. from one sequence to another.
- If you wish to edit many annotations consecutively, the double-click editing makes this very fast (see section 12.3.2).

12.3.2 Adding annotations

Adding annotations to a sequence can be done in two ways:

Open the sequence in a sequence view (double-click in the Navigation Area) | make a selection covering the part of the sequence you want to annotate[1] **| right-click the selection | Add Annotation ( )**

or **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 ( ) | right click anywhere in the annotation table | select Add Annotation ( )**

This will display a dialog like the one in figure 12.13.

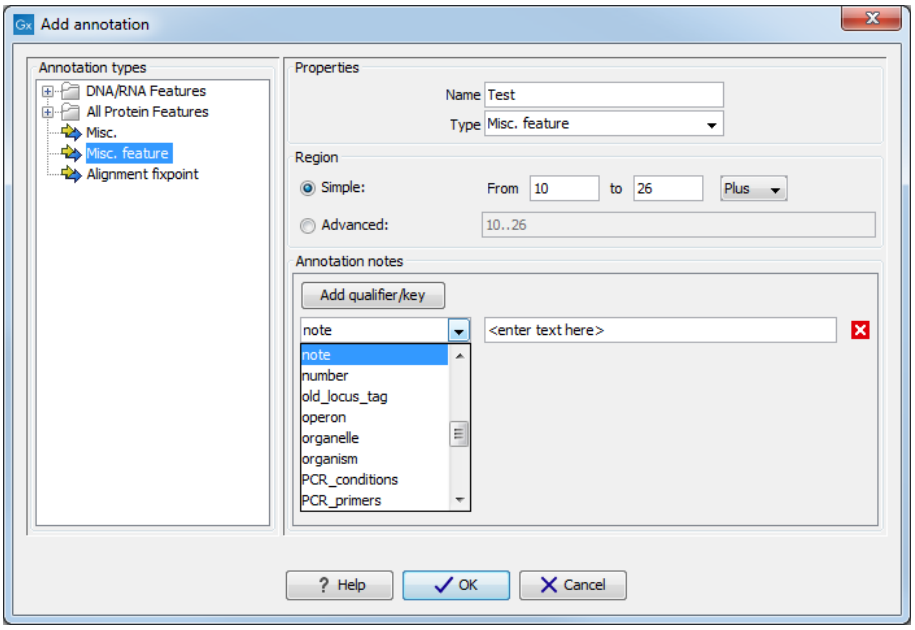


Figure 12.13: *The Add Annotation dialog.*

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 [2].

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 12.3.1).
- **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

[1] (See section 2.2.3 on how to make selections that are not contiguous.)

[2] Note 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, you own annotation type will be preserved

and GenBank. The following are examples of how to use the syntax (based on http://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 12.3.1) 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.

12.3.3 Edit annotations

To edit an existing annotation from within a sequence view:

> **right-click the annotation | Edit Annotation ( )**

This will show the same dialog as in figure 12.13, with the exception that some of the fields are filled out depending on how much information the annotation contains.

There is another way of quickly editing annotations which is particularly useful when you wish to edit several annotations.

To edit the information, simply double-click and you will be able to edit e.g. the name or the annotation type. If you wish to edit the qualifiers and double-click in this column, you will see the dialog for editing annotations.

Advanced editing of annotations

Sometimes you end up with annotations which do not have a meaningful name. In that case there is an advanced batch rename functionality:

> **Open the Annotation Table ( ) | select the annotations that you want to rename | right-click the selection | Advanced Rename**

This will bring up the dialog shown in figure 12.14.

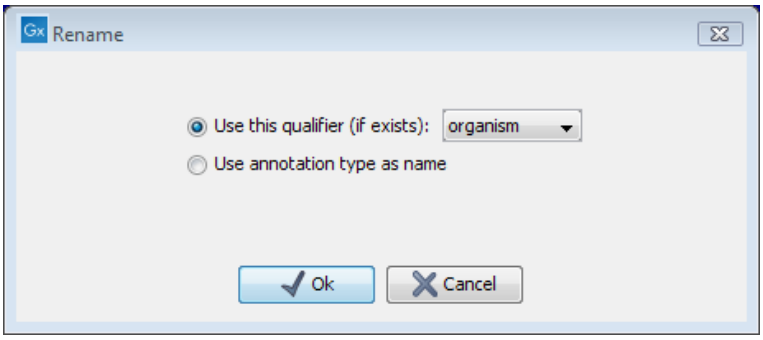


Figure 12.14: *The Advanced Rename dialog.*

In this dialog, you have two options:

- **Use this qualifier.** Use one of the qualifiers as name. A list of all qualifiers of all the selected annotations is shown. Note that if one of the annotations do not have the qualifier you have chosen, it will not be renamed. If an annotation has multiple qualifiers of the same type, the first is used for naming.
- **Use annotation type as name.** The annotation's type will be used as name (e.g. if you have an annotation of type "Promoter", it will get "Promoter" as its name by using this option).

A similar functionality for batch re-typing annotations is available in the right-click menu as well, in case your annotations are not typed correctly:

> **Open the Annotation Table ( ) | select the annotations that you want to retype | right-click the selection | Advanced Retype**

This will bring up the dialog shown in figure 12.15.

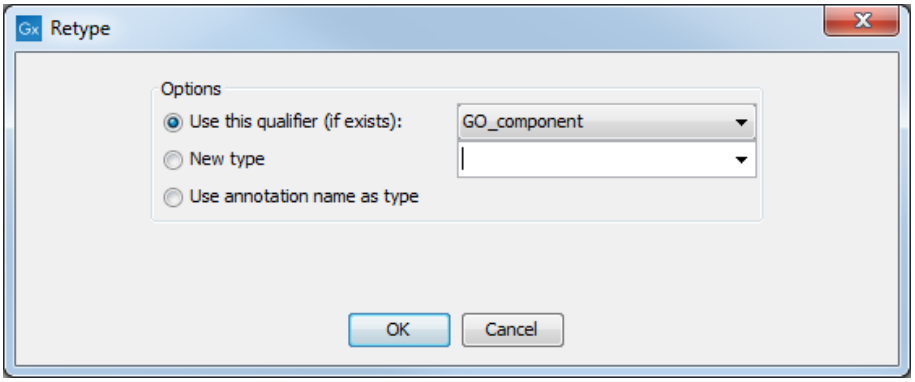


Figure 12.15: *The Advanced Retype dialog.*

In this dialog, you have two options:

- **Use this qualifier.** Use one of the qualifiers as type. A list of all qualifiers of all the selected annotations is shown. Note that if one of the annotations do not have the qualifier you have chosen, it will not be retyped. If an annotation has multiple qualifiers of the same type, the first is used for the new type.
- **New type**. You can select from a list of all the pre-defined types as well as enter your own annotation type. All the selected annotations will then get this type.
- **Use annotation name as type.** The annotation's name will be used as type (e.g. if you have an annotation named "Promoter", it will get "Promoter" as its type by using this option).

12.3.4 Removing annotations

Annotations can be hidden using the **Annotation Types** preferences in the **Side Panel** to the right of the view (see section 12.3.1). 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

12.4 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 12.16.



Figure 12.16: *The initial display of sequence info for the HUMHBB DNA sequence from the Example data.*

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.
- **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 section 2.1.1 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.

12.5 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 12.4.)

In the **Side Panel**, you find a search field for searching the text in the view.

12.6 Sequence Lists

The **Sequence List** is a file containing a number of sequences. Having sequences in a sequence list can help organizing sequence data. A Sequence List can be displayed in a graphical sequence view or in a tabular format. The two different views of the same sequence list are shown in split screen in figure 12.17.

The **graphical view of sequence lists** is almost identical to the view of single sequences (see section 12). The main difference is that you now can see more than one sequence in the same view, and additionally have a few extra options for sorting, deleting and adding sequences:

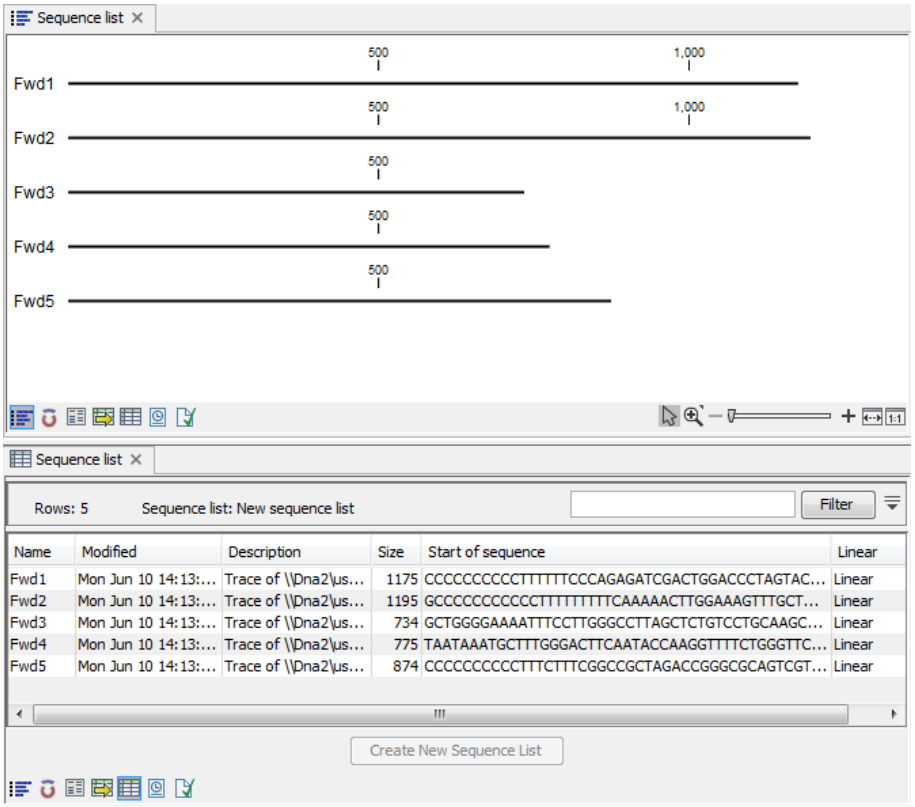


Figure 12.17: *A sequence list containing multiple sequences can be viewed in either a table or in a graphical sequence list. The graphical view is useful for viewing annotations and the sequence itself, while the table view provides other information like sequence lengths, and the number of sequences in the list (number of Rows reported).*

- To add extra sequences to the list, right-click an empty (white) space in the view, and select **Add Sequences**.
- To delete a sequence from the list, right-click the sequence's name and select **Delete Sequence**.
- To sort the sequences in the list, right-click the name of one of the sequences and select **Sort Sequence List by Name** or **Sort Sequence List by Length**.
- To rename a sequence, right-click the name of the sequence and select **Rename Sequence**.

Each sequence in the **table sequence list** is displayed with:

- Name
- Accession
- Description
- Modification date
- Length
- First 50 residues

The number of sequences in the list is reported as the number of Rows at the top of the table view. Adding and removing sequences from the list is easy: adding is done by dragging the sequence from another list or from the Navigation Area and drop it in the table. To delete sequences, simply select them and press **Delete** ( ). To extract a sequence from a sequence list, drag the sequence directly from the table into the Navigation Area. Another option is to extract all sequences found in the list using the **Extract Sequences** tool. A description of how to use the **Extract Sequences** tool can be found in section 15.1.

Sequence lists are generated automatically when you import files containing more than one sequence. They may also be created as the output from particular Workbench tool, including database searches. For more information about creating sequence lists from a database search, see (chapter 7.1).

You can create a subset of a Sequence List: select the relevant sequences, right-click on the selected elements and choose **Create New Sequence List** from the drop down menu. This will generate a new sequence list that only includes the selected sequences.

A **Sequence List** can also be created from single sequences or by merging already existing sequence lists with the Workbench. To do this, select two or more sequences or sequence lists in the Navigation Area, right click on the selected elements and choose

New | Sequence List ( )

Alternatively, you can launch this tool via the "File" menu system.

This opens the **Sequence List** Wizard (figure 12.18). The dialog allows you to select more sequences to include in the list, or to remove already chosen sequences from the list.

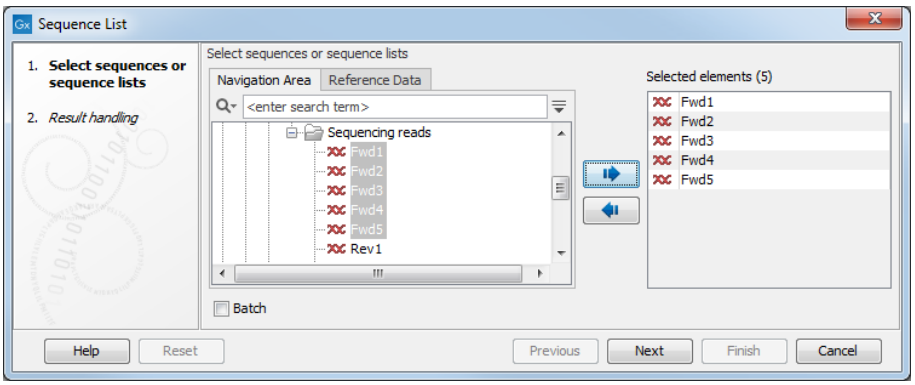


Figure 12.18: *A Sequence List dialog.*

If you are trying to create a new sequence list from a mixture of paired and unpaired datasets, a warning message will let you know that the resulting sequence list will be set as unpaired (figure 12.19).

This warning also appears when trying to create a Sequence List out of paired reads lists for which the Minimum and Maximum distances are different between lists. If that is the case, distances can be edited to be similar for all lists that needs to be merged in a new one.

For this, open all Sequence Lists one after the other and click on the Show Element Info icon at the bottom of the view (figure 12.20). Edit the distances by clicking on the button "Edit" next to the entry "Paired status" and click OK. Save the Sequence lists with the edited Paired statuses before attempting to create a merged sequence List. This final list's status will be set as Paired reads.

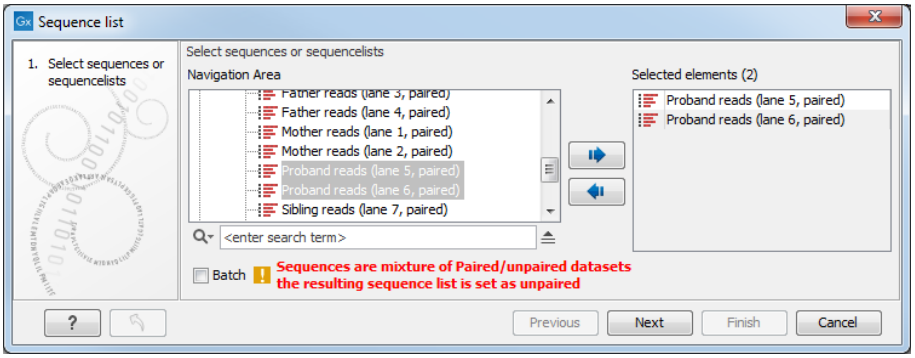


Figure 12.19: *A warning appears when trying to create a new sequence list from a mixture of paired and unpaired datasets.*

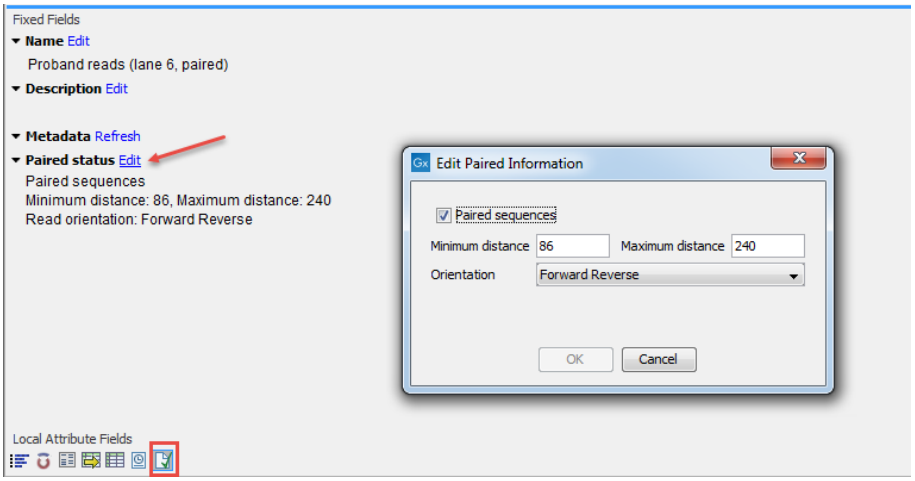


Figure 12.20: *Edit the Minimum and Maximum distances of several sequence lists to be able to merge them into one.*

Chapter 13

BLAST search

Contents

13.1 Running BLAST searches . . . 293
13.1.1 BLAST at NCBI . . . 293
13.1.2 BLAST against local data . . . 296
13.2 Output from BLAST searches . . . 300
13.2.1 Graphical overview for each query sequence . . . 300
13.2.2 Overview BLAST table . . . 300
13.2.3 BLAST graphics . . . 302
13.2.4 BLAST HSP table . . . 303
13.2.5 BLAST hit table . . . 305
13.2.6 Extracting a consensus sequence from a BLAST result . . . 306
13.3 Local BLAST databases . . . 306
13.3.1 Make pre-formatted BLAST databases available . . . 306
13.3.2 Download NCBI pre-formatted BLAST databases . . . 306
13.3.3 Create local BLAST databases . . . 308
13.4 Manage BLAST databases . . . 309
13.5 Bioinformatics explained: BLAST . . . 310
13.5.1 How does BLAST work? . . . 311
13.5.2 Which BLAST program should I use? . . . 313
13.5.3 Which BLAST options should I change? . . . 314
13.5.4 Where can I get the BLAST+ programs . . . 315
13.5.5 What you cannot get out of BLAST . . . 315
13.5.6 Other useful resources . . . 316

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 13.5.

Figure 13.8 shows an example of a BLAST result in the *CLC Genomics Workbench*.

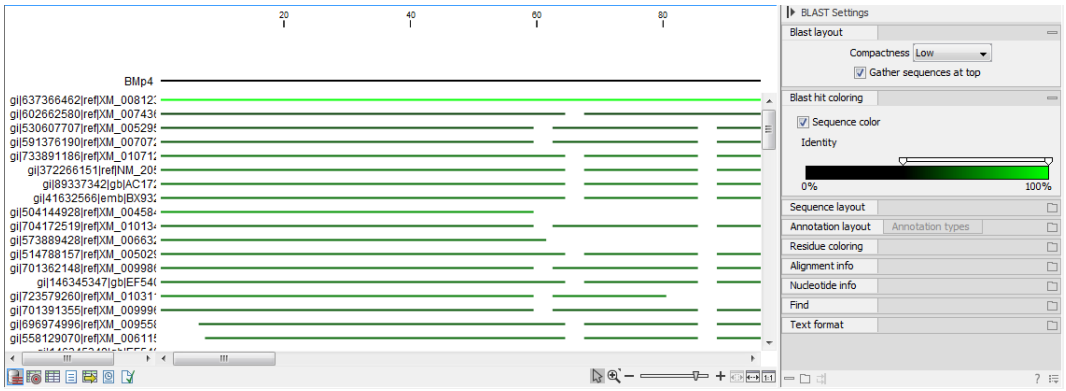


Figure 13.1: *Display of the output of a BLAST search. At the top is there a graphical representation of BLAST hits with tool-tips showing additional information on individual hits. Below is a tabular form of the BLAST results.*

13.1 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 (http://www.ncbi.nlm.nih.gov/) 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.

13.1.1 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:

Toolbox | **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 13.2

Select one or more sequences of the same type (either DNA or protein) and click **Next**.

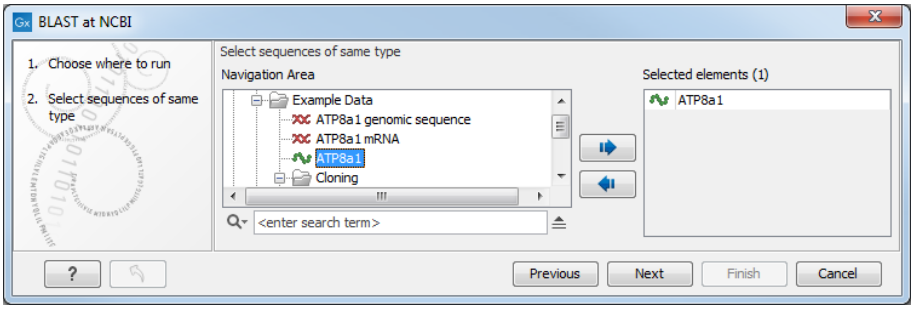


Figure 13.2: *Choose one or more sequences to conduct a BLAST search with.*

In this dialog, you choose which type of BLAST search to conduct, and which database to search against (figure 13.3). 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 chose 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.

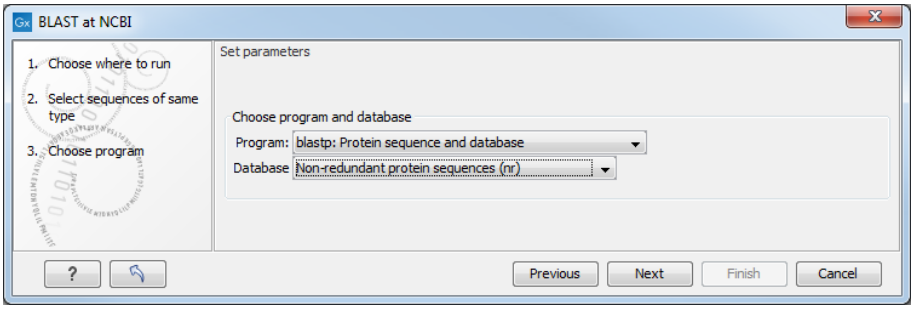


Figure 13.3: *Choose a BLAST Program and a database for the search.*

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 13.4, allows you to choose parameters to tune your BLAST search, to meet your requirements.

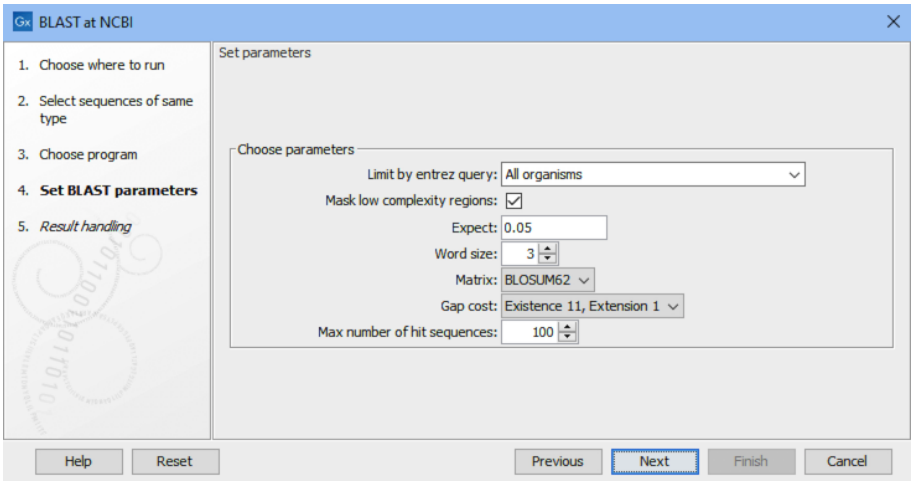


Figure 13.4: *Parameters that can be set before submitting a BLAST search.*

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.

The following description of BLAST search parameters is based on information from http://www.ncbi.nlm.nih.gov/BLAST/blastcgihelp.shtml.

- **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 http://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: http://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 notiation. 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 13.3 and the rest of the options are the same as when performing a BLAST search with a full sequence.

13.1.2 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 13.3.3)
- You can add pre-formatted databases (see section 13.3.1)
- You can use sequence data from the **Navigation Area** directly, without creating a database first.

To conduct a local BLAST search, go to:

Toolbox | **BLAST (**  **)**| **BLAST (**  **)**

This opens the dialog seen in figure 13.5:

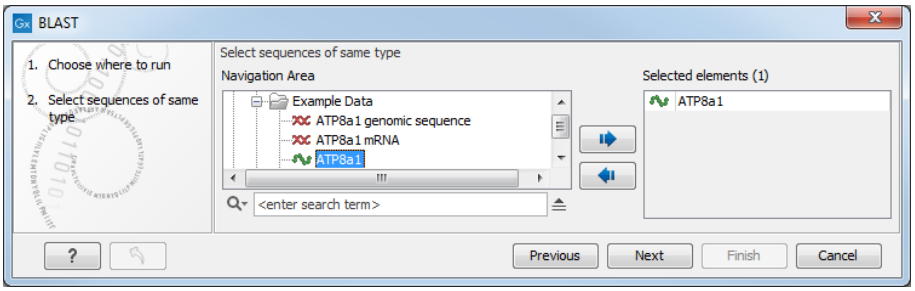


Figure 13.5: *Choose one or more sequences to conduct a BLAST search.*

Select one or more sequences of the same type (DNA or protein) and click **Next**.

This opens the dialog seen in figure 13.6:

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.
- **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.

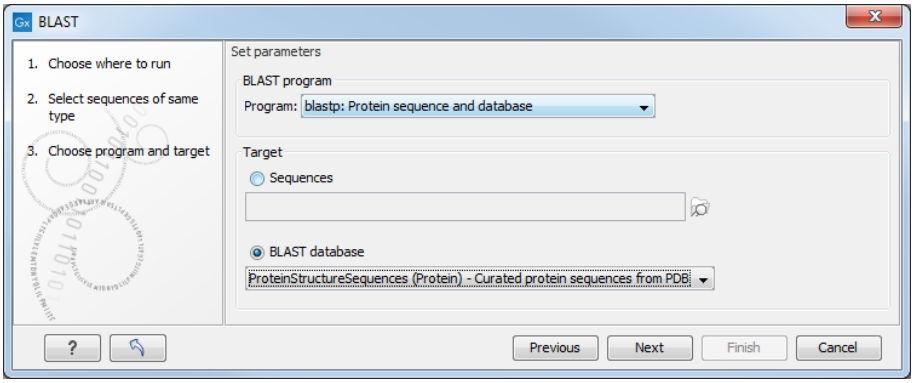


Figure 13.6: *Choose a BLAST program and a target database.*

- **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 14.1.3).

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 create a BLAST database first, see section 13.3.3.
- **BLAST Database**. Select a database already available in one of your designated BLAST database folders. Read more in section 13.4.

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 13.7).

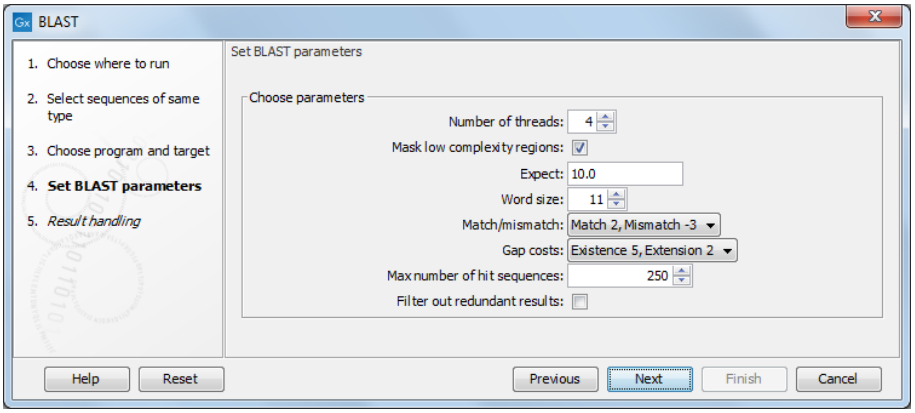


Figure 13.7: *Parameters that can be set before submitting a local BLAST search.*

- **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: `http://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 notiation. 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 13.6 and the rest of the options are the same as when performing a BLAST search with a full sequence.

13.2 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 13.2.2.

13.2.1 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 13.8. In cases where only one sequence was entered into a BLAST search, such a graphical overview is the default output.

Figure 13.8 shows an example of a BLAST result for an individual query sequence in the *CLC Genomics Workbench*.

Detailed descriptions of the overview BLAST table and the graphical BLAST results view are described below.

13.2.2 Overview BLAST table

In the overview BLAST table for a multi-sequence blast search, as shown in figure 13.9, there is one row for each query sequence. Each row represents the BLAST result for this query sequence.

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

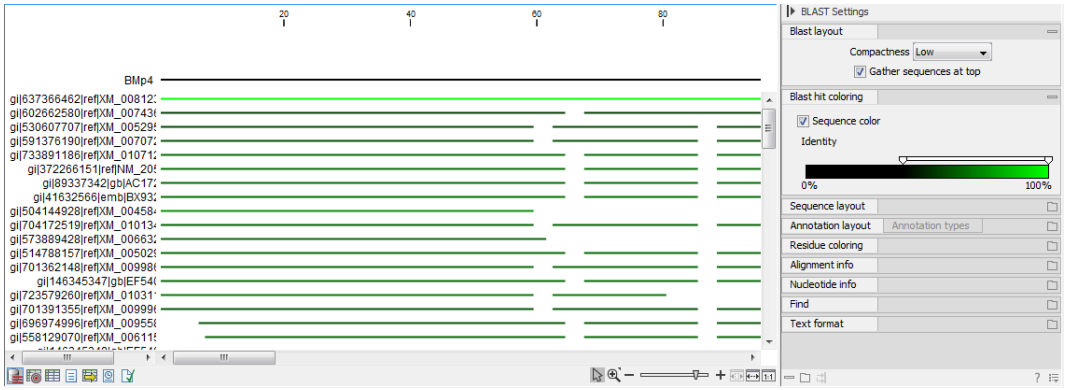


Figure 13.8: *Default display of the output of a BLAST search for one query sequence. At the top is there a graphical representation of BLAST hits with tooltips showing additional information on individual hits.*

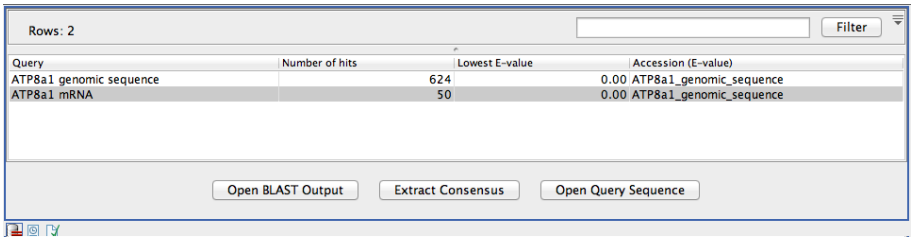


Figure 13.9: *An overview BLAST table summarizing the results for a number of query sequences.*

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.

13.2.3 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 12.

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.

13.2.4 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 13.10 is an example of a BLAST HSP Table.

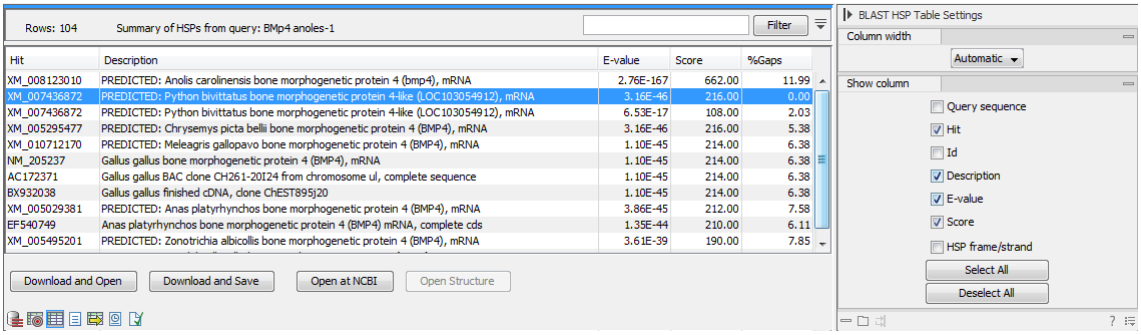


Figure 13.10: *BLAST HSP Table. The HSPs can be sorted by the different columns, simply by clicking the column heading.*

The BLAST HSP Table includes the following information:

- **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.

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 13.2.3 so that selecting a HSP in the table will make a selection on the corresponding sequence in the graphical view.

13.2.5 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 13.11 is an example of a BLAST Hit Table.

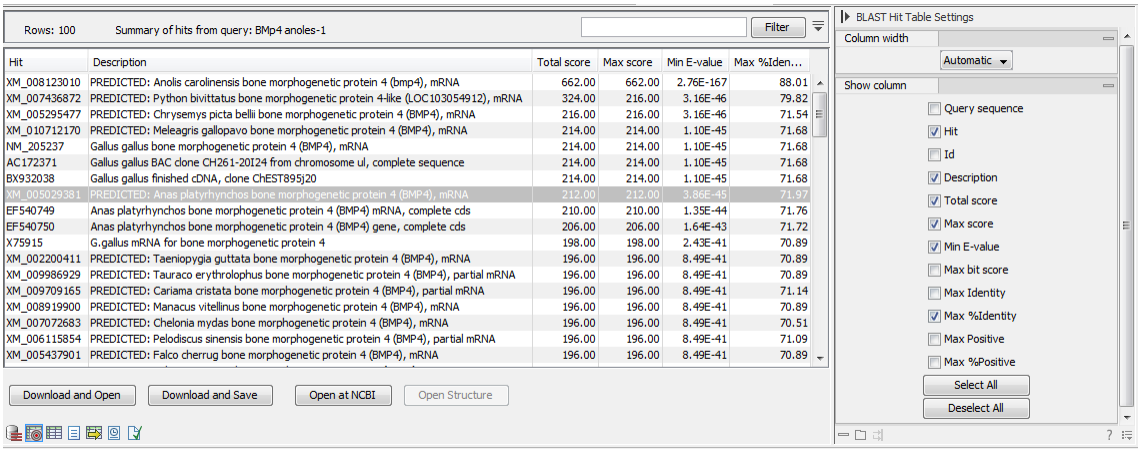


Figure 13.11: *BLAST Hit Table. The hits can be sorted by the different columns, simply by clicking the column heading.*

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.
- **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.

13.2.6 Extracting a consensus sequence from a BLAST result

A consensus sequence can be extracted from nucleotide BLAST results, as described in section 27.6. That section focuses on working with read mappings, but the same underlying tool is used to extract consensus sequences from nucleotide BLAST results.

13.3 Local BLAST databases

BLAST databases on your local system can be made available for searches via your *CLC Genomics Workbench* (see section 13.3.1). To make adding databases even easier, you can download pre-formatted BLAST databases from the NCBI from within your Workbench (section 13.3.2). You can also easily create your own local blast databases from sequences within your Workbench (section 13.3.3).

13.3.1 Make pre-formatted BLAST databases available

To use databases that have been downloaded or created outside the Workbench, you can either:

- Put the database files in one of the locations defined in the BLAST database manager (see section 13.4). All the files that comprise a given BLAST database must be included. This may be as few as three files, but can be more (figure 13.12).
- Add the location where your BLAST databases are stored using the BLAST database manager (see section 13.4).

13.3.2 Download NCBI pre-formatted BLAST databases

Many popular pre-formatted databases are available for download from the NCBI. You can download any of the databases available from the list at `ftp://ftp.ncbi.nlm.nih.gov/blast/db/` from within your Workbench.

You must be connected to the internet to use this tool.

To download a database, go to:

Toolbox | **BLAST** ( )| **Download BLAST Databases** ( )

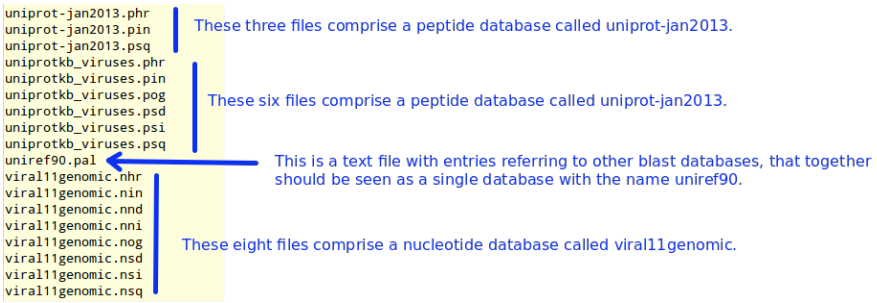


Figure 13.12: *BLAST databases are made up of several files. The exact number varies and depends on the tool used to build the databases as well as how large the database is. Large databases will be split into the number of volumes and there will be several files per volume. If you have made your BLAST database, or downloaded BLAST database files, outside the Workbench, you will need to ensure that all the files associated with that BLAST database are available in a CLC Blast database location.*

A window like the one in figure 13.13 pops up showing you the list of databases available for download.

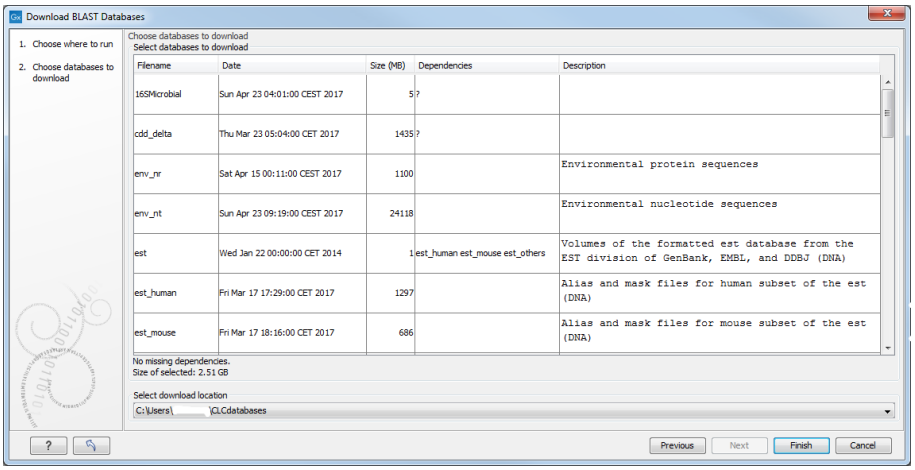


Figure 13.13: *Choose from pre-formatted BLAST databases at the NCBI available for download.*

In this window, you can see the names of the databases, the date they were made available for download on the NCBI site, the size of the files associated with that database, and a brief description of each database. You can also see whether the database has any dependencies. This aspect is described below.

You can also specify which of your database locations you would like to store the files in. Please see the **Manage BLAST Databases** section for more on this (section 13.4).

There are two very important things to note if you wish to take advantage of this tool.

- Many of the databases listed are very large. Please make sure you have space for them. If you are working on a shared system, we recommend you discuss your plans with your system administrator and fellow users.
- Some of the databases listed are dependent on others. This will be listed in the **Dependencies** column of the **Download BLAST Databases** window. This means that while the database your are interested in may seem very small, it may require that you also download a very big database on which it depends.

An example of the second item above is *Swissprot*. To download a database from the NCBI that would allow you to search just Swissprot entries, you need to download the whole *nr* database in addition to the entry for Swissprot.

13.3.3 Create local BLAST databases

You can create a local database that you can use for local BLAST searches. You can specify a location on your computer to save the BLAST database files to. The Workbench will list the BLAST databases found in these locations when you set up a local BLAST search (see section 13.1.2).

DNA, RNA, and protein sequences located in the **Navigation Area** can be used to create BLAST databases from. Any given BLAST database can only include one molecule type. If you wish to use a pre-formatted BLAST database instead, see section 13.3.1.

To create a BLAST database, go to:

Toolbox | **BLAST (**  **)**| **Create BLAST Database (**  **)**

This opens the dialog seen in figure 13.14.

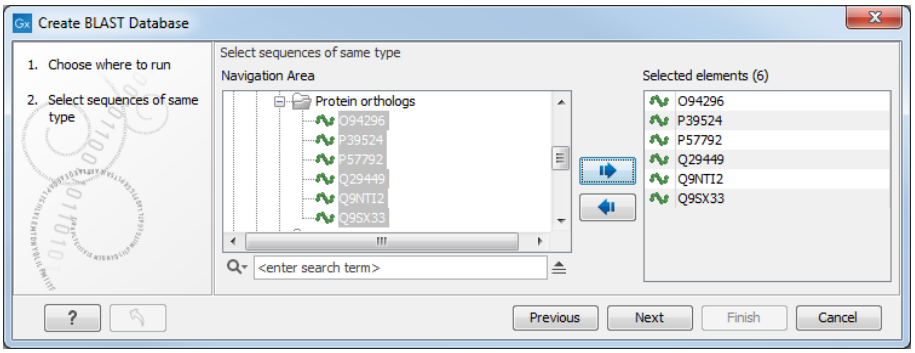


Figure 13.14: *Add sequences for the BLAST database.*

Select sequences or sequence lists you wish to include in your database and click **Next**.

In the next dialog, shown in figure 13.15, you provide the following information:

- **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 location to save the BLAST database files to. You can add or change the locations in this list using the **Manage BLAST Databases** tool, see section 13.4.

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 13.4, and when running local BLAST (see section 13.1.2).

Create BLAST Database creates BLAST+ version 4 (dbV4) databases.

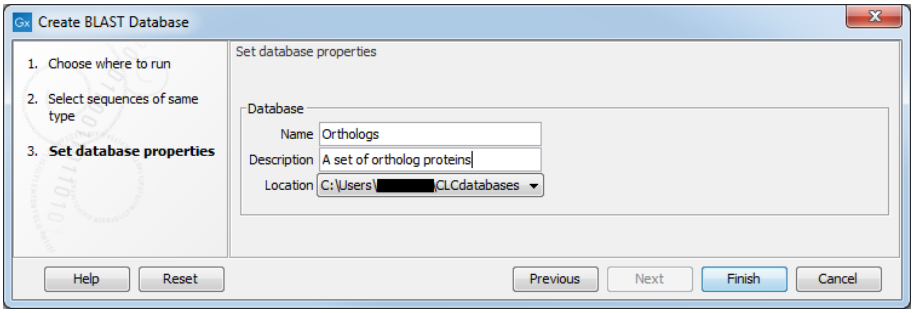


Figure 13.15: *Providing a name and description for the database, and the location to save the files to.*

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+. This, however, has the implication that databases created using **Create BLAST Databases** in *CLC Main Workbench*, *CLC Genomics Workbench* or *CLC Genomics Server* version 21.0 and later are intended for use only with the BLAST search tools in these software versions.

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.

13.4 Manage BLAST databases

The BLAST databases available as targets for running local BLAST searches (see section 13.1.2) can be managed through the Manage BLAST Databases dialog (see figure 13.16):

Toolbox | **BLAST** ( )| **Manage BLAST Databases** ( )

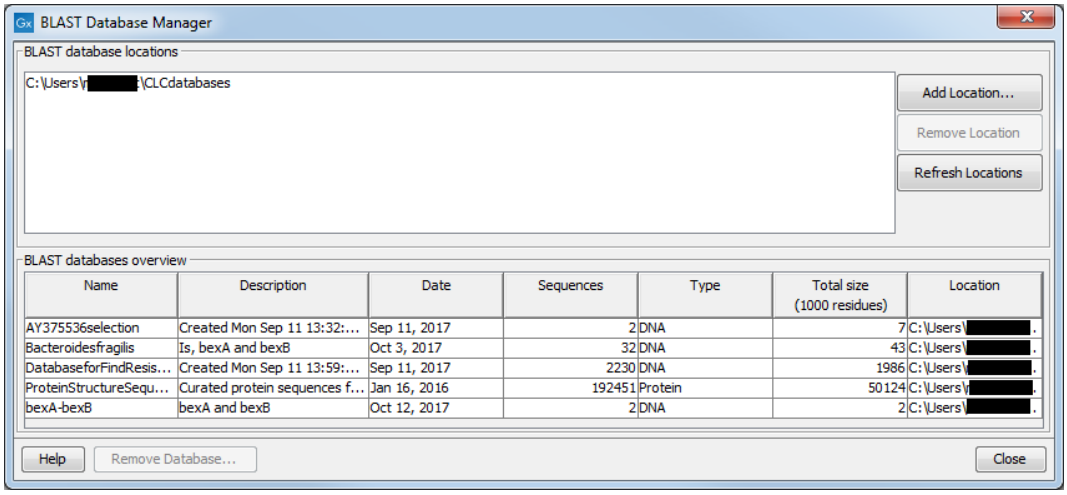


Figure 13.16: *Overview of available BLAST databases.*

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

from within the Workbench using the **Create BLAST Database tool**, see section 13.3.3, 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.

Note: The BLAST database location and all folders in its path should **not** have any spaces in their names.

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.

13.5 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.

13.5.1 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 13.17 for an illustration of words in a protein sequence.

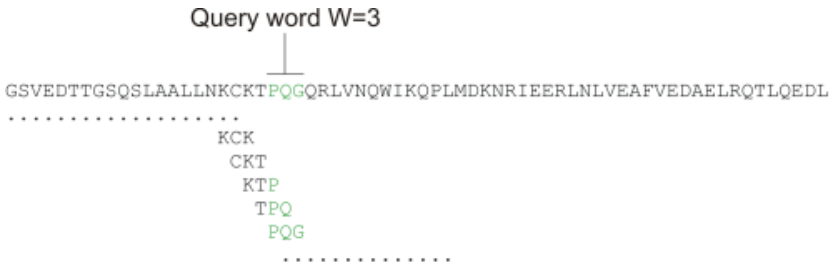


Figure 13.17: *Generation of exact BLAST words with a word size of W=3.*

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 13.17). 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 13.18). 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.

After the initial finding of words (seeding), the BLAST algorithm will extend the (only 3 residues long) alignment in both directions (see figure 13.19). Each time the alignment is extended, an

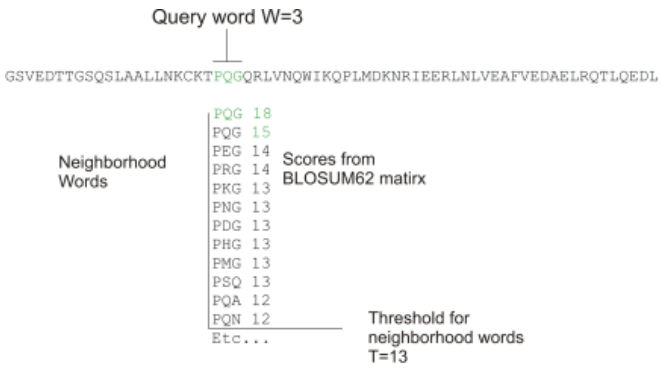


Figure 13.18: *Neighborhood BLAST words based on the BLOSUM62 matrix. Only words where the threshold* T *exceeds 13 are included in the initial seeding.*

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.

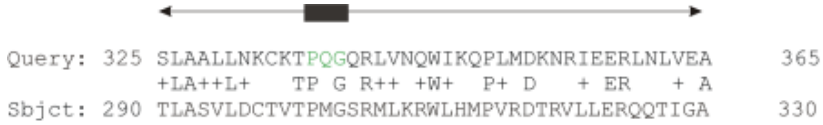


Figure 13.19: *Blast aligning in both directions. The initial word match is marked green.*

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 13.20.

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.

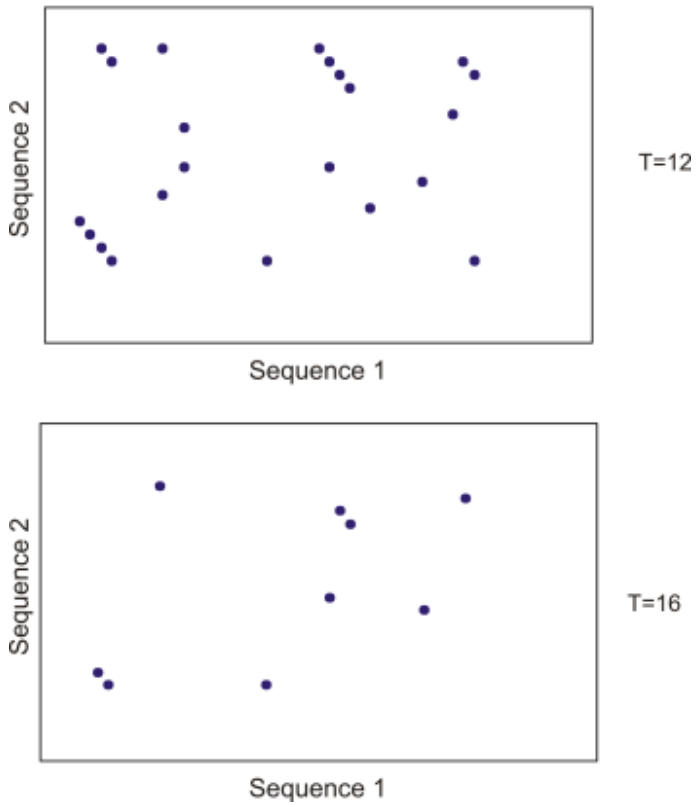


Figure 13.20: *Each dot represents a word match. Increasing the threshold of* T *limits the search space significantly.*

13.5.2 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.

13.5.3 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 'fftfflllsss', 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.

13.5.4 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 13.21). It is also much easier to batch download a selection of hit sequences for further inspection.

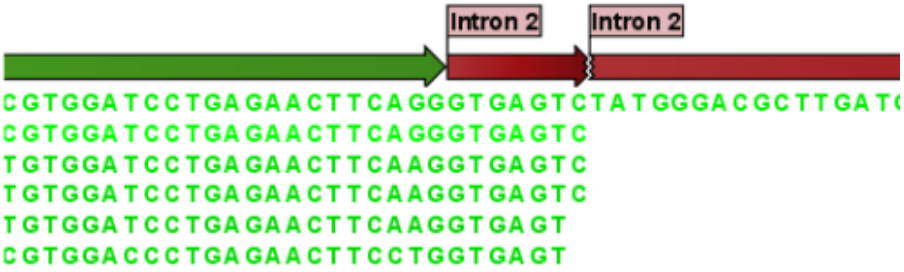


Figure 13.21: *Snippet of alignment view of BLAST results. Individual alignments are represented directly in a graphical view. The top sequence is the query sequence and is shown with a selection of annotations.*

13.5.5 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.

13.5.6 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 14

3D Molecule Viewer

Contents

14.1 Importing molecule structure files 318
14.1.1 From the Protein Data Bank 318
14.1.2 From your own file system 319
14.1.3 BLAST search against the PDB database 320
14.1.4 Import issues 322
14.2 Viewing molecular structures in 3D 322
14.3 Customizing the visualization 323
14.3.1 Visualization styles and colors 324
14.3.2 Project settings 329
14.4 Tools for linking sequence and structure 332
14.4.1 Show sequence associated with molecule 332
14.4.2 Link sequence or sequence alignment to structure 333
14.4.3 Transfer annotations between sequence and structure 334
14.5 Align Protein Structure 336
14.5.1 Example: alignment of calmodulin 337
14.5.2 The Align Protein Structure algorithm 339
14.6 Generate Biomolecule 340

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) `http://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.

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

14.1 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

14.1.1 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 14.1). The search hits will appear in the table below the search field.

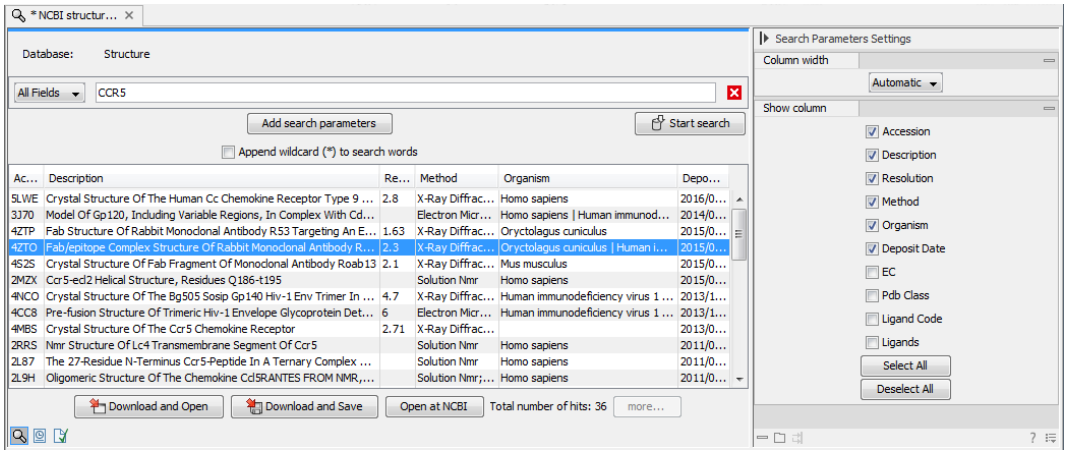


Figure 14.1: *Download protein structure from the Protein Data Bank. It is possible to open a structure file directly from the output of the search by clicking the "Download and Open" button or by double clicking directly on the relevant row.*

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.

14.1.2 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 14.2). 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**.

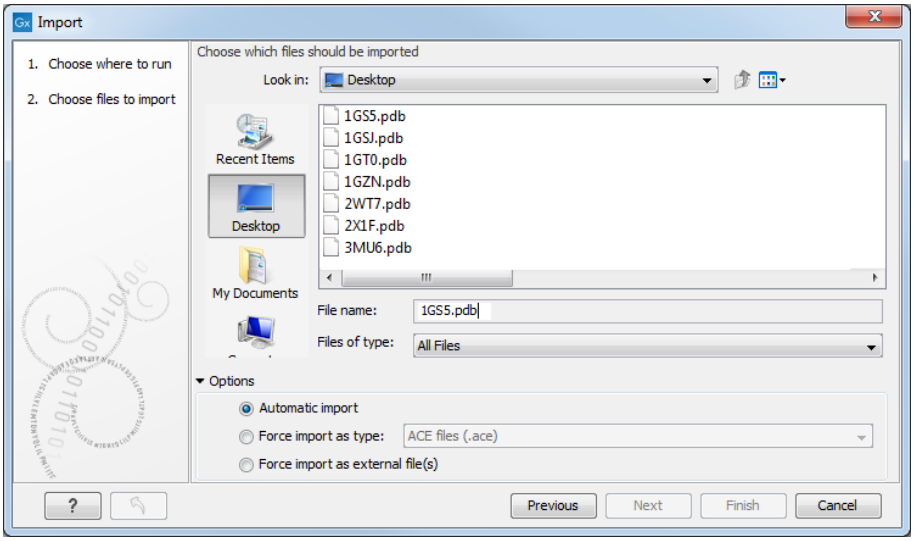


Figure 14.2: *A PDB file can be imported using the Standard Import tool.*

14.1.3 BLAST search against the PDB database

It is also possible to make a BLAST search against the PDB database, by going to:

Toolbox | **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 14.3).

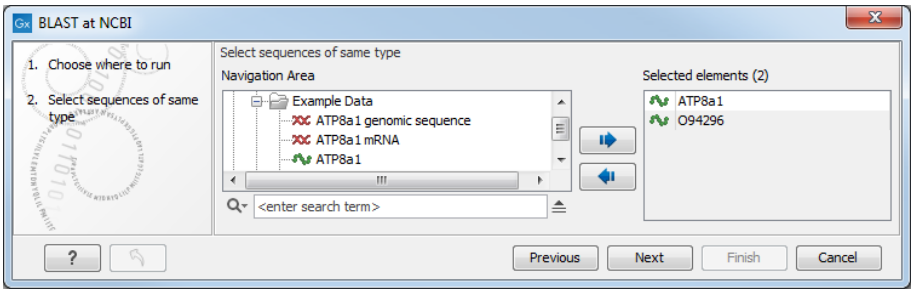


Figure 14.3: *Select the input sequence of interest. In this example a protein sequence for ATPase class I type 8A member 1 and an ATPase ortholog from S. pombe have been selected.*

Click **Next** and choose program and database (figure 14.4). 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 13.1.1 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.

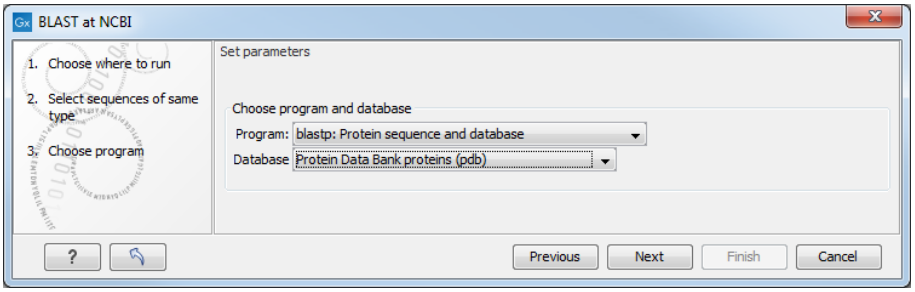


Figure 14.4: *Select database and program.*

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 14.5). 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**.

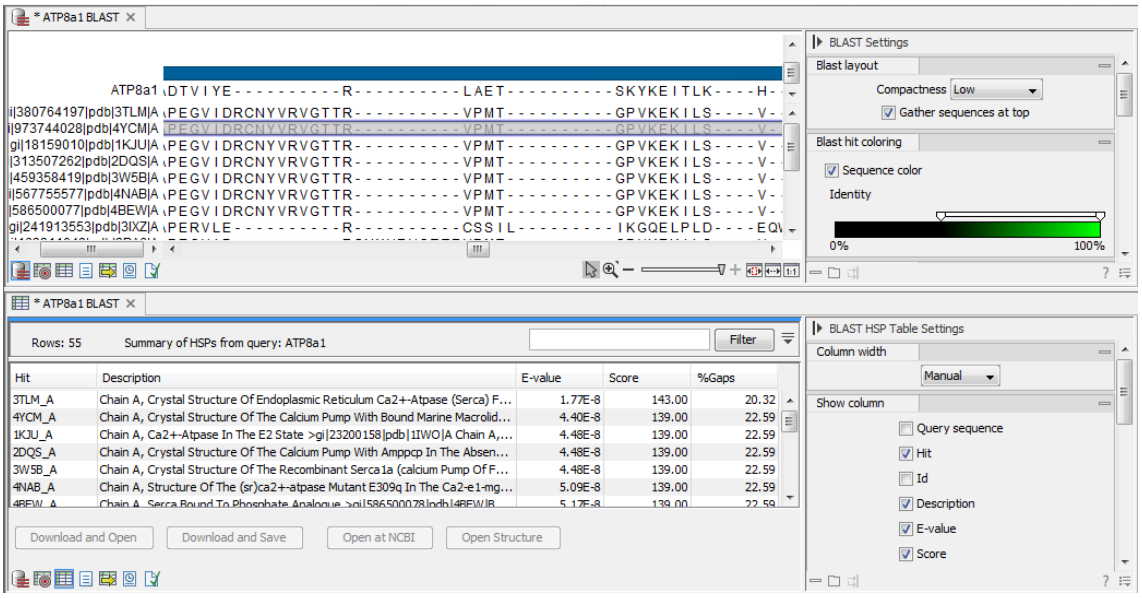


Figure 14.5: *Top: The output from "BLAST at NCBI". Bottom: The "BLAST table". One of the protein sequences has been selected. This activates the four buttons under the table. Note that the table and the BLAST Graphics are linked, this means that when a sequence is selected in the table, the same sequence will be highlighted in the BLAST Graphics view.*

14.1.4 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 14.6).

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.

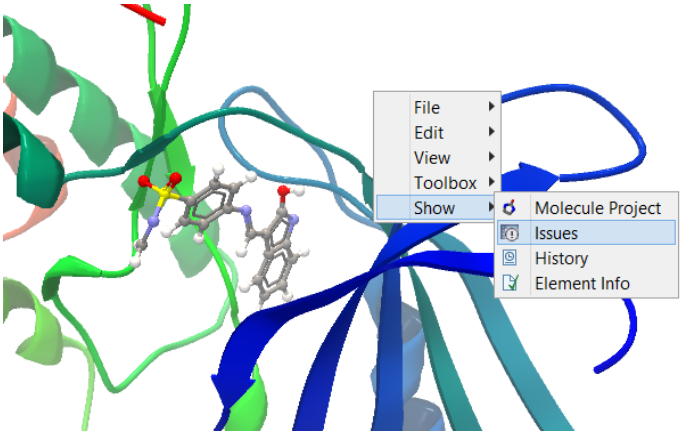


Figure 14.6: *At the bottom of the Molecule Project it is possible to switch to the "Show Issues" view by clicking on the "table-with-exclamation-mark" icon.*

14.2 Viewing molecular structures in 3D

An example of a 3D structure that has been opened as a **Molecule Project** is shown in figure 14.7.

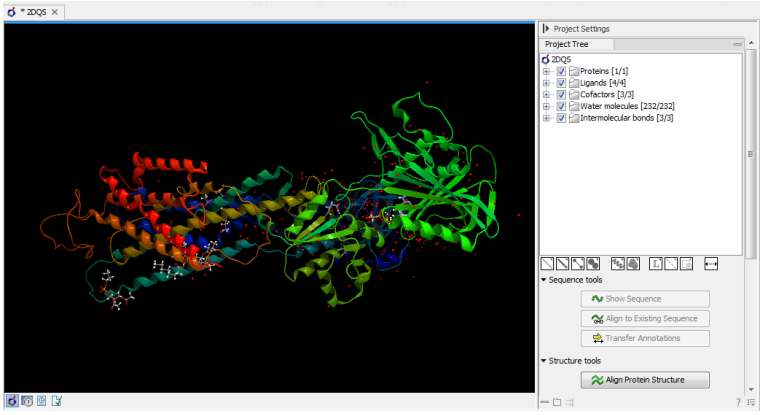


Figure 14.7: *3D view of a calcium ATPase. All molecules in the PDB file are shown in the Molecule Project. The Project Tree in the right side of the window lists the involved molecules.*

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-cheking 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.

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.

14.3 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.

14.3.1 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 Cα 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 14.3.1)

It is possible to adjust the opacity of a surface by adjusting the transparency slider at the bottom of the menu.

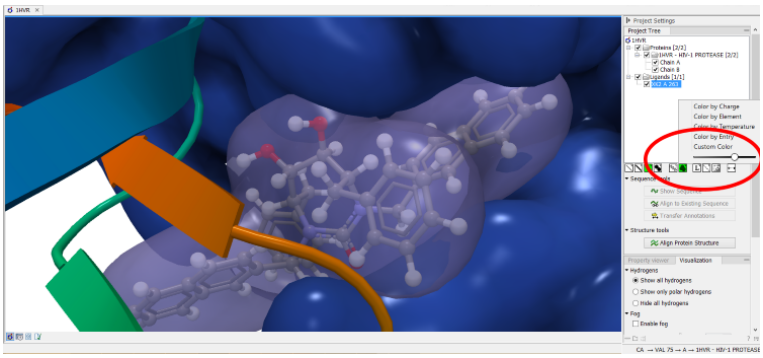


Figure 14.8: *Transparent surfaces*

Notice that visual artifacts may appear when rotating a transparent surface. These artifacts disappear as soon as the mouse is released.

Labels

( )

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 14.9).

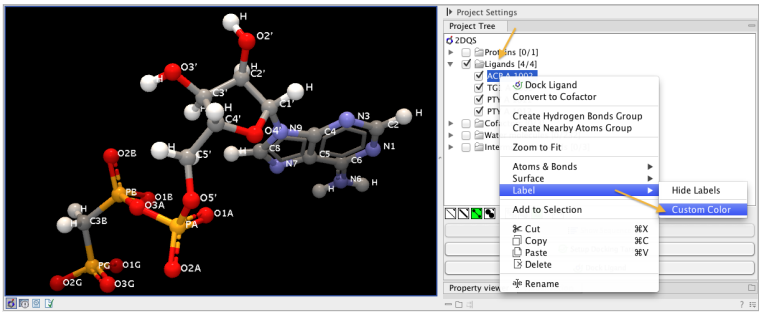


Figure 14.9: *The color of the labels can be adjusted in two different ways. Either directly using the label button by right clicking the button, or by right clicking on the molecule or category of interest in the Project Tree.*

- 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

( )

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.

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

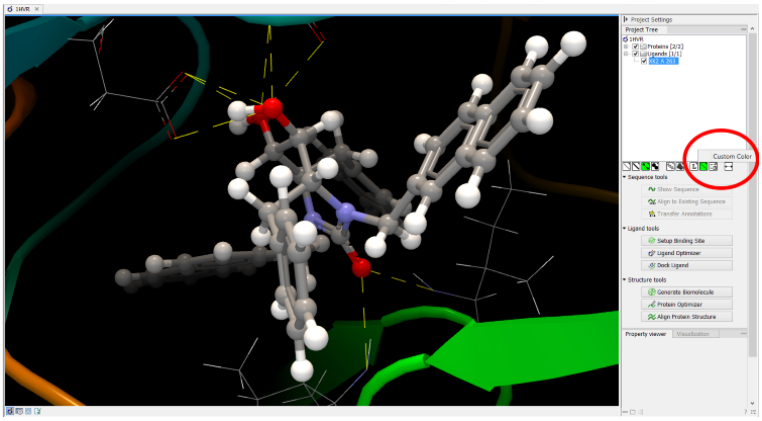


Figure 14.10: *The hydrogen bond visualization setting, with custom bond color.*

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.

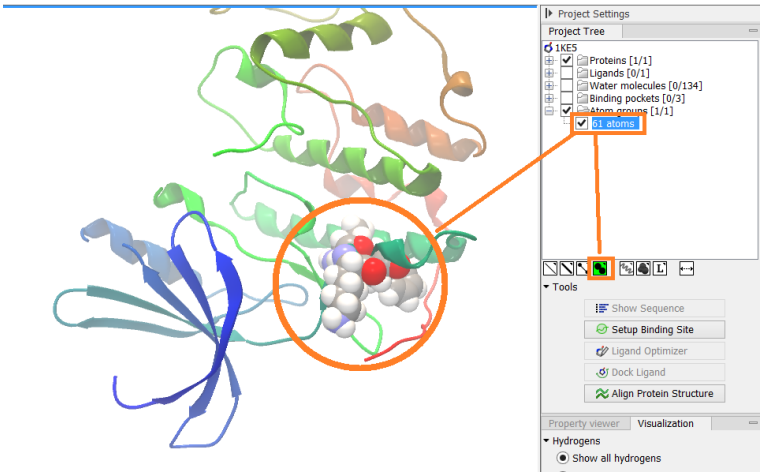


Figure 14.11: *An atom group that has been highlighted by adding a unique visualization style.*

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 (section 14.4.1). A split-view will appear with a sequence editor for each of the sequence data types (Protein, DNA, RNA) (figure 14.12). 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 (section 14.4.2). 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.

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.

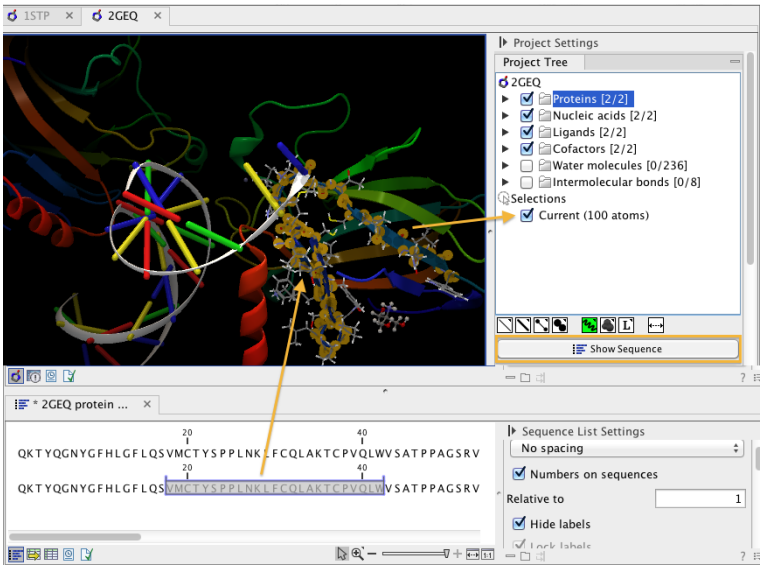


Figure 14.12: *The protein sequence in the split view is linked with the protein structure. This means that when a part of the protein sequence is selected, the same region in the protein structure will be selected.*

- 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

(⇠⇢)

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 (⇠⇢) at the bottom of the Project Tree view (figure 14.13). Double-clicking an entry in the Project Tree will have the same effect.

14.3.2 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.

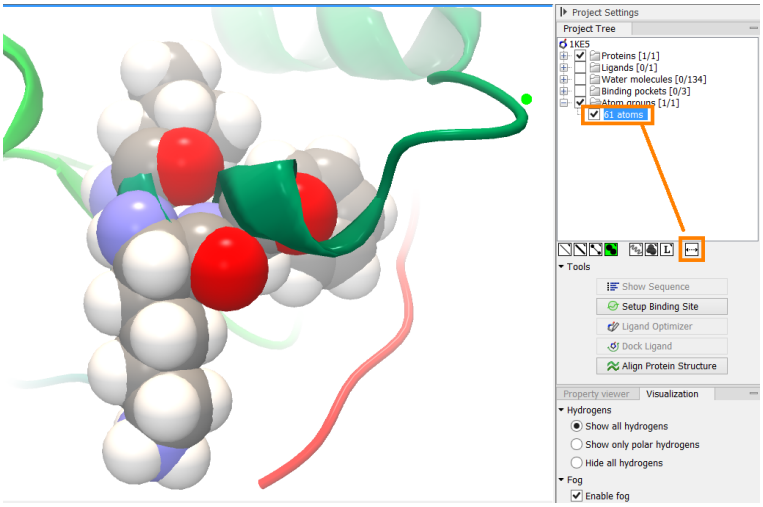


Figure 14.13: *The "Fit to screen" button can be used to bring a particular molecule or category of molecules in focus.*

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 14.4.1.
- **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 14.4.2.
- **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 14.4.3.
- **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 14.5.

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 17.6.2. 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.

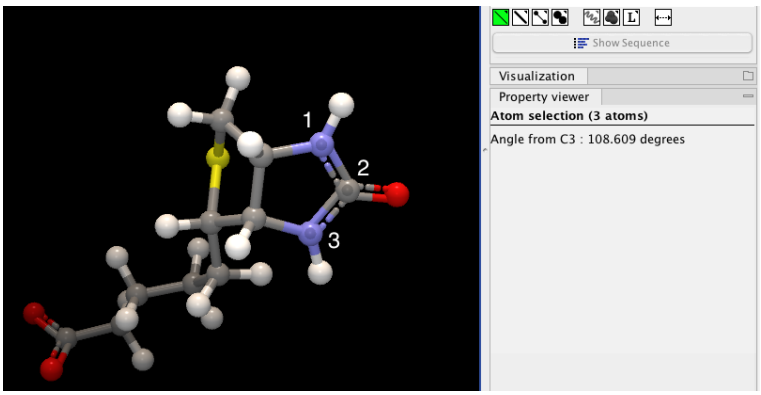


Figure 14.14: *Selecting two, three, or four atoms will display the distance, angle, or dihedral angle, respectively.*

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.

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).

14.4 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 14.4.3).

14.4.1 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 14.15). 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 14.3.1. 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 14.4.2).

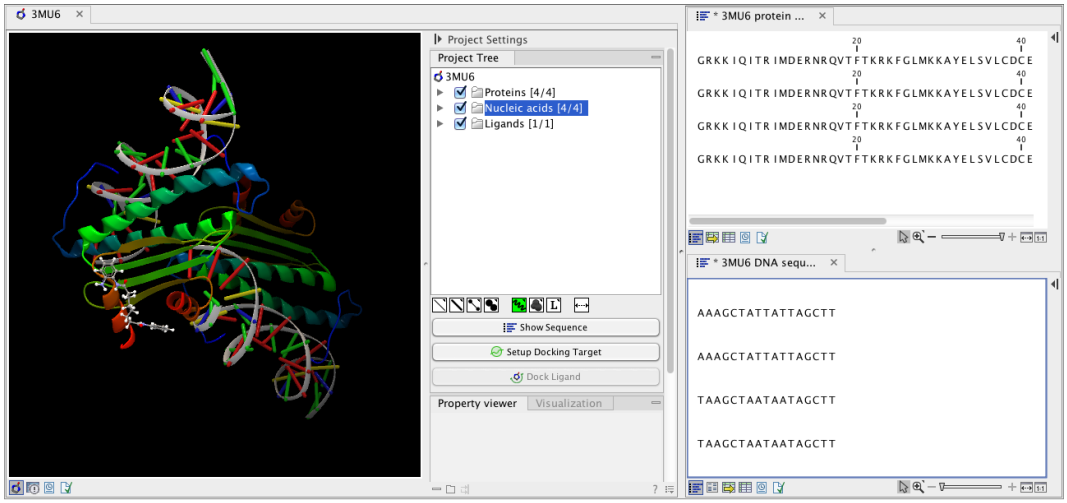


Figure 14.15: *Protein chain sequences and DNA sequences are shown in separate views.*

14.4.2 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 14.4.1) or Align to Existing Sequence.

Select a single protein chain in the project tree (see figure 14.16). 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.

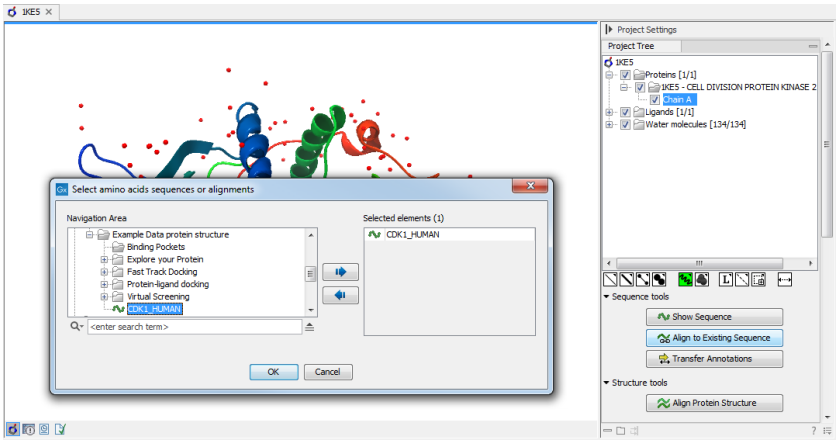


Figure 14.16: *Select a single protein chain in the Project Tree and invoke "Align to Existing Sequence".*

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 - section 14.4.1), 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 21.1) with the following settings:

- Gap open cost: 10.0

- 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 14.4.3). 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 21.2
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 14.19 and section 21.3).

14.4.3 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 12.3 and more about atom groups in section 14.3.1.

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 14.4.1) or by mapping to an existing sequence or sequence alignment using the 'Align to Existing Sequence' button (see section 14.4.2).

Invoke the Transfer Annotations dialog by selecting a linked protein chain in the Project Tree and press 'Transfer Annotations' (see figure 14.17).

The dialog contains two tables (see figure 14.18). 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

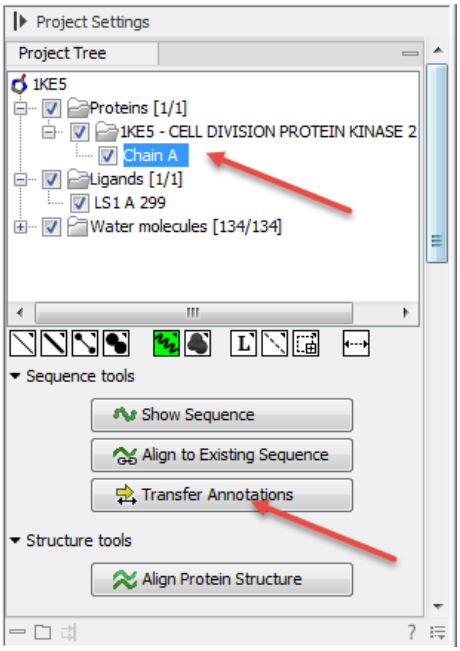


Figure 14.17: *Select a single protein chain in the Project Tree and invoke "Transfer Annotations".*

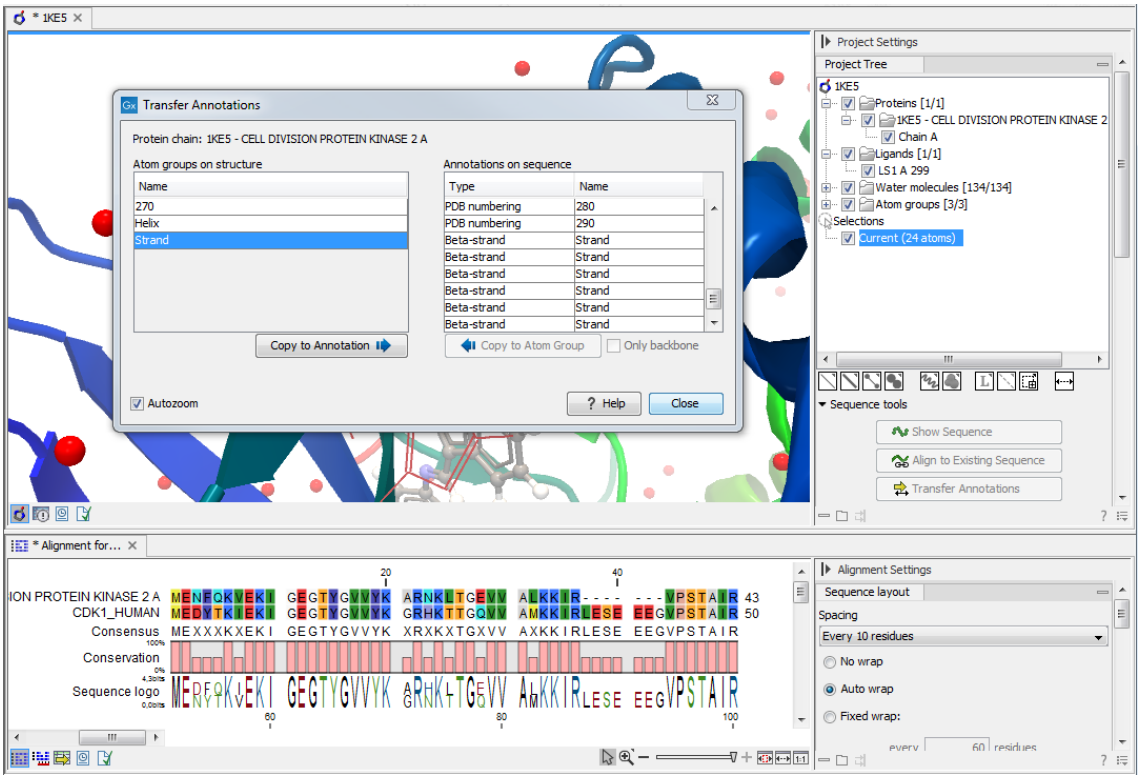


Figure 14.18: *The Transfer Annotations dialog allow you to select annotations listed in the two tables, and copy them from structure to sequence or vice versa.*

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 14.19 and section 21.3).

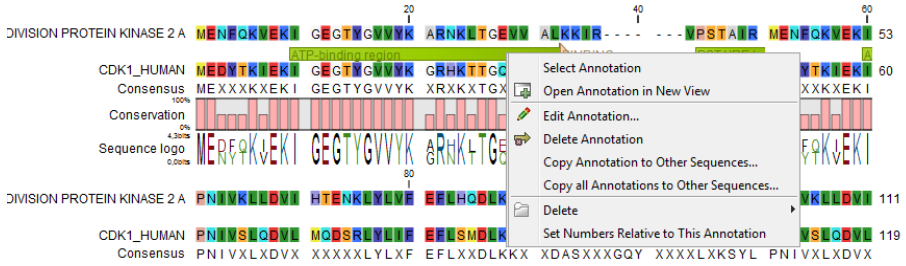


Figure 14.19: *Copy annotations from sequences in the alignment to the sequence linked to the 3D view.*

14.5 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 14.20). 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:

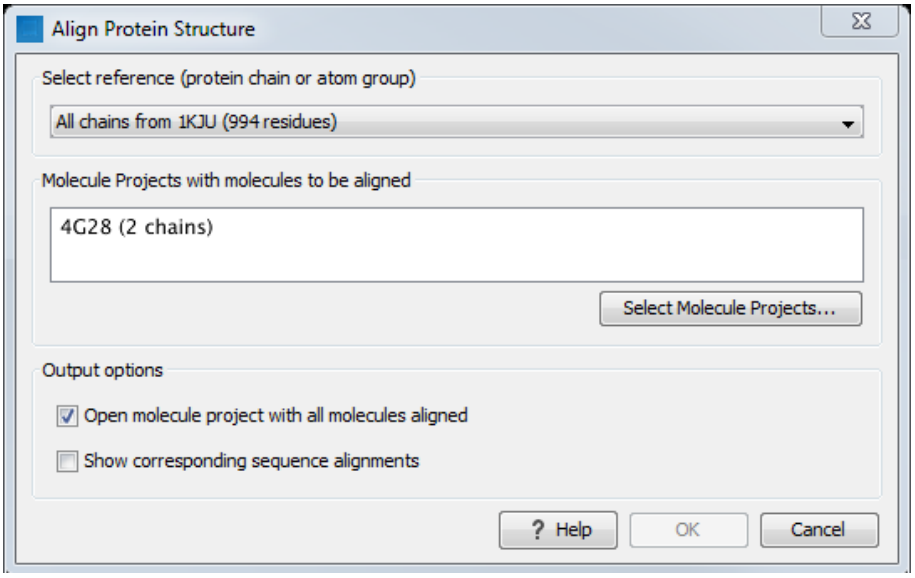


Figure 14.20: *The Align Protein Structure dialog box.*

- **Select reference (protein chain or atom group)** This drop-down menu shows all the protein 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.

14.5.1 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 14.20. 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 14.21. 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).

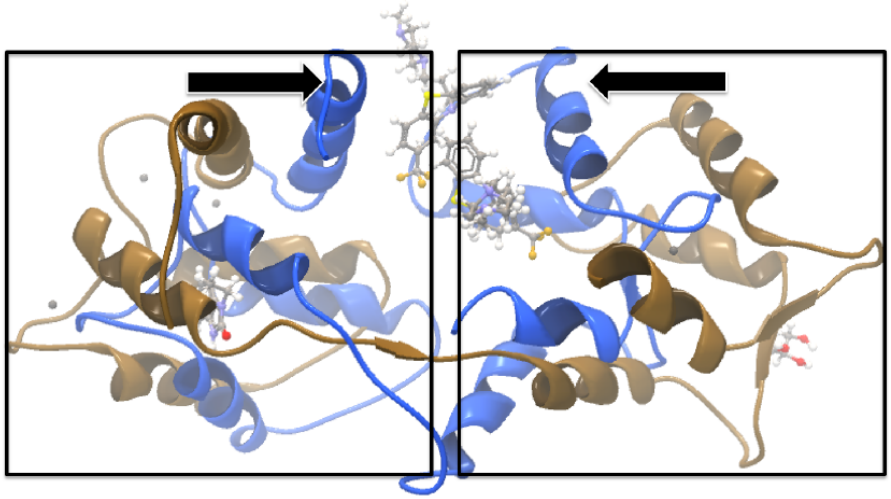


Figure 14.21: *Global alignment of two calmodulin structures (blue and brown). The two domains of calmodulin (shown within black boxes) can undergo large changes in relative orientation. In this case, the different orientation of the domains in the blue and brown structures makes a good global alignment 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 14.22). Using the new atom group as the reference in the alignment dialog leads to the alignment shown in figure 14.23. 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.

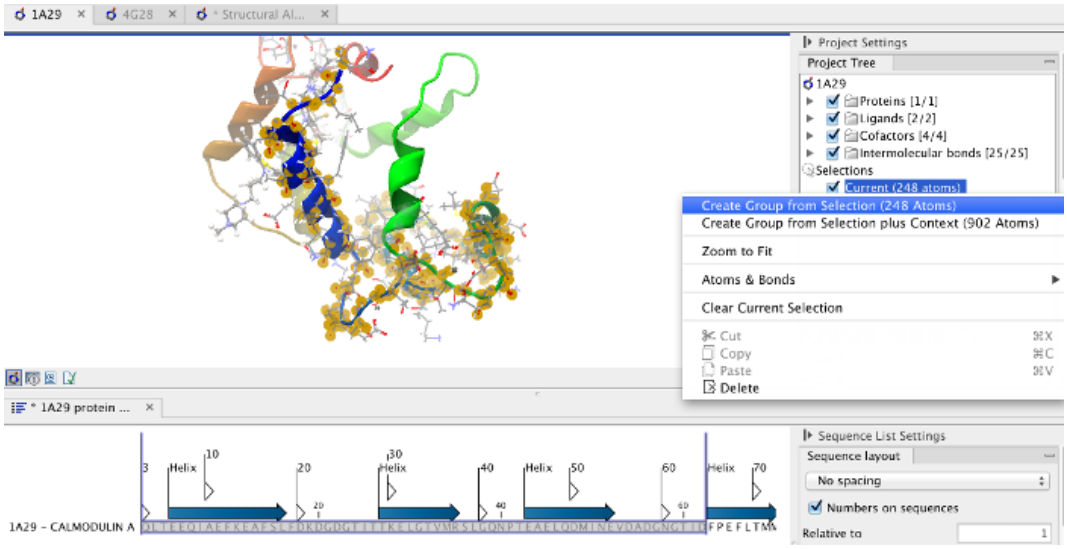


Figure 14.22: *Creation of an atom group containing the N-terminal domain of calmodulin.*

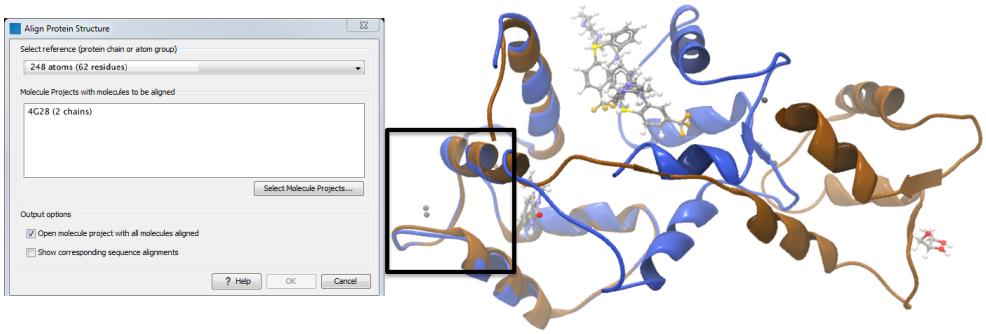


Figure 14.23: *Alignment of the same two calmodulin proteins as in figure 14.21, but this time with a focus on the N-terminal domain. The blue and brown structures are now well-superimposed in the N-terminal region. The black box encloses two calcium atoms that are bound to the structures.*

Aligning a binding site Two bound calcium atoms, one from each calmodulin structure, are shown in the black box of figure 14.23. 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 14.24.

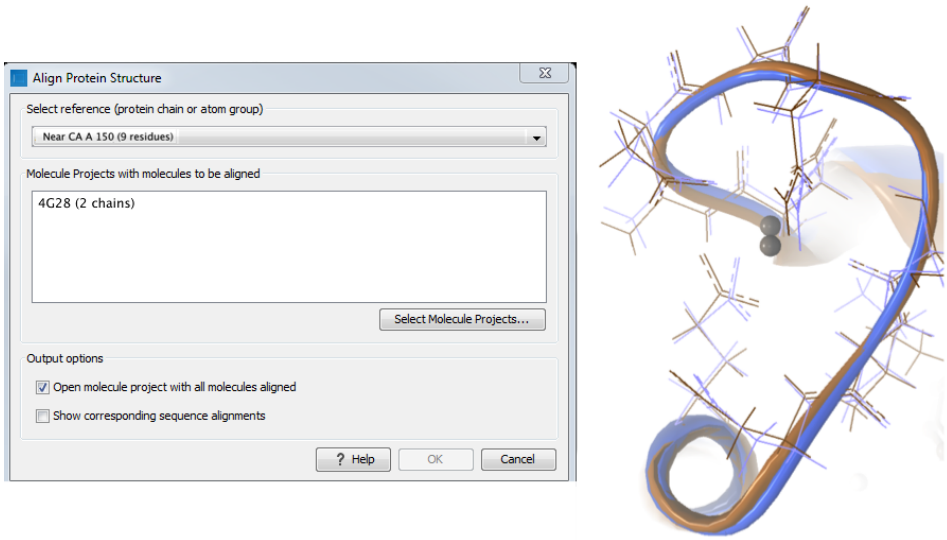


Figure 14.24: *Alignment of the same two calmodulin domains as in figure 14.21, but this time with a focus on the calcium atom within the black box of figure 14.23. The calcium atoms are less than 1 Å apart – compatible with thermal motion encoded in the atoms' temperature factors.*

14.5.2 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 ith 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).

14.6 Generate Biomolecule

Protein structures imported from a PBD 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 14.25). The button ( ) is found in the Structure tools section below the Project Tree.

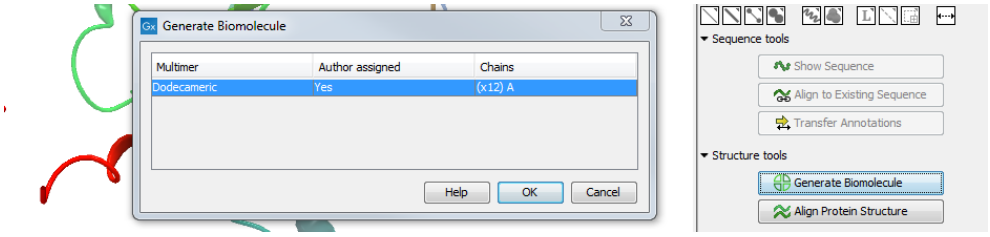


Figure 14.25: *The Generate Biomolecule dialog lists all possibilities for biomolecules, as given in the PDB files imported to the Molecule Project. In this case, only one biomolecule option is available. The Generate Biomolecule button that invokes the dialog can be seen in the bottom right corner of 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 14.26). 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.

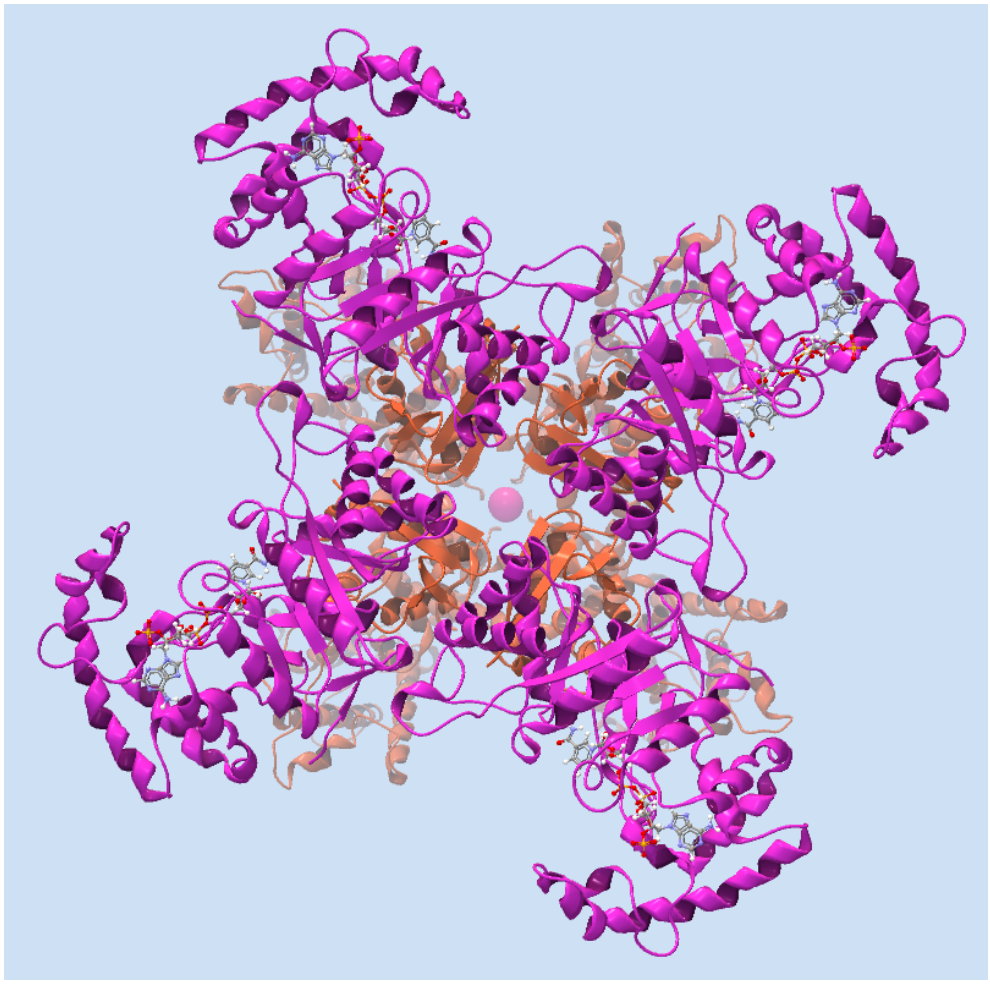


Figure 14.26: *One of the biomolecules that can be generated after downloading the PDB 2R9R to* **CLC Genomics Workbench**. *It is a voltage gated potassium channel.*

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.

Chapter 15

General sequence analyses

Contents

15.1 Extract sequences . . . 342
15.2 Shuffle sequence . . . 344
15.3 Dot plots . . . 345
15.3.1 Create dot plots . . . 345
15.3.2 View dot plots . . . 346
15.3.3 Bioinformatics explained: Dot plots . . . 347
15.3.4 Bioinformatics explained: Scoring matrices . . . 351
15.4 Local complexity plot . . . 353
15.5 Sequence statistics . . . 354
15.5.1 Bioinformatics explained: Protein statistics . . . 356
15.6 Join Sequences . . . 358
15.7 Pattern discovery . . . 359
15.7.1 Pattern discovery search parameters . . . 360
15.7.2 Pattern search output . . . 361
15.8 Motif Search . . . 361
15.8.1 Dynamic motifs . . . 361
15.8.2 Motif search from the Toolbox . . . 363
15.8.3 Java regular expressions . . . 365
15.9 Create motif list . . . 366

CLC Genomics Workbench offers different kinds of sequence analyses that apply to both protein and DNA.

The analyses are described in this chapter.

15.1 Extract sequences

This tool allows the extraction of sequences from other types of data in the Workbench, such as sequence lists or alignments. The data types you can extract sequences from are:

- Alignments ()

- BLAST result ( ) For BLAST results, the sequence hits are extracted but not the original query sequence or the consensus sequence.
- BLAST overview tables ( )
- sequence lists ( )
- 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 ( )

Note that paired reads will be extracted in accordance with the read group settings, which is 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.

Note! When the Extract Sequences tool is run via the Workbench toolbox on an entire file of one of the above types, **all** sequences are extracted from the data used as input. If only a **subset** of the sequences is desired, for example, the reads from just a small area of a mapping, or the sequences for only a few blast results, then a data set containing just this subsection or subset should be created and the Extract Sequences tool should be run on that. For extracting a subset of a mapping, please see section 19.7.6.

The Extract Sequences tool can be launched via the Toolbox menu, by going to:

> **Toolbox** | **Classical Sequence Analysis** ( ) | **General Sequence Analysis** ( )| **Extract Sequences** ( )

First select the elements from which sequences should be extracted, and click **Next**. The following dialog (figure 15.1) allows you to choose whether the extracted sequences should be extracted as single sequences or placed in a new sequence list. For most data types, it will make most sense to choose to extract the sequences into a sequence list. But when working with a sequence list, where choosing to "extract to a new sequence list" would just create a copy of the same sequence list, choose to "extract to single sequences" to generate individual sequence objects for each sequence in the sequence list.

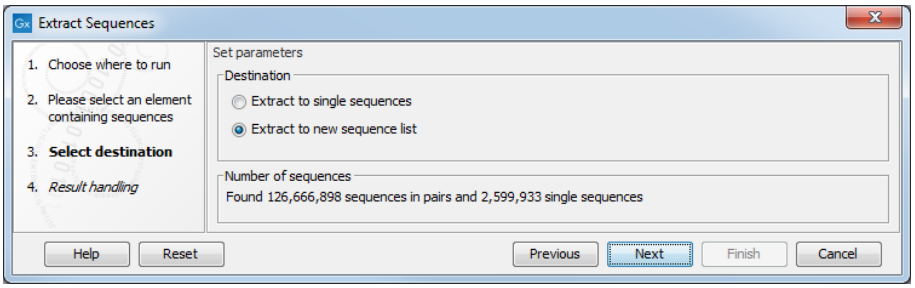


Figure 15.1: *Choosing whether the extracted sequences should be placed in a new list or as single sequences.*

Below these options, in the dialog, you can see the number of sequences that will be extracted. Click **Next** to choose where to save the output, and **Finish** to start the tool.

15.2 Shuffle sequence

In some cases, it is beneficial to shuffle a sequence. This is an option in the **Toolbox** menu under **General Sequence Analyses**. It is normally used for statistical analyses, e.g. 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 tool, go to:

> **Toolbox** | **Classical Sequence Analysis** ( ) | **General Sequence Analysis** ( )| **Shuffle Sequence** ( )

Choose a sequence for shuffling. If a sequence was selected before choosing the Toolbox action, 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** to determine how the shuffling should be performed.

In this step, shown in figure 15.2:

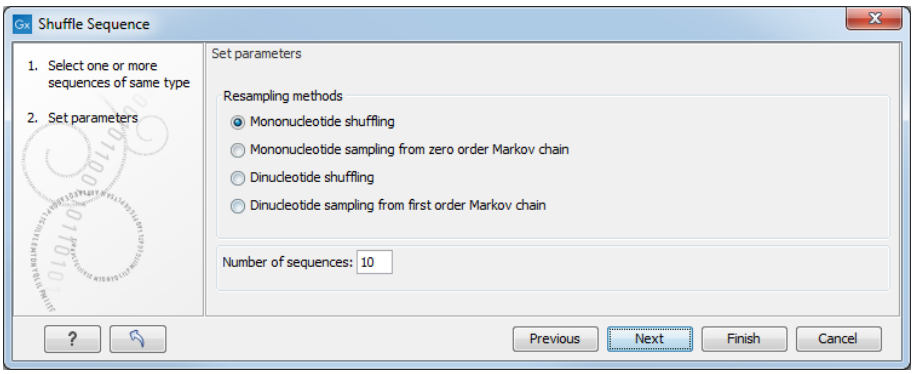


Figure 15.2: *Parameters for shuffling.*

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.

15.3 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.

15.3.1 Create dot plots

To create a dot plot, go to:

> **Toolbox** | **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 15.3).

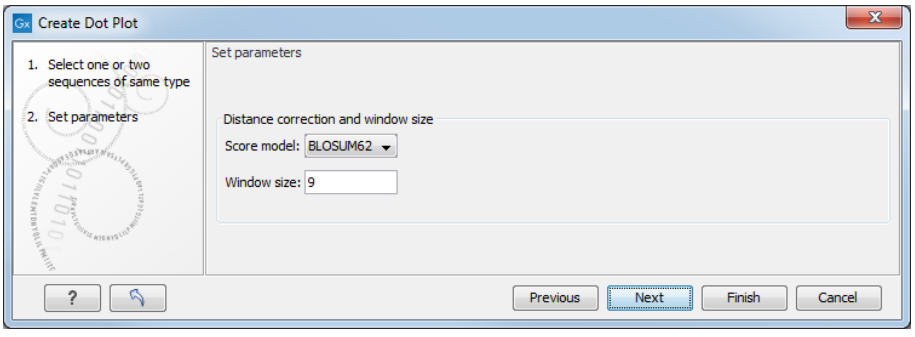


Figure 15.3: *Setting the dot plot parameters.*

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.

15.3.2 View dot plots

A view of a dot plot can be seen in figure 15.4. You can select **Zoom in** ( ) in the Toolbar and click the dot plot to zoom in to see the details of particular areas.

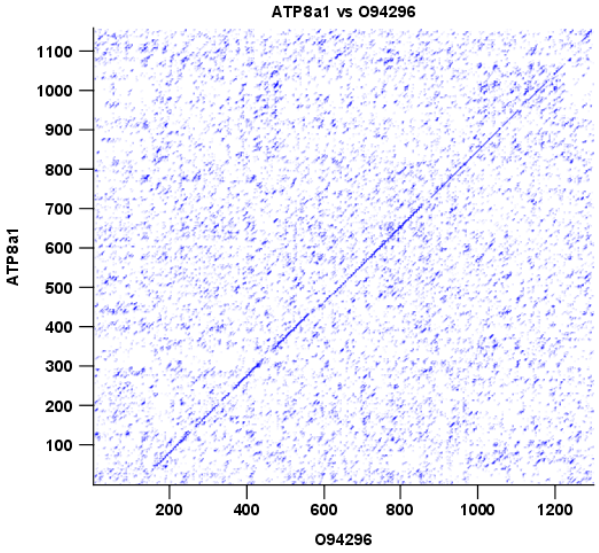


Figure 15.4: *A view is opened showing the dot plot.*

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 15.5).

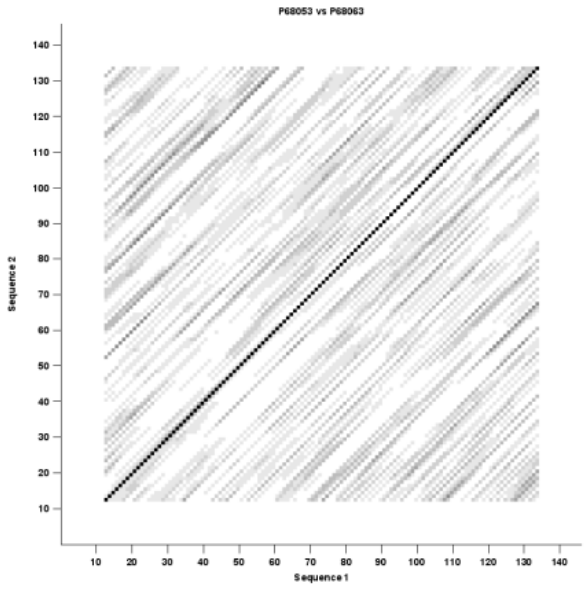


Figure 15.5: *Dot plot with inverted colors, practical for printing.*

15.3.3 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.
  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 15.6 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`.

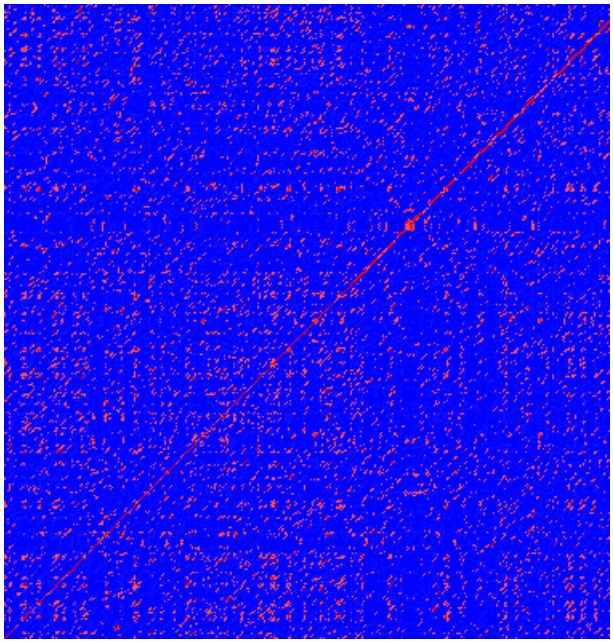


Figure 15.6: *Dot plot of DQ232610 vs. DQ023146 (Influenza A virus nucleoproteins) showing and overall similarity*

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 15.7: *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 15.8 you can see a sequence with repeats.

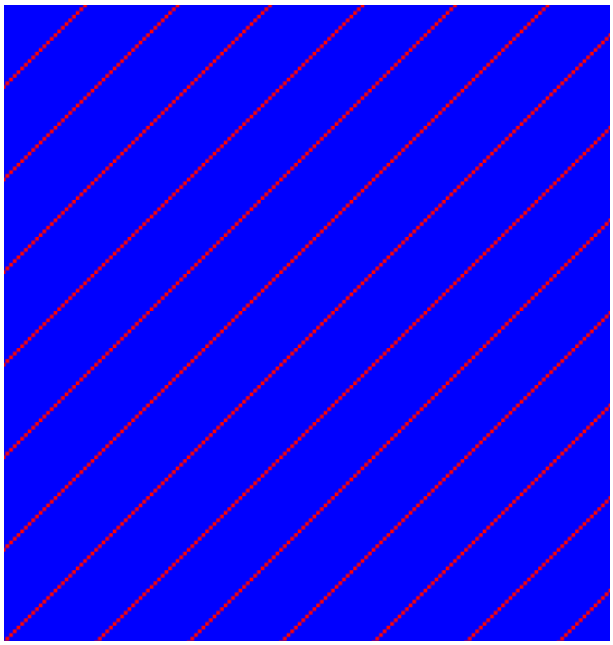


Figure 15.8: *The dot plot of a sequence showing repeated elements. See also figure 15.7.*

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 15.9. 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)

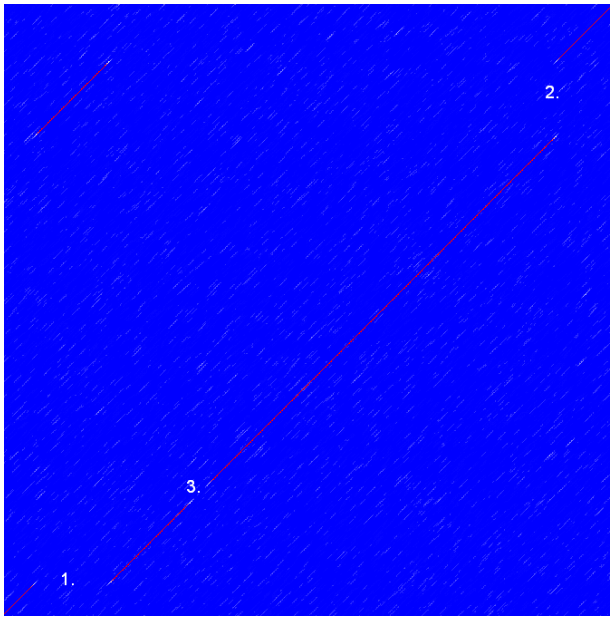


Figure 15.9: *This dot plot show various frame shifts in the sequence. See text for details.*

Sequence inversions In dot plots you can see an inversion of sequence as contrary diagonal to the diagonal showing similarity. In figure 15.10 you can see a dot plot (window length is 3) with an inversion.

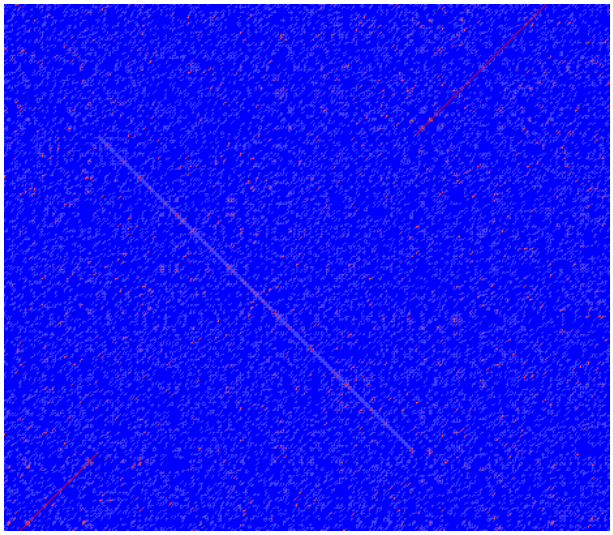


Figure 15.10: *The dot plot showing an inversion in a sequence. See also figure 15.7.*

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 15.11 is a square shows the low-complexity region of this sequence.

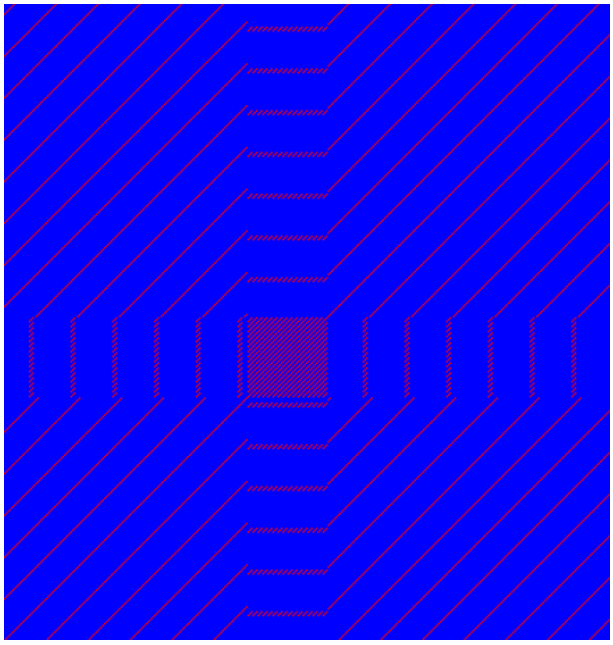


Figure 15.11: *The dot plot showing a low-complexity region in the sequence. The sequence is artificial and low complexity regions do not always show as a square.*

15.3.4 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 15.1: **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 15.1. 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 15.12).

  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 15.12) 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.

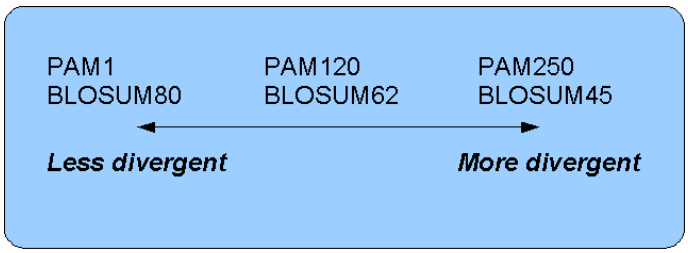


Figure 15.12: *Relationship between scoring matrices. The BLOSUM62 has become a* de facto *standard scoring matrix for a wide range of alignment programs. It is the default matrix in BLAST.*

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

15.4 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:

Toolbox | **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 15.13 shows 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.

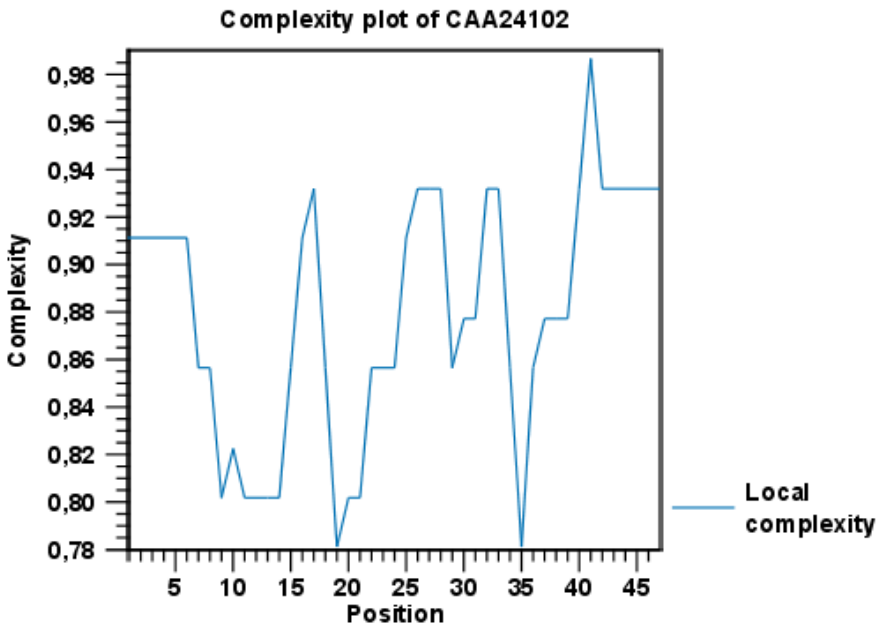


Figure 15.13: *An example of a local complexity plot.*

15.5 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:

Toolbox | **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 15.14), 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.

You can also 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.)

Click **Finish** to start the tool. An example of protein sequence statistics is shown in figure 15.15.

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.

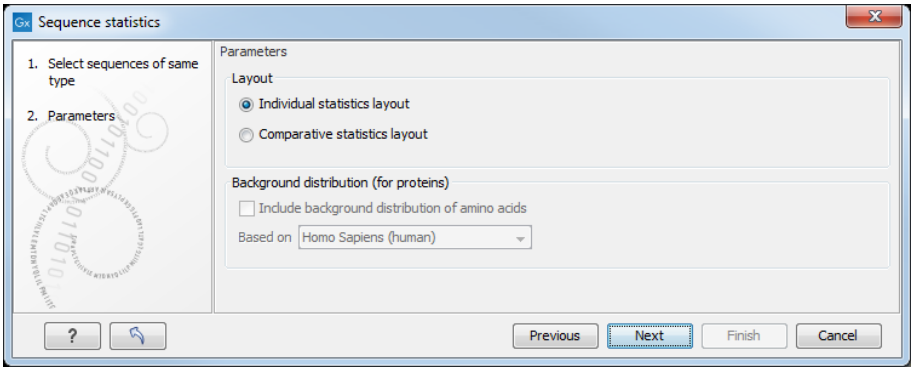


Figure 15.14: *Setting parameters for the Sequence statistics tool.*

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 15.15: *Example of 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**:
  - Sequence type
  - Length
  - Organism
  - Name
  - Description
  - Modification Date
  - Weight. This is calculated like this: $sum_{unitsinsequence}(weight(unit)) - links * weight(H2O)$ 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.

A short description of the different areas of the statistical output is given in section 15.5.1.

15.5.1 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.

  $Aliphaticindex = X(Ala) + a * X(Val) + b * X(Leu) + b * (X)Ile$

  X(Ala), *X(Val)*, *X(Ile)* and *X(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].

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 15.2: **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*.

- **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 15.2). 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 using the following equation:

  $$Ext(Protein) = count(Cystine) * Ext(Cystine) + count(Tyr) * Ext(Tyr) + count(Trp) * Ext(Trp)$$

  where Ext is the extinction coefficient of amino acid in question. At 280nm the extinction coefficients are: Cys=120, Tyr=1280 and Trp=5690. This equation is only valid under the following conditions:

  - pH 6.5
  - 6.0 M guanidium hydrochloride
  - 0.02 M phosphate buffer

The extinction coefficient values of the three important amino acids at different wavelengths are found in [Gill and von Hippel, 1989]. Knowing the extinction coefficient, the absorbance (optical density) can be calculated using the following formula: $Absorbance(Protein) = \frac{Ext(Protein)}{Molecular\ weight}$

Two values are reported. The first value is computed assuming that all cysteine residues appear as half cystines, meaning they form di-sulfide bridges to other cysteines. The second number assumes that no di-sulfide bonds are formed.

- **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.

15.6 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 by:

Toolbox | **Classical Sequence Analysis** ( ) | **General Sequence Analyses** ( ) | **Join Sequences** ( )

This opens the dialog shown in figure 15.16.

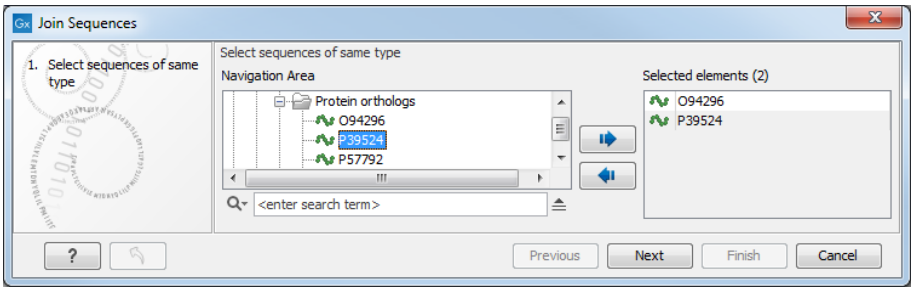


Figure 15.16: *Selecting two sequences to be joined.*

If you have selected some sequences before choosing the Toolbox action, they are now listed in the **Selected Elements** window of the dialog. Use the arrows to add or remove sequences from the selected elements. Click **Next** opens the dialog shown in figure 15.17.

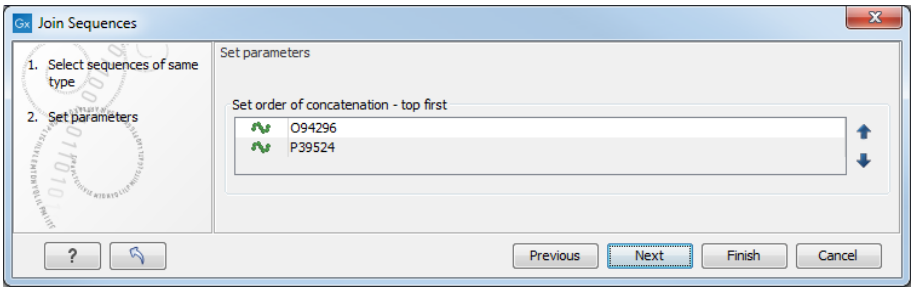


Figure 15.17: *Setting the order in which sequences are joined.*

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 15.18.

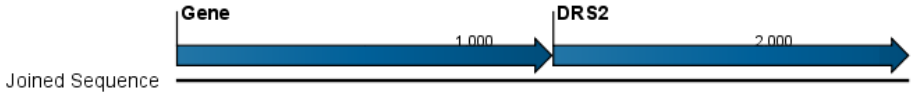


Figure 15.18: *The result of joining sequences is a new sequence containing the annotations of the joined sequences (they each had a HBB annotation).*

15.7 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:

Toolbox | **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 15.19).

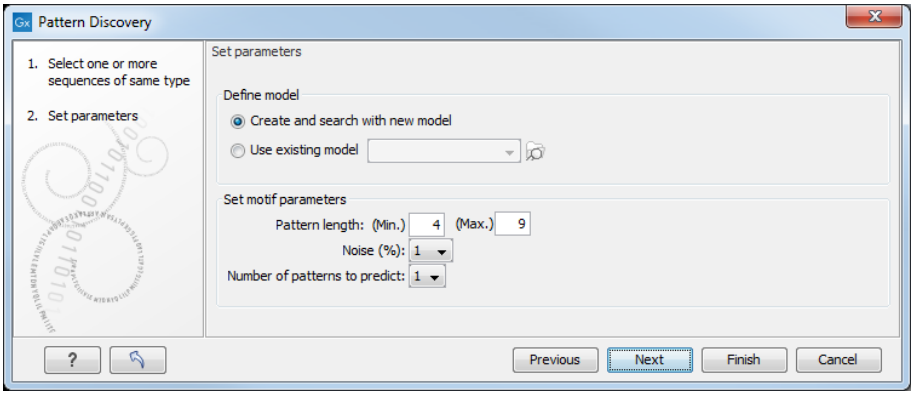


Figure 15.19: *Setting parameters for the pattern discovery. See text for details.*

In order to search unknown sequences with an already existing model:

Select to use an already existing model which is seen in figure 15.19. Models are represented with the following icon in the **Navigation Area** ( ).

15.7.1 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 15.19.

- **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 15.20). If you have selected several sequences, a corresponding number of views will be opened.



Figure 15.20: *Sequence view displaying two discovered patterns.*

15.7.2 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.

15.8 Motif Search

CLC Genomics Workbench offers advanced and versatile options to search for known motifs represented either by a simple sequence or a more advanced regular expression. These advanced search capabilities are available for use in both DNA and protein sequences.

There are two ways to access this functionality:

- When viewing sequences, it is possible to have motifs calculated and shown on the sequence in a similar way as restriction sites (see section 20.1.1). This approach is called *Dynamic motifs* and is an easy way to spot known sequence motifs when working with sequences for cloning etc.
- A more refined and systematic search for motifs can be performed through the **Toolbox**. This will generate a table and optionally add annotations to the sequences.

The two approaches are described below.

15.8.1 Dynamic motifs

In the **Side Panel** of sequence views, there is a group called **Motifs** (see figure 15.21).

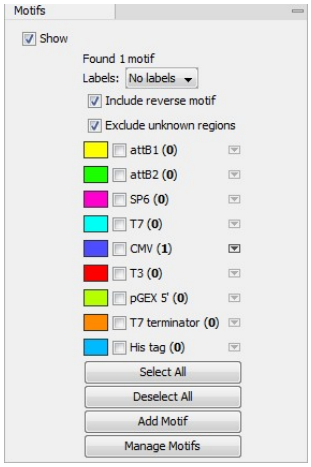


Figure 15.21: *Dynamic motifs in the Side Panel.*

The Workbench will look for the listed motifs in the sequence that is open and by clicking the check box next to the motif it will be shown in the view as illustrated in figure 15.22.

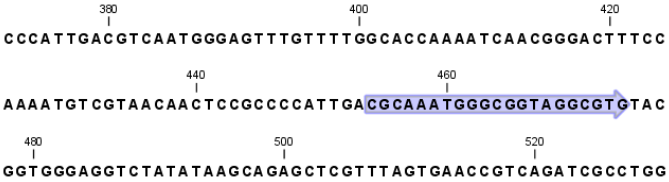


Figure 15.22: *Showing dynamic motifs on the sequence.*

This case shows the CMV promoter primer sequence which is one of the pre-defined motifs in *CLC Genomics Workbench*. The motif is per default shown as a faded arrow with no text. The direction of the arrow indicates the strand of the motif.

Placing the mouse cursor on the arrow will display additional information about the motif as illustrated in figure 15.23.



Figure 15.23: *Showing dynamic motifs on the sequence.*

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.

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 15.21 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 15.9)). 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 15.24.

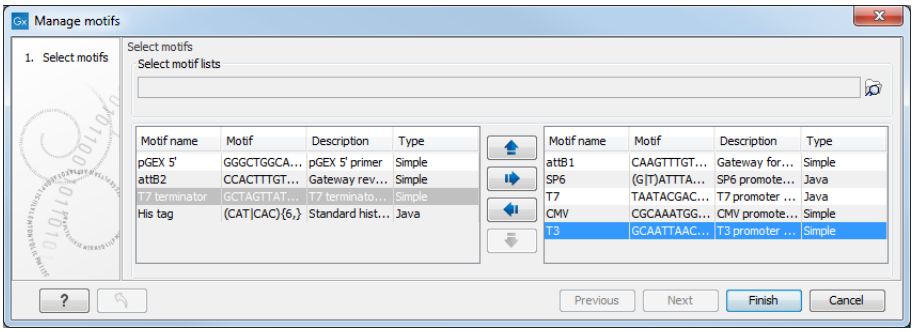


Figure 15.24: *Managing the motifs to be shown.*

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**.

15.8.2 Motif search from the Toolbox

The dynamic motifs described in section 15.8.1 provide a quick way of routinely scanning a sequence for commonly used motifs, but in some cases a more systematic approach is needed. The motif search in the **Toolbox** provides an option to search for motifs with a user-specified similarity to the target sequence, and furthermore the motifs found can be displayed in an overview table. This is particularly useful when searching for motifs on many sequences.

To start the Toolbox motif search, go to:

Toolbox | **Classical Sequence Analysis ( )** | **General Sequence Analysis ( )**| **Motif Search ( )**

A dialog window will be launched. Use the arrows to add or remove sequences or sequence lists between the Navigation Area and the selected elements.

You can perform the analysis on several DNA or several protein sequences at a time. In this case, the method will search for patterns in the sequences and create an overview table of the motifs found in all sequences.

Click **Next** to adjust parameters (see figure 15.25).

The options for the motif search are:

- **Motif types.** Choose what kind of motif to be used:

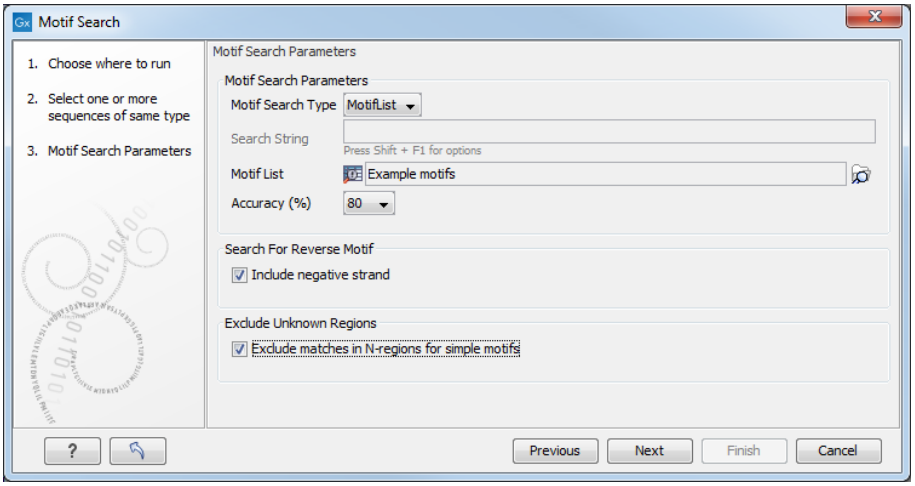


Figure 15.25: *Setting parameters for the motif search.*

  - Simple motif. Choosing this option means that you enter a simple motif, e.g. ATGATGNNATG.
  - Java regular expression. See section 15.8.3.
  - 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 `http://www.expasy.org/cgi-bin/prosite-list.pl`).
  - Use motif list. Clicking the small button ( ) will allow you to select a saved motif list (see section 15.9).

- **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 15.8.3. 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 two types of results that can be produced:

- **Add annotations**. This will add an annotation to the sequence when a motif is found (an example is shown in figure 15.26.
- **Create table**. This will create an overview table of all the motifs found for all the input sequences.

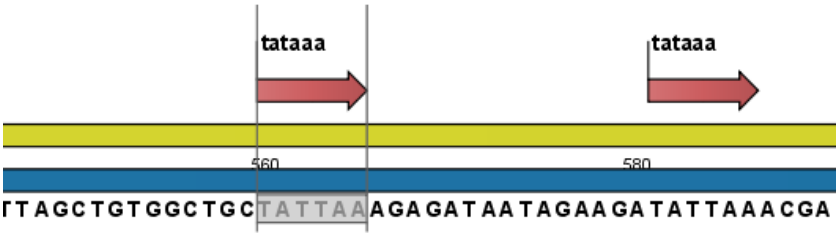


Figure 15.26: *Sequence view displaying the pattern found. The search string was 'tataaa'.*

15.8.3 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 `http://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*.

15.9 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:

Toolbox | **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 15.27.

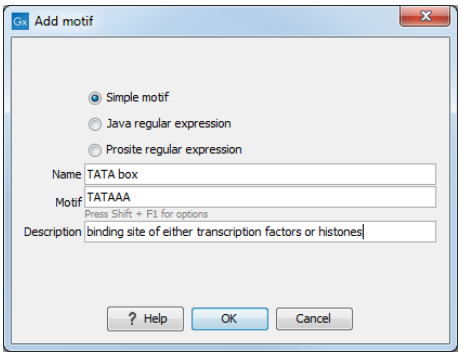


Figure 15.27: *Entering a new motif in the list.*

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 15.8.2 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 15.8.2.

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 15.8).

Chapter 16

Nucleotide analyses

Contents

16.1 Convert DNA to RNA 368
16.2 Convert RNA to DNA 369
16.3 Reverse complements of sequences 369
16.4 Translation of DNA or RNA to protein 370
16.5 Find open reading frames 372
16.5.1 Open reading frame parameters 373

CLC Genomics Workbench offers different kinds of sequence analyses, which only apply to DNA and RNA.

16.1 Convert DNA to RNA

CLC Genomics Workbench lets you convert a DNA sequence into RNA, substituting the T residues (Thymine) for U residues (Uracil):

Toolbox | **Classical Sequence Analysis** ( ) | **Nucleotide Analysis** ( )| **Convert DNA to RNA** ( )

This opens the dialog displayed in figure 16.1:

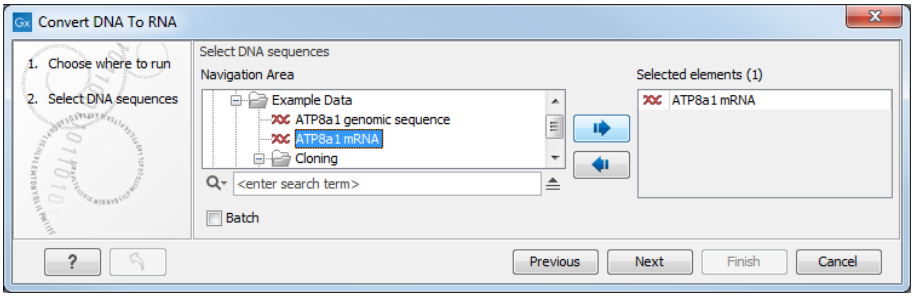


Figure 16.1: *Translating DNA to RNA.*

If a sequence was selected before choosing the Toolbox action, 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 **Finish** to start the tool.

Note! You can select multiple DNA sequences and sequence lists at a time. If the sequence list contains RNA sequences as well, they will not be converted.

16.2 Convert RNA to DNA

CLC Genomics Workbench lets you convert an RNA sequence into DNA, substituting the U residues (Uracil) for T residues (Thymine):

Toolbox | **Classical Sequence Analysis** ( ) | **Nucleotide Analysis** ( )| **Convert RNA to DNA** ( )

This opens the dialog displayed in figure 16.2:

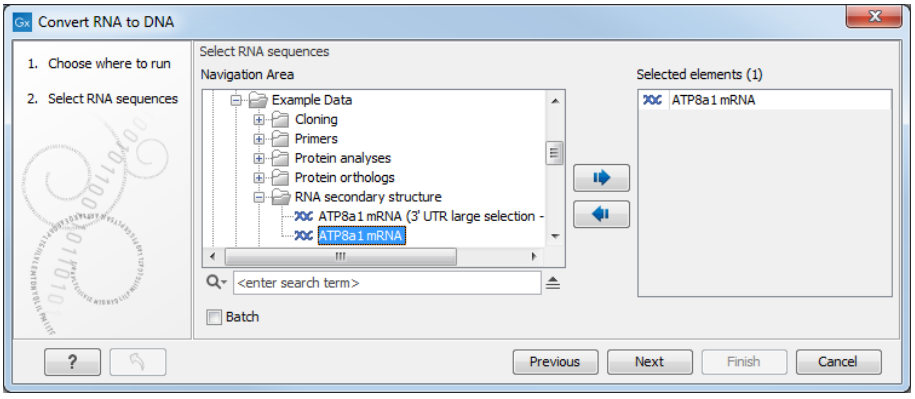


Figure 16.2: *Translating RNA to DNA.*

If a sequence was selected before choosing the Toolbox action, 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 **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.

Note! You can select multiple RNA sequences and sequence lists at a time. If the sequence list contains DNA sequences as well, they will not be converted.

16.3 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:

Toolbox | **Classical Sequence Analysis** ( ) | **Nucleotide Analysis** ( )| **Reverse Complement Sequence** ( )

This opens the dialog displayed in figure 16.3:

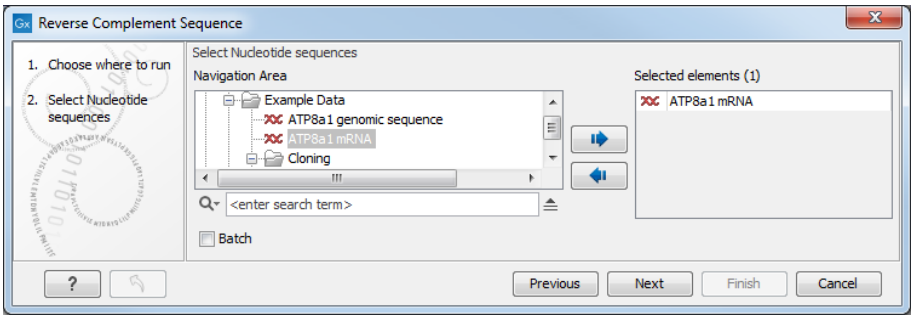


Figure 16.3: *Creating a reverse complement sequence.*

If a sequence was selected before choosing the Toolbox action, the 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 **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.

16.4 Translation of DNA or RNA to protein

In *CLC Genomics Workbench* you can translate a nucleotide sequence into a protein sequence using the **Toolbox** tools. Usually, you use the +1 reading frame which means that the translation starts from the first nucleotide. Stop codons result in an asterisk being inserted in the protein sequence at the corresponding position. It is possible to translate in any combination of the six reading frames in one analysis. To translate, go to:

Toolbox | **Classical Sequence Analysis** ( ) | **Nucleotide Analysis** ( )| **Translate to Protein** ( )

This opens the dialog displayed in figure 16.4:

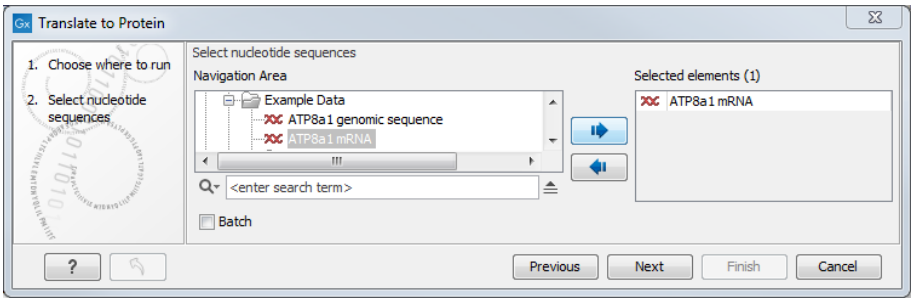


Figure 16.4: *Choosing sequences for translation.*

If a sequence was selected before choosing the Toolbox action, the 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.

Clicking **Next** generates the dialog seen in figure 16.5:

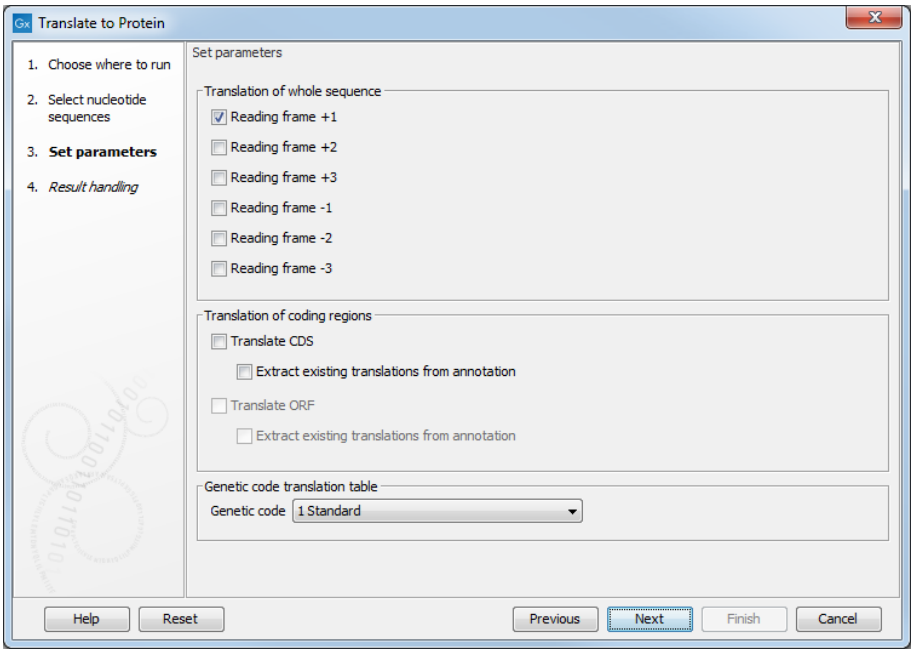


Figure 16.5: *Choosing translation of CDSs using standard translation table.*

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 "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 translation table Lets you specify the genetic code for the translation. The translation tables are occasionally updated from NCBI. The tables are not available in this printable version of the user manual. Instead, the tables are included in the **Help**-menu in the **Menu Bar** (in the appendix).

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 16.5).

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 16.6): either a specific genetic code translation table, or to extract the existing translation from annotation (if the annotation contains information about the translation). Choose the option needed and click **Translate**.

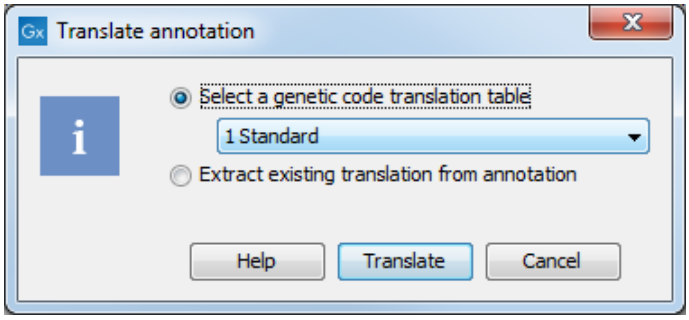


Figure 16.6: *Choosing how to translate CDS or ORF annotations.*

The CDS and ORF annotations are colored yellow as default.

16.5 Find open reading frames

The *CLC Genomics Workbench* **Find Open Reading Frames** function can be used to find all open reading frames (ORF) in a sequence, or, by choosing particular start codons to use, it can be used as a rudimentary gene finder. ORFs identified will be shown as annotations on the sequence. You have the option of choosing a translation table, the start codons to use, minimum ORF length as well as a few other parameters. These choices are explained in this section.

To find open reading frames:

Toolbox | Classical Sequence Analysis ( ) | Nucleotide Analysis ( )| Find Open Reading Frames ( )

This opens the dialog displayed in figure 16.7:

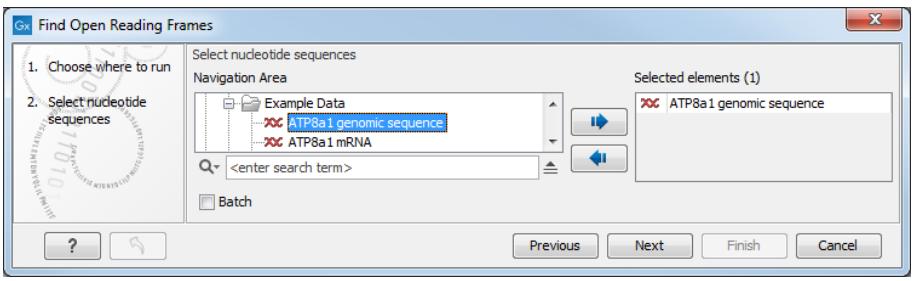


Figure 16.7: *Create Reading Frame dialog.*

If a sequence was selected before choosing the Toolbox action, the 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.

The **Find Open Reading Frames** tool simply looks for start and stop codons and reports any open reading frames that satisfy the parameters. If you want to adjust the parameters for finding open reading frames click **Next**.

16.5.1 Open reading frame parameters

This opens the dialog displayed in figure 16.8:

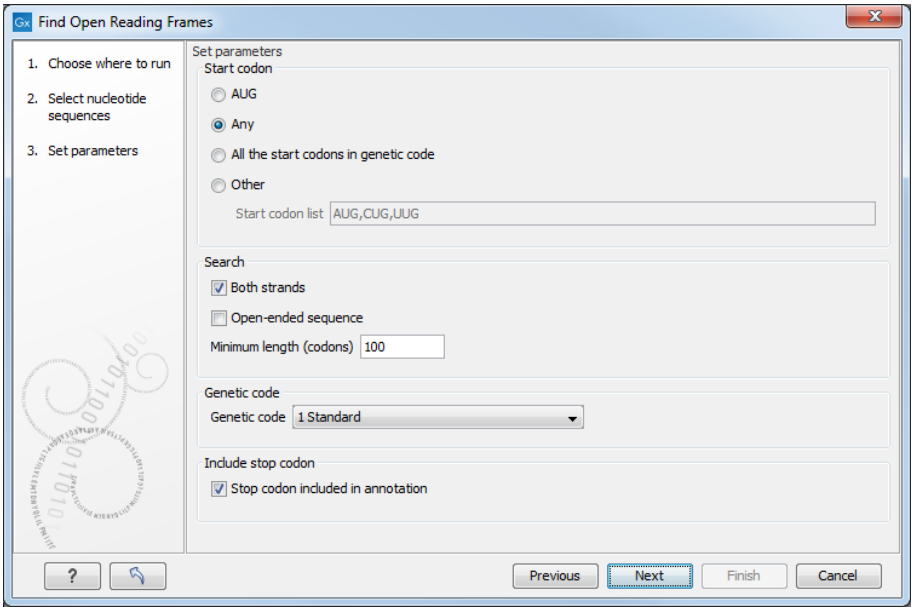


Figure 16.8: *Parameters for the Reading Frame tool.*

The adjustable parameters for the search are:

- **Start codon**:
  - **AUG**. Most commonly used start codon.
  - **Any**. Find all open reading frames of specified length. Any combination of three bases that is not a stop-codon is interpreted as a start codon, and translated according to the specified genetic code.
  - **All start codons in genetic code**.
  - **Other**. Here you can specify a number of start codons separated by commas.
- **Both strands**. Finds reading frames on both strands.
- **Open-ended Sequence**. Allows the ORF to start or end outside the sequence. If the sequence studied is a part of a larger sequence, it may be advantageous to allow the ORF to start or end outside the sequence.
- **Genetic code translation table**.

- **Include stop codon in result** The ORFs will be shown as annotations which can include the stop codon if this option is checked. The translation tables are occasionally updated from NCBI. The tables are not available in this printable version of the user manual. Instead, the tables are included in the **Help** menu in the **Menu Bar** (in the appendix).
- **Minimum Length**. Specifies the minimum length for the ORFs to be found. The length is specified as number of codons.

Using open reading frames for gene finding is a fairly simple approach which is likely to predict genes which are not real. Setting a relatively high minimum length of the ORFs will reduce the number of false positive predictions, but at the same time short genes may be missed (see figure 16.9).

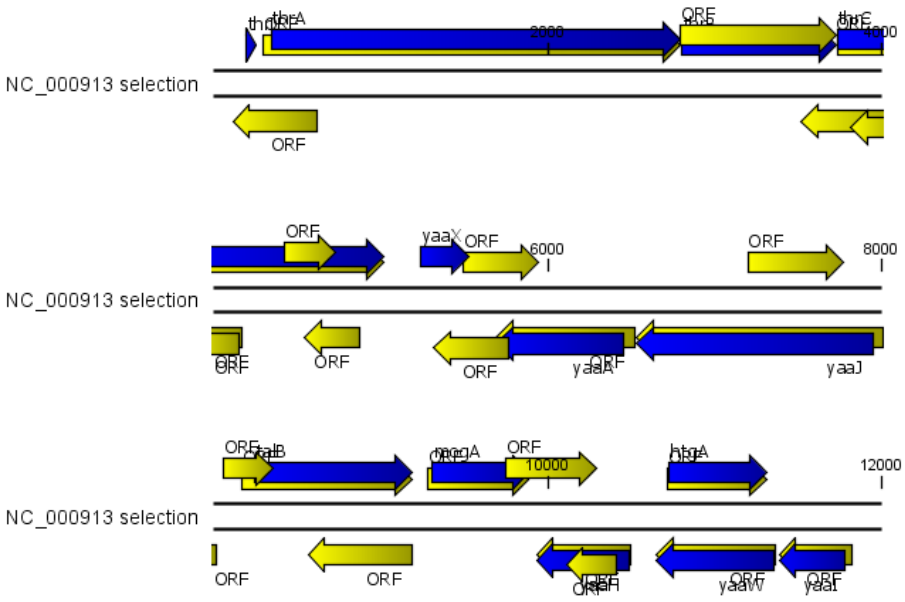


Figure 16.9: *The first 12,000 positions of the* E. coli *sequence NC_000913 downloaded from GenBank. The blue (dark) annotations are the genes while the yellow (brighter) annotations are the ORFs with a length of at least 100 amino acids. On the positive strand around position 11,000, a gene starts before the ORF. This is due to the use of the standard genetic code rather than the bacterial code. This particular gene starts with CTG, which is a start codon in bacteria. Two short genes are entirely missing, while a handful of open reading frames do not correspond to any of the annotated genes.*

Click **Finish** to start the tool.

Finding open reading frames is often a good first step in annotating sequences such as cloning vectors or bacterial genomes. For eukaryotic genes, ORF determination may not always be very helpful since the intron/exon structure is not part of the algorithm.

Chapter 17

Protein analyses

Contents

17.1 Protein charge . . . 375
17.2 Antigenicity . . . 377
17.3 Hydrophobicity . . . 378
17.3.1 Hydrophobicity graphs along sequence . . . 379
17.3.2 Bioinformatics explained: Protein hydrophobicity . . . 381
17.4 Download Pfam Database . . . 383
17.5 Pfam domain search . . . 383
17.6 Find and Model Structure . . . 385
17.6.1 Create structure model . . . 387
17.6.2 Model structure . . . 388
17.7 Secondary structure prediction . . . 393
17.8 Protein report . . . 395
17.9 Reverse translation from protein into DNA . . . 397
17.9.1 Bioinformatics explained: Reverse translation . . . 399
17.10 Proteolytic cleavage detection . . . 400
17.10.1 Bioinformatics explained: Proteolytic cleavage . . . 402

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 http://resources.qiagenbioinformatics.com/manuals/signalP/current/SignalP_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.

17.1 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:

> **Toolbox** | **Classical Sequence Analysis ()** | **Protein Analysis ()**| **Create Protein Charge Plot ()**

This opens the dialog displayed in figure 17.1:

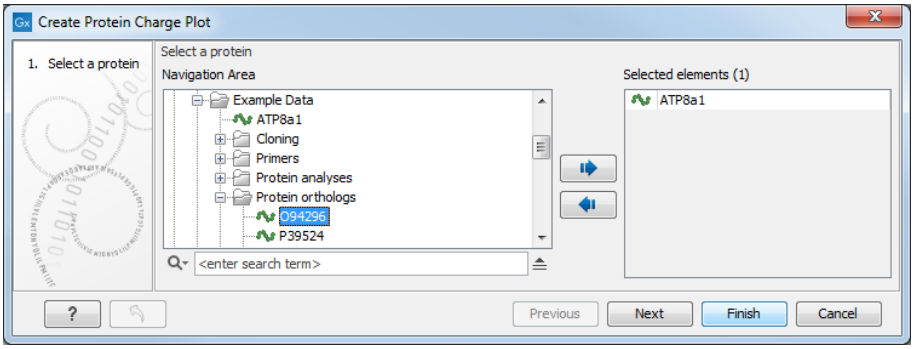


Figure 17.1: *Choosing protein sequences to calculate protein charge.*

If a sequence was selected before choosing the Toolbox action, the 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.

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 17.2 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.

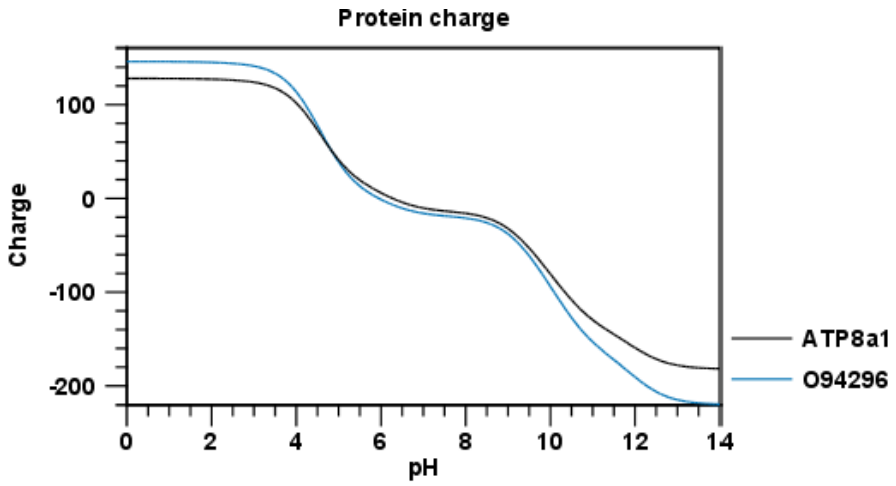


Figure 17.2: *View of the protein charge.*

17.2 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:

> **Toolbox** | **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 Toolbox action, these are shown in the **Selected Elements**. Clicking **Next** takes you through to **Step 2**, which is displayed in figure 17.3.

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 chose from a number of antigenicity scales. Click **Finish** to start the tool. The result can be seen in figure 17.4.

See section B in the appendix for information about the graph view.

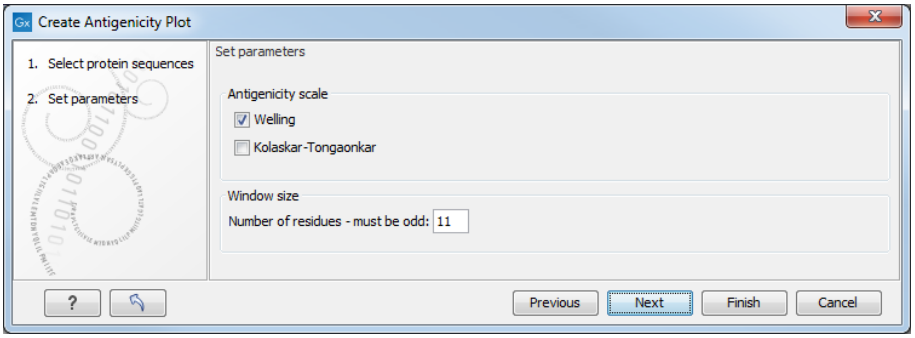


Figure 17.3: *Step two in the Antigenicity Plot allows you to choose different antigenicity scales and the window size.*

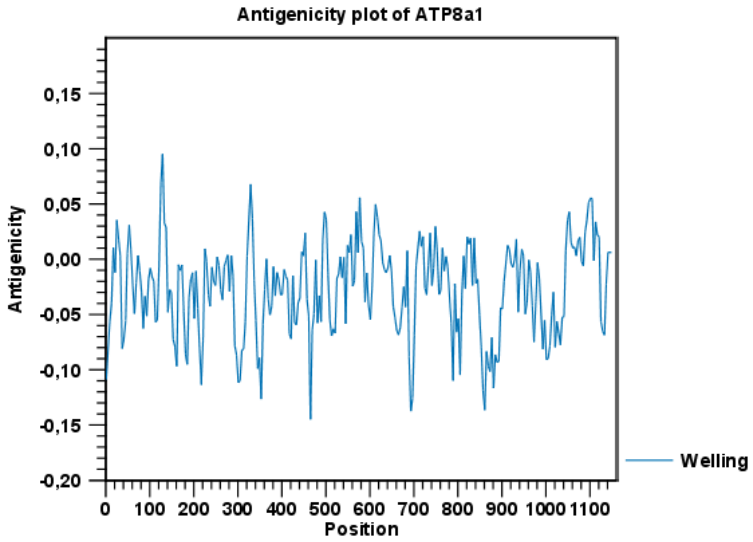


Figure 17.4: *The result of the antigenicity plot calculation and the associated Side Panel.*

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 17.3.1).

17.3 Hydrophobicity

CLC Genomics Workbench can calculate the hydrophobicity of protein sequences in different ways, using different algorithms (see section 17.3.2). 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:

Toolbox | **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 17.5.

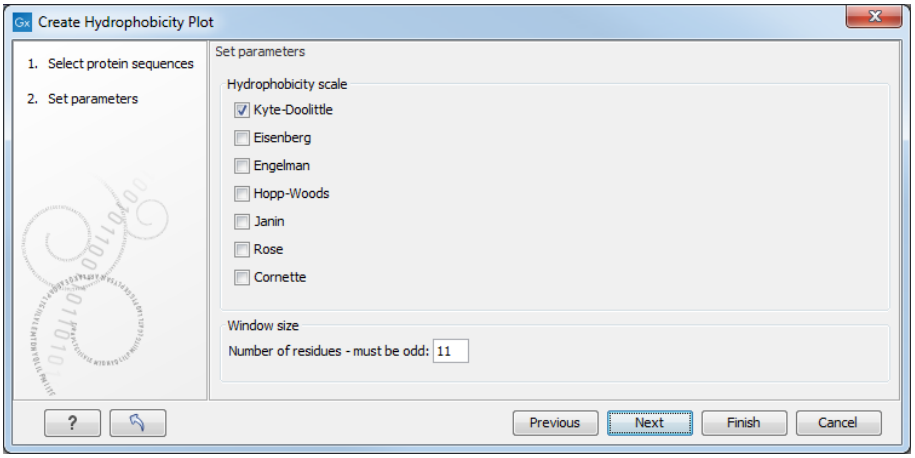


Figure 17.5: *Step two in the Hydrophobicity Plot allows you to choose hydrophobicity scale and the window size.*

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 chose from a number of hydrophobicity scales which are further explained in section 17.3.2 Click **Finish** to start the tool. The result can be seen in figure 17.6.

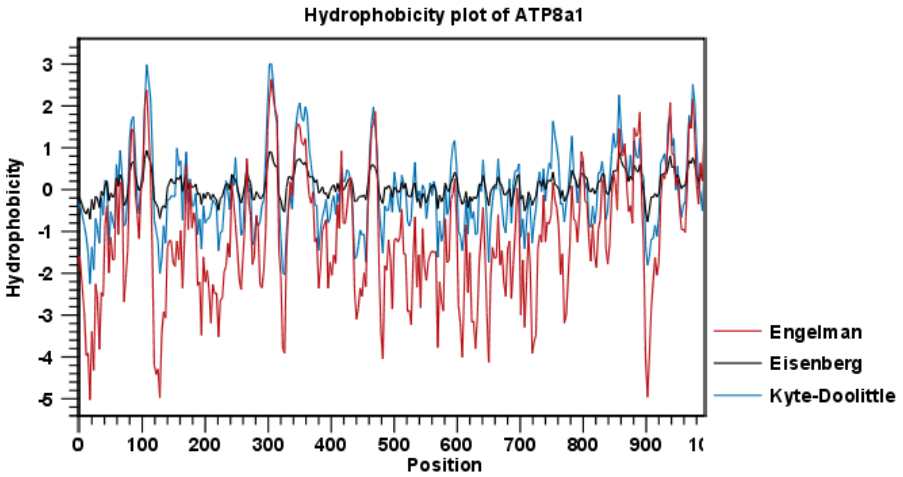


Figure 17.6: *The result of the hydrophobicity plot calculation and the associated Side Panel.*

See section B in the appendix for information about the graph view.

17.3.1 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 17.7.



Figure 17.7: *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 17.3.2).

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 17.8).

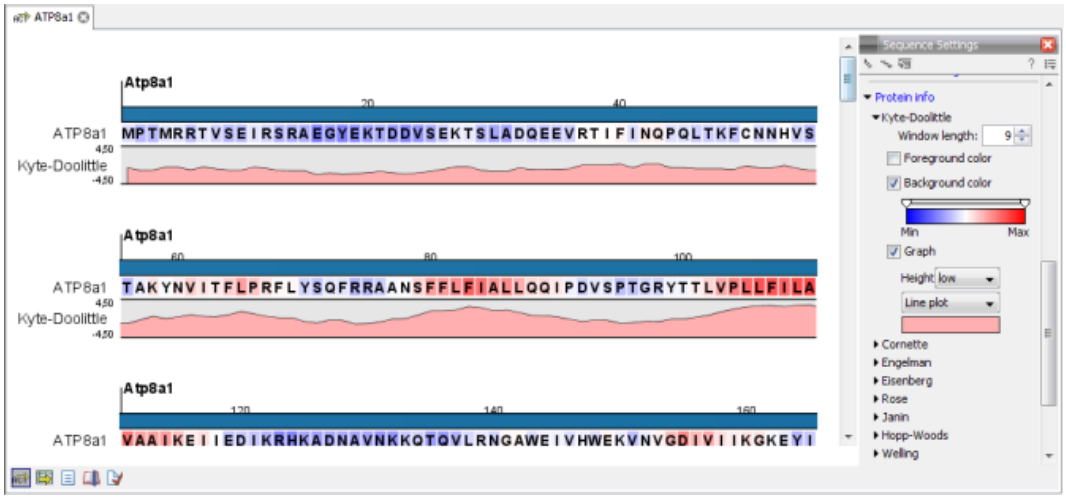


Figure 17.8: *The different ways of displaying the hydrophobicity scores, using the Kyte-Doolittle scale.*

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 17.8. Notice that you can choose the height of the graphs underneath the sequence.

17.3.2 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.

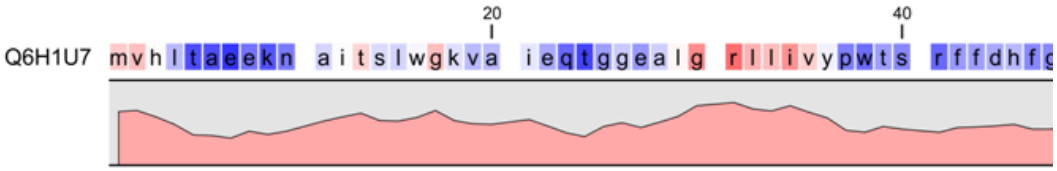


Figure 17.9: *Plot of hydrophobicity along the amino acid sequence. Hydrophobic regions on the sequence have higher numbers according to the graph below the sequence, furthermore hydrophobic regions are colored on the sequence. Red indicates regions with high hydrophobicity and blue indicates regions 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 17.9).

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 17.1: *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

17.4 Download Pfam Database

To be able to run the **Pfam Domain Search** tool you must first download the Pfam database. The Pfam database can be downloaded using:

Toolbox | **Classical Sequence Analysis** () | **Protein Analysis** ()| **Download Pfam Database** ()

Specify where you would like to save the downloaded Pfam database. The output of the **Download Pfam Database** tool is a database object, which can be selected as a parameter for the Pfam Domain Search tool. It doesn't really make sense to try to open the database object directly from the **Navigation Area** as all you can see directly is the element history (which version of the Workbench that has been used and the name of the downloaded files) and the element info, which in this case only provides information about the database name.

17.5 Pfam domain search

With *CLC Genomics Workbench* you can perform a search for domains in protein sequences using the Pfam database. The Pfam database [Bateman et al., 2004] at http://pfam.sanger.ac.uk/ was initially developed to aid the annotation of the *C. elegans* genome. The database is a large collection of multiple sequence alignments, and contains profile hidden Markov models (HMMs) for individual domain alignments, which can be used to quickly identify domains in

protein sequences.

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.

Using the **Pfam Domain Search** tool in *CLC Genomics Workbench*, you can search for domains in sequence data which otherwise do not carry any annotation information. The domain search is performed using the hmmsearch tool from the HMMER3 package version 3.1b1 (`http://hmmer.janelia.org/`). The Pfam search tool annotates protein sequences with all domains in the Pfam database that have a significant match. It is possible to lower the significance cutoff thresholds in the hmmsearch algorithm, which will reduce the number of domain annotations. Individual domain annotations can be removed manually as described in section 12.3.4.

When you have downloaded the Pfam database you are ready to perform a Pfam domain search. To do this start the Pfam search tool:

> **Toolbox** | **Classical Sequence Analysis** ( ) | **Protein Analysis** ( )| **Pfam Domain Search** ( )

If a sequence was selected before choosing the Toolbox action, 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.

You can perform the analysis on several protein sequences at a time. This will add annotations to all the sequences. Click **Next** to adjust parameters (see figure 17.10).

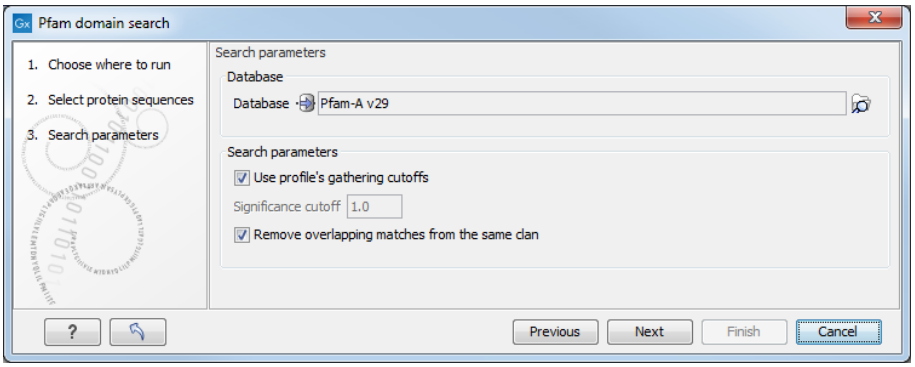


Figure 17.10: *Setting parameters for Pfam Domain Search.*

- **Database.** Choose which database to use when searching for Pfam domains. For information on how to download a Pfam database see section 17.4
- 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 compared to 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.

Click **Next** to adjust the output of the tool. The Pfam search tool can produce two types of output. It can add annotations on the input sequences that show the domains found (see figure 17.11) and it can output a table with all the domains found.

Click **Finish** to start the tool.

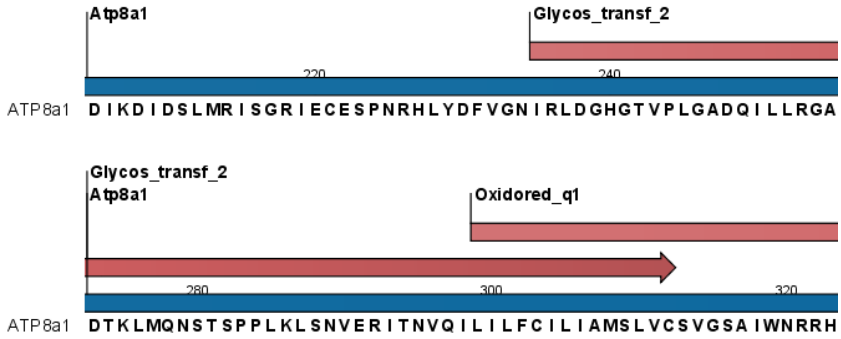


Figure 17.11: *Annotations (in red) that were added by the Pfam search tool.*

Domain annotations added by the Pfam search tool have the type **Region**. If the annotations are not visible they have to be enabled in the side panel. Detailed information for each domain annotation, such as the bit score which is the basis for the prediction of domains, is available through the annotation tool tip.

A more detailed description of the scores provided in the annotation tool tips can be found here: http://pfam.sanger.ac.uk/help#tabview=tab5.

17.6 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:

Toolbox | **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 29.5.4).

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.

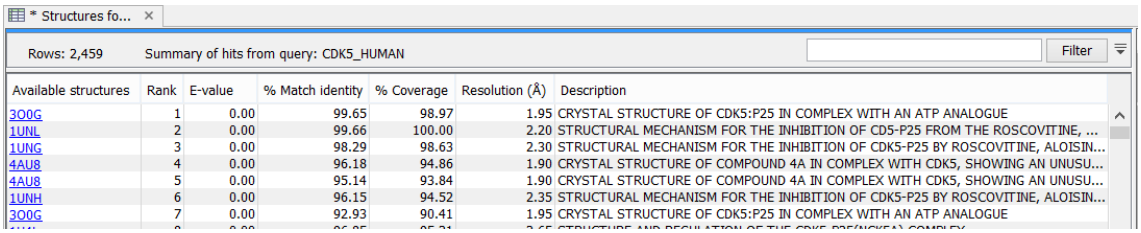
The Find and Model Structure tool carries out the following steps, to find and rank available structures representing the query sequence:

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 17.12), 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 17.6.1. 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...
4AU8	4	0.00	96.18	94.86	1.90	CRYSTAL STRUCTURE OF COMPOUND 4A IN COMPLEX WITH CDK5, SHOWING AN UNUSU...
4AU8	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 17.12: *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 29.5.4).

BLAST hits with E-value > 0.0001 are rejected and a maximum of 2500 BLAST hits are retrieved. Read more about BLAST in section 13.5.

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 17.12). Read more about the *Template quality score* in section 17.6.2.

17.6.1 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 14.4).

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 14.6). 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 17.6.2). 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 17.13), to indicate local model uncertainty (see section 17.6.2). 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 14.4).

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).

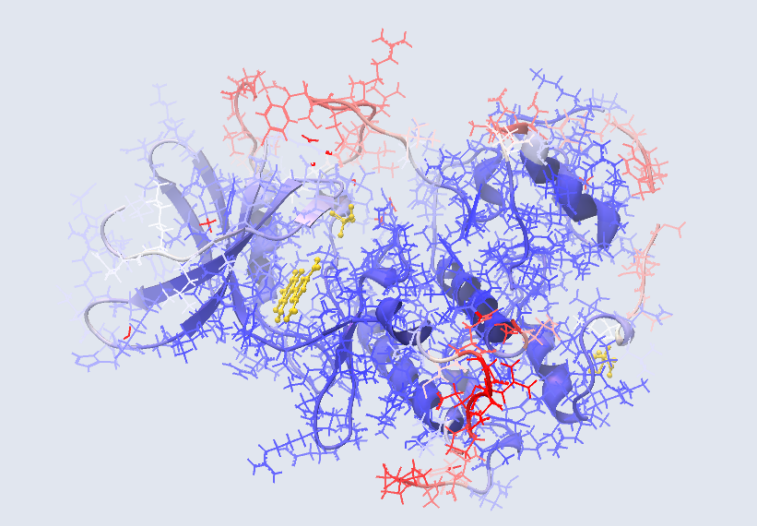


Figure 17.13: *Structure Model of CDK5_HUMAN. The atoms and backbone are colored by temperature, showing uncertain structure in red and well defined structure in blue.*

17.6.2 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 $Å^2$, and is typically in the range [0, 100].

The temperature color scale ranges from blue (0) over white (50) to red (100) (see section 14.3.1).

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 17.14 and 17.15.

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 17.16.

Note: For NMR structures, the temperature factor is set to zero in the PDB file, and the "Color by

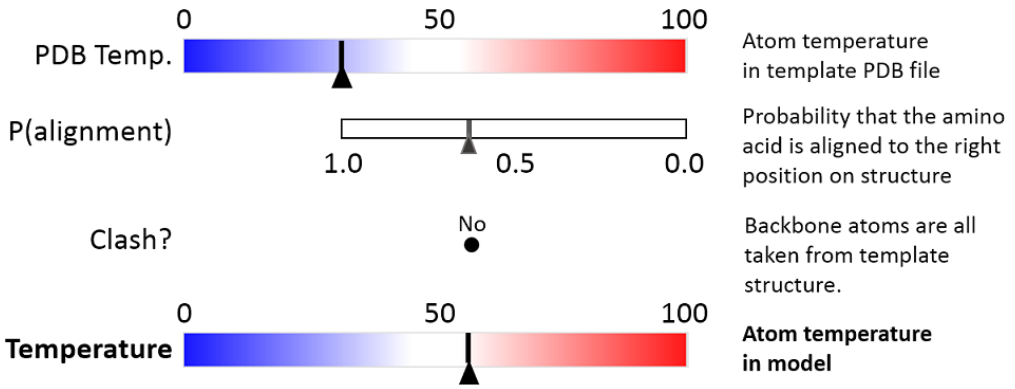


Figure 17.14: *Evaluation of temperature color for backbone atoms in structure models.*

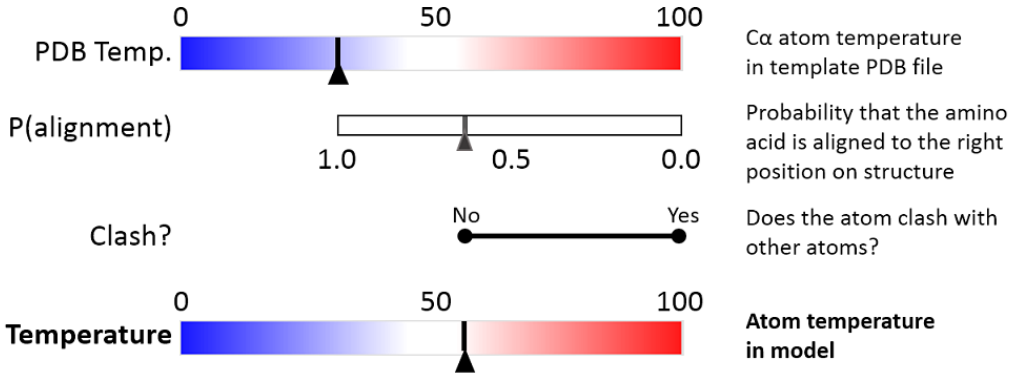


Figure 17.15: *Evaluation of temperature color for side chain atoms in structure models.*

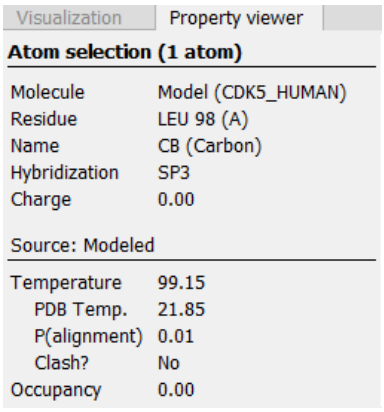


Figure 17.16: *Information displayed in the Side Panel Property viewer for a modeled atom.*

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.

The estimate of the probability that two amino acids are correctly aligned, P(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 29.5.4).

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)

Each of the five descriptors are scaled to [0,1], based on the linear functions seen in figure 17.17. 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 13.5.

% Match identity is the identity between the query sequence and the BLAST hit in the matched region. It is evaluated as

$$\text{\% Match identity} = 100\% \cdot (\text{Identity in BLAST alignment})/L_{\text{B}}$$

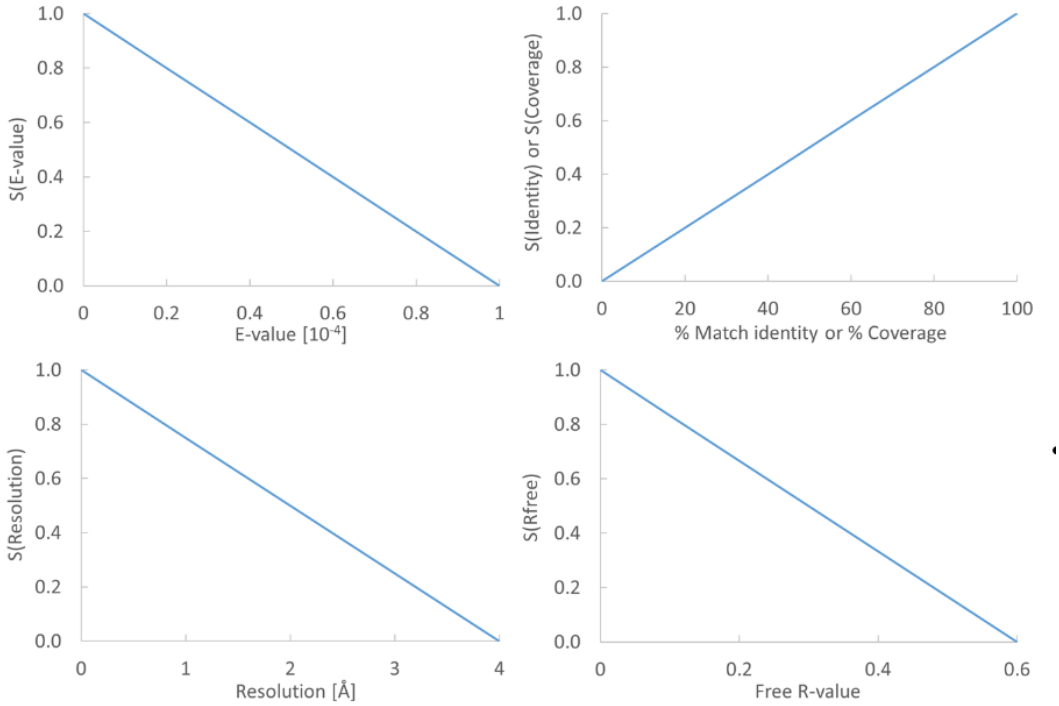


Figure 17.17: *From the E-value, % Match identity, % Coverage, Resolution, and Free R-value, the contributions to the "Template quality score" are determined from the linear functions shown in the graphs.*

where L_B is the length of the BLAST alignment of the matched region, as indicated in figure 17.18, 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 17.18). 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.

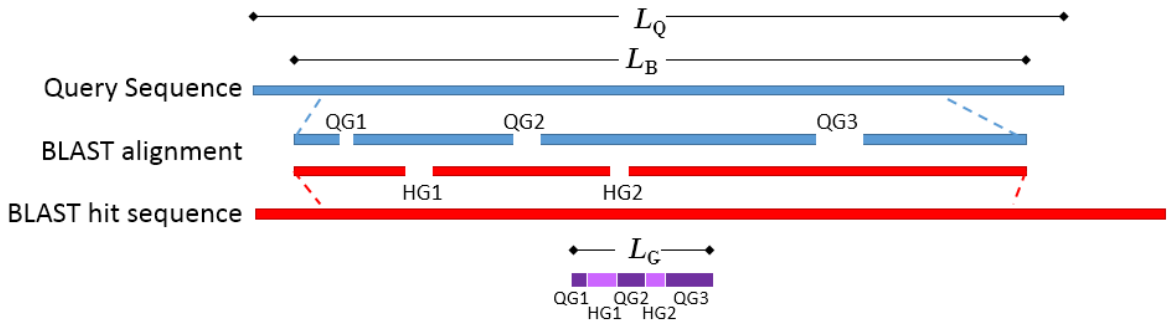


Figure 17.18: *Schematic of a query sequence matched to a BLAST hit. L_Q is the length of the query sequence, L_B is the length of the BLAST alignment of the matched region, QG1-3 are gaps in the matched region of the query sequence, HG1-2 are gaps in the matched region of the BLAST hit sequence, L_G 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 17.19):

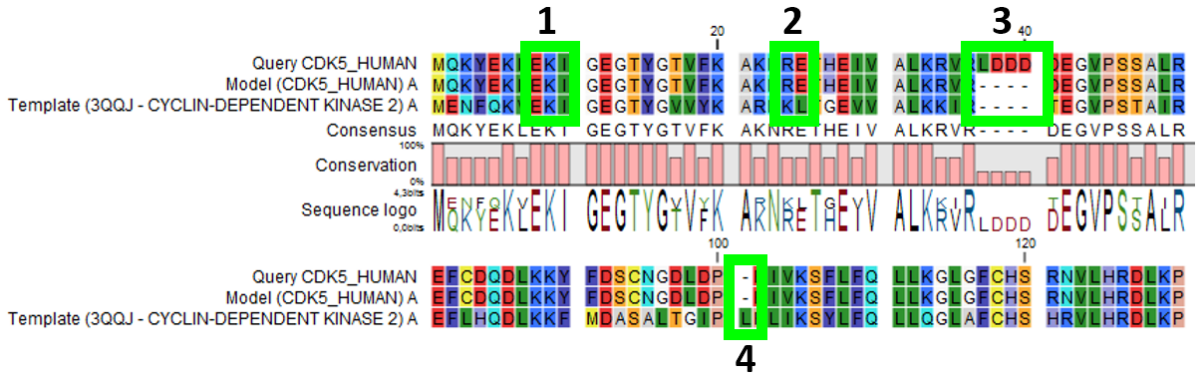


Figure 17.19: *Sequence alignment mapping query sequence (Query CDK5_HUMAN) to the structure with sequence "Template(3QQJ - CYCLIN-DEPENDENT KINASE 2)", producing a structure with sequence "Model(CDK5_HUMAN)". Examples are highlighted: 1. Identical amino acids, 2. Amino acid changes, 3. Amino acids in 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 17.19) => Copy atom positions from the PDB file. If the side chain is missing atoms in the PDB file, the side chain is rebuilt (section 17.6.2).
- For amino acid changes (example 2 in figure 17.19) => Copy backbone atom positions from the PDB file. Model side chain atom positions to match the query sequence (section 17.6.2).
- For amino acids in the query sequence not aligned to a position on the template structure (example 3 in figure 17.19) => No atoms are modeled. The model backbone will have a gap at this position and a "Structure modeling" issue is raised (see section 14.1.4).
- For amino acids on the template structure, not aligned to the query sequence (example 4 in figure 17.19) => The residues are deleted from the structure and a "Structure modeling" issue is raised (see section 14.1.4).

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.

17.7 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 databank (http://www.rcsb.org/pdb/) 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:

Toolbox | **Classical Sequence Analysis** ( ) | **Protein Analysis** ( )| **Predict secondary structure** ( )

This opens the dialog displayed in figure 17.20:

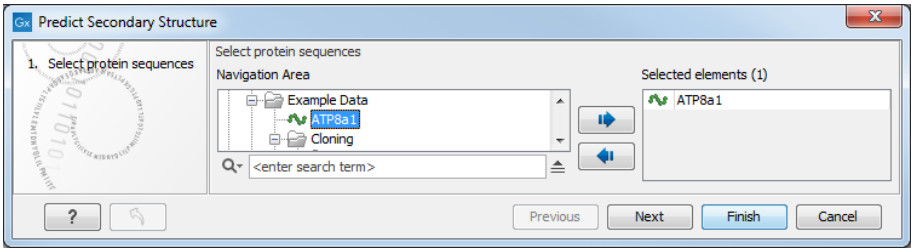


Figure 17.20: *Choosing one or more protein sequences for secondary structure prediction.*

If a sequence was selected before choosing the Toolbox action, 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.

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 17.21).

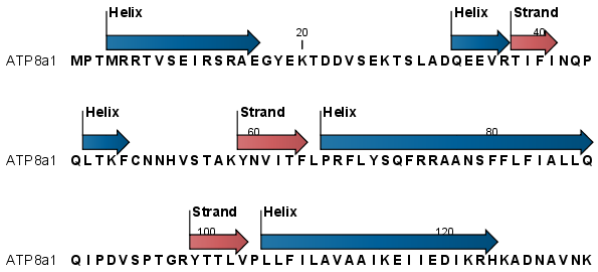


Figure 17.21: *Alpha-helices and beta-strands shown as annotations on the sequence.*

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 12.3.2.

Undesired alpha-helices or beta-sheets can be removed through the **Delete Annotation** () right-click mouse menu. See section 12.3.4.

17.8 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:

> **Toolbox** | **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 Toolbox action, this will be shown in the **Selected Elements**. However, you can use the arrows to change this. When the correct one is chosen, click **Next**.

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 15.5.1.
- **Protein charge plot.** Plot of charge as function of pH, see section 17.1.
- **Hydrophobicity plot.** See section 17.3.
- **Complexity plot.** See section 15.4.
- **Dot plot.** See section 15.3.
- **Secondary structure prediction.** See section 17.7.
- **Pfam domain search.** See section 17.5.
- **BLAST against NCBI databases.** See section 13.1.1.

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 13.1.1.

An example of Protein report can be seen in figure 17.22.

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.

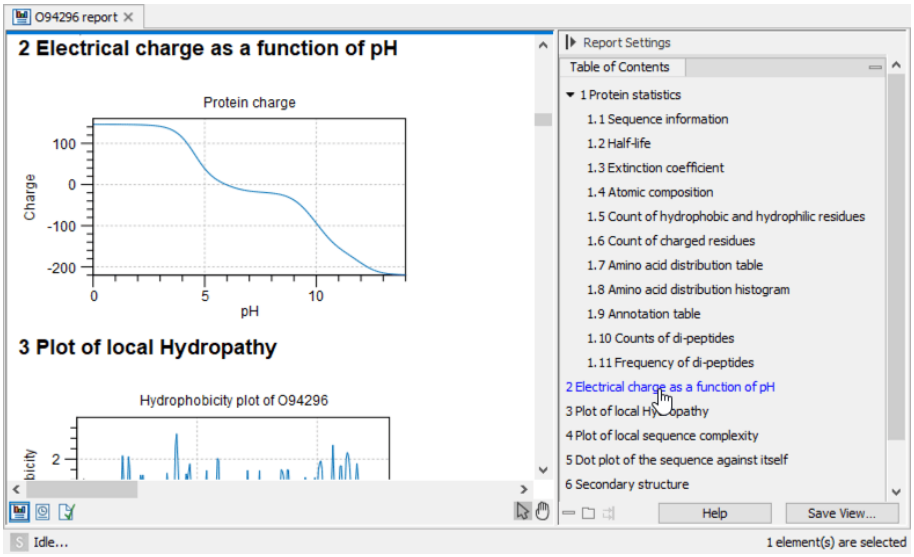


Figure 17.22: *A protein report. There is a Table of Contents in the Side Panel that makes it easy to browse the report.*

17.9 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 17.9.1.

In order to make a reverse translation:

> **Toolbox** | **Classical Sequence Analysis ()** | **Protein Analysis ()**| **Reverse Translate ()**

This opens the dialog displayed in figure 17.23:

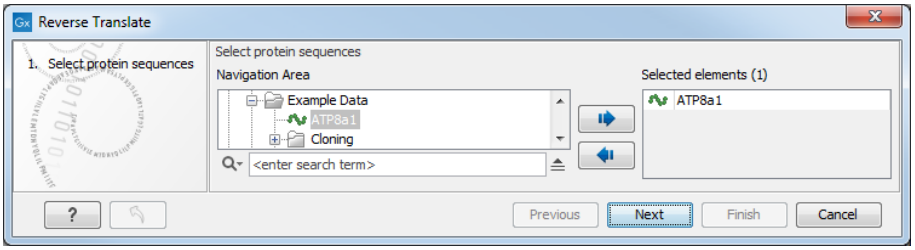


Figure 17.23: *Choosing a protein sequence for reverse translation.*

If a sequence was selected before choosing the Toolbox action, the 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. You can translate several protein sequences at a time.

Adjust the parameters for the translation in the dialog shown in figure 17.24.

- **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

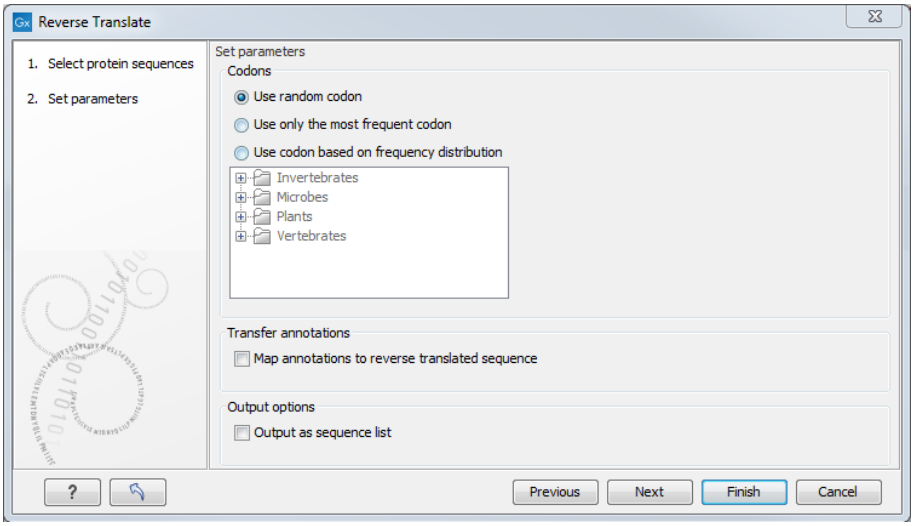


Figure 17.24: *Choosing parameters for the reverse translation.*

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 `http://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.

17.9.1 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 21'st 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

17.10 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:

Toolbox | **Classical Sequence Analysis** ( ) | **Protein Analysis** ( )| **Proteolytic Cleavage** ( )

This opens the dialog shown in figure 17.25. You can select one or several sequences.

In the second dialog, you can select proteolytic cleavage enzymes. Presently, the list contains the enzymes shown in figure 17.26. The full list of enzymes and their cleavage patterns can be seen in Appendix, section D.

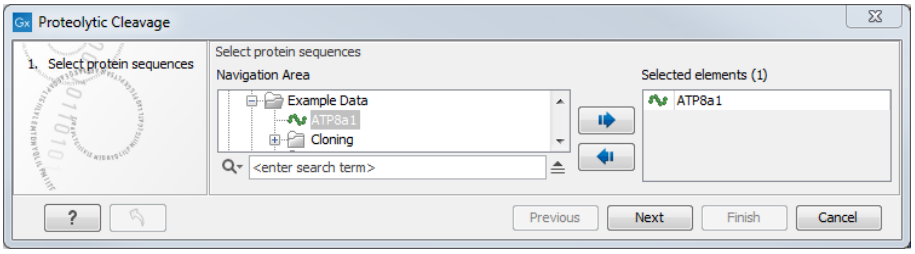


Figure 17.25: *Choosing a protein sequence for proteolytic cleavage.*

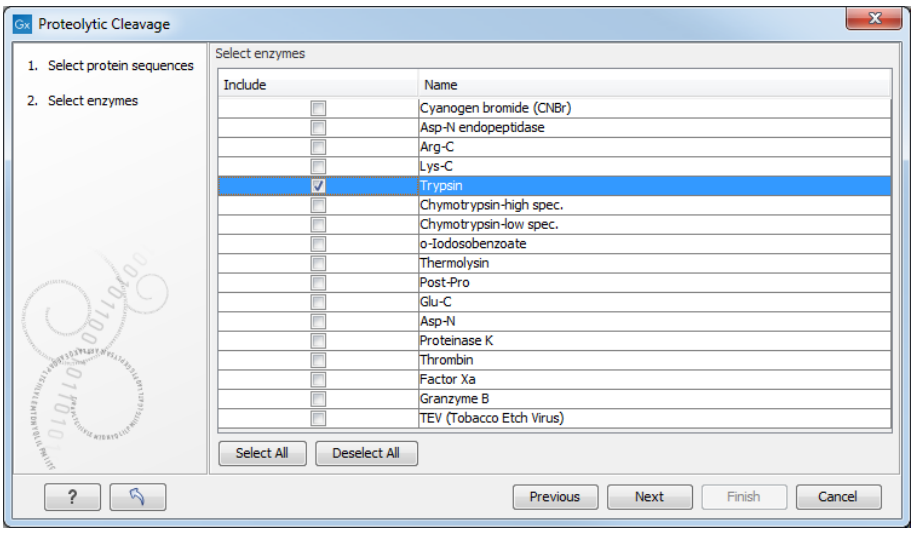


Figure 17.26: *Setting parameters for proteolytic cleavage detection.*

You can then set parameters for the detection. This limits the number of detected cleavages (figure 17.27).

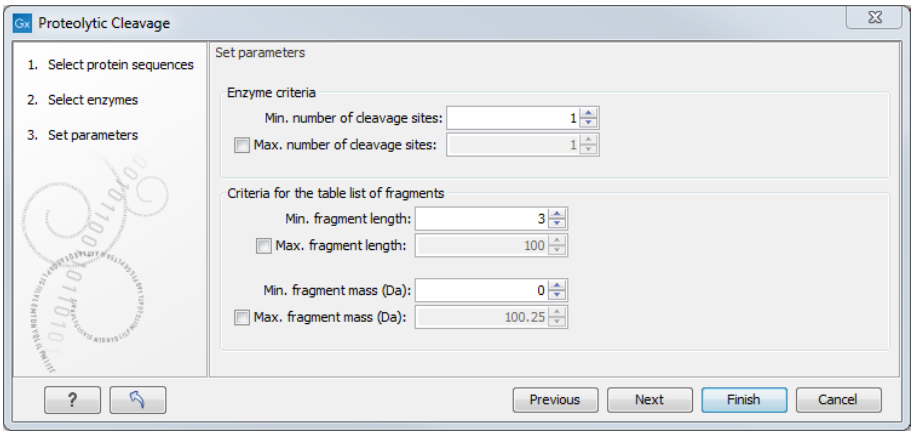


Figure 17.27: *Setting parameters for proteolytic cleavage detection.*

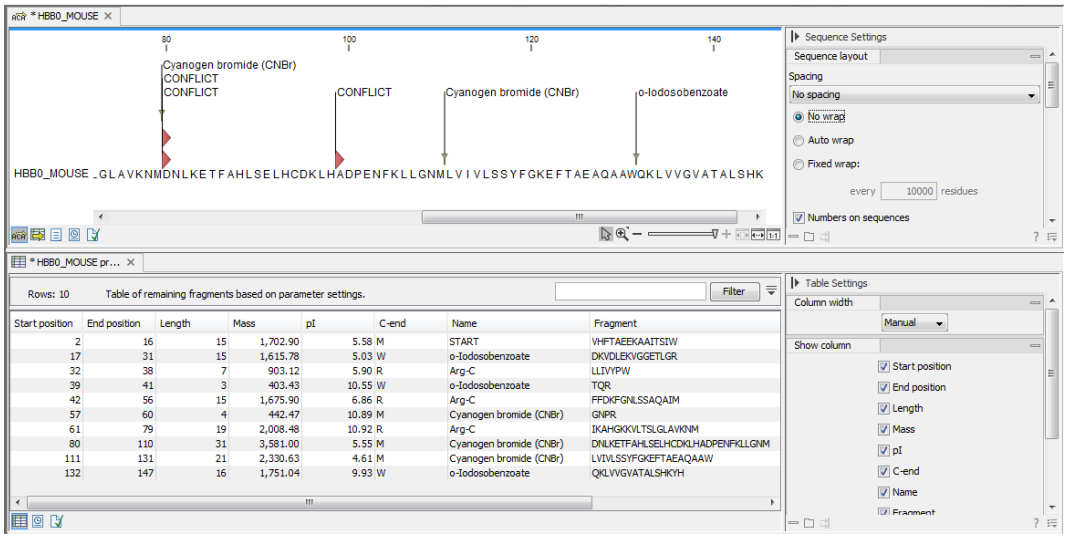
- **Min. and max. number of cleavage sites.** Certain proteolytic enzymes cleave at many positions in the amino acid sequence. For instance proteinase K cleaves at nine different amino acids, regardless of the surrounding residues. Thus, it can be very useful to limit the number of actual cleavage sites before running the analysis.

- **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 17.28.



Start position	End position	Length	Mass	pI	C-end	Name	Fragment
2	16	15	1,702.90	5.58	M	START	VHFTAEEKAAITSIW
17	31	15	1,615.78	5.03	W	o-Iodosobenzoate	DKVDLEKVGGETLGR
32	38	7	903.12	5.90	R	Arg-C	LLIVYPW
39	41	3	403.43	10.55	W	o-Iodosobenzoate	TQR
42	56	15	1,675.90	6.86	R	Arg-C	FFDKFGNLSSAQAIM
57	60	4	442.47	10.89	M	Cyanogen bromide (CNBr)	GNPR
61	79	19	2,008.48	10.92	R	Arg-C	IKAHGKKVLTSLGLAVKNM
80	110	31	3,581.00	5.55	M	Cyanogen bromide (CNBr)	DNLKETFAHLSELHCDKLHADPENFKLLGNM
111	131	21	2,330.63	4.61	M	Cyanogen bromide (CNBr)	LVIVLSSYFGKEFTAEAQAAW
132	147	16	1,751.04	9.93	W	o-Iodosobenzoate	QKLVVGVATALSHKYH

Figure 17.28: *The result of the proteolytic cleavage detection.*

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 these.

17.10.1 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 17.29.

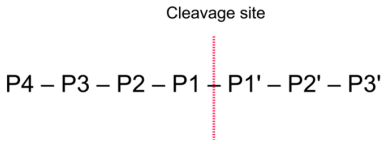


Figure 17.29: *Nomenclature of the peptide substrate. The substrate is cleaved between position P1-P1'.*

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 17.30).

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/`

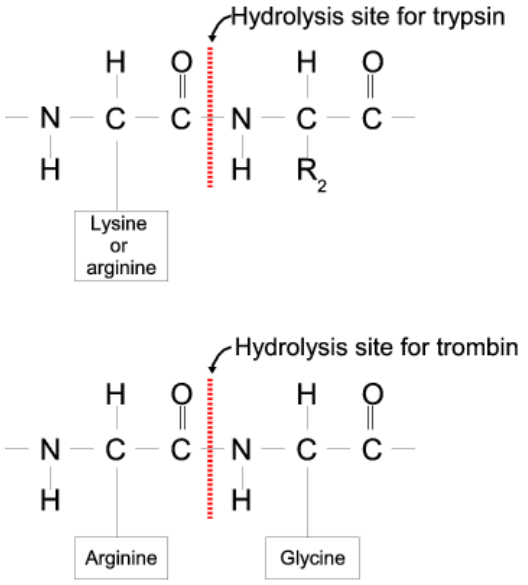


Figure 17.30: *Hydrolysis of the peptide bond between two amino acids. Trypsin cleaves unspecifically at lysine or arginine residues whereas trombin cleaves at arginines if asparate or glutamate is absent.*

Chapter 18

Primers

Contents

18.1 Primer design - an introduction . . . 406
18.1.1 General concept . . . 406
18.1.2 Scoring primers . . . 408
18.2 Setting parameters for primers and probes . . . 408
18.2.1 Primer Parameters . . . 408
18.3 Graphical display of primer information . . . 410
18.3.1 Compact information mode . . . 410
18.3.2 Detailed information mode . . . 411
18.4 Output from primer design . . . 412
18.5 Standard PCR . . . 413
18.5.1 When a single primer region is defined . . . 414
18.5.2 When both forward and reverse regions are defined . . . 415
18.5.3 Standard PCR output table . . . 416
18.6 Nested PCR . . . 417
18.7 TaqMan . . . 418
18.8 Sequencing primers . . . 420
18.9 Alignment-based primer and probe design . . . 421
18.9.1 Specific options for alignment-based primer and probe design . . . 421
18.9.2 Alignment based design of PCR primers . . . 423
18.9.3 Alignment-based TaqMan probe design . . . 425
18.10 Analyze primer properties . . . 426
18.11 Find binding sites and create fragments . . . 427
18.11.1 Binding parameters . . . 428
18.11.2 Results - binding sites and fragments . . . 429
18.12 Order primers . . . 431

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.

18.1 Primer design - an introduction

Primer design can be accessed in two ways:

Toolbox | **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 18.1), the basic options for viewing the template sequence are the same as for the standard sequence view (see section 12 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.

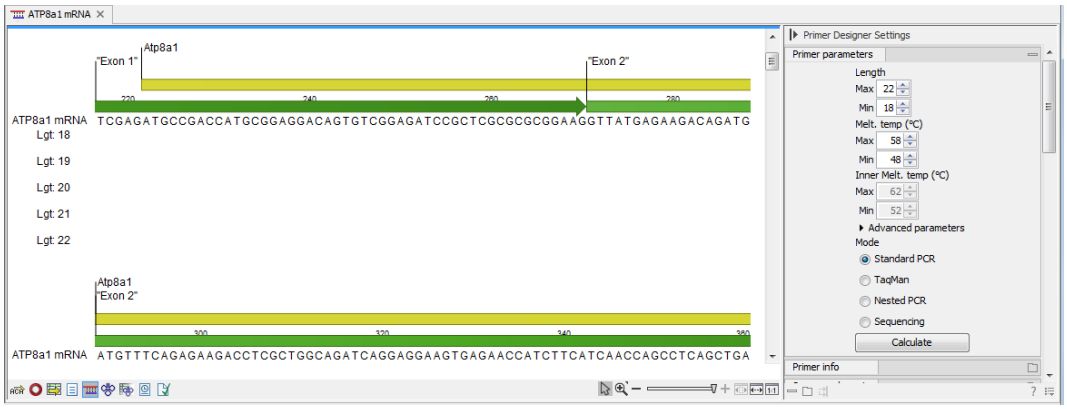


Figure 18.1: *The initial view of the sequence used for primer design.*

18.1.1 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 possibly 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 18.2).

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 showed 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 fulfils all criteria and a red circle indicates a primer which fails to meet one or

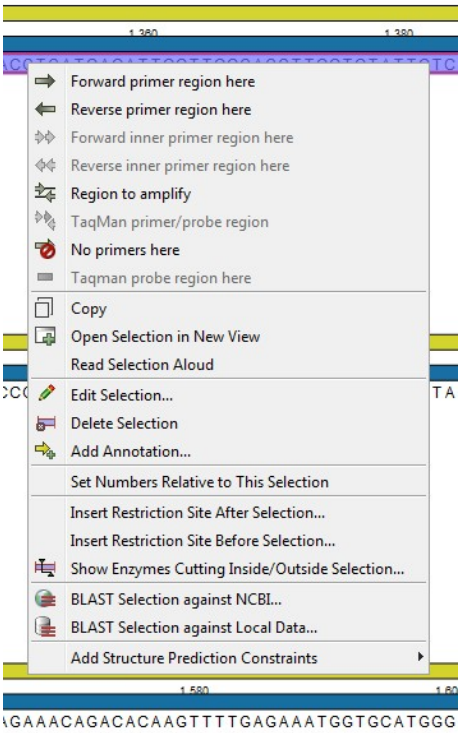


Figure 18.2: *Right-click menu allowing you to specify regions for the primer design*

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.

18.1.2 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.

18.2 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 18.3).

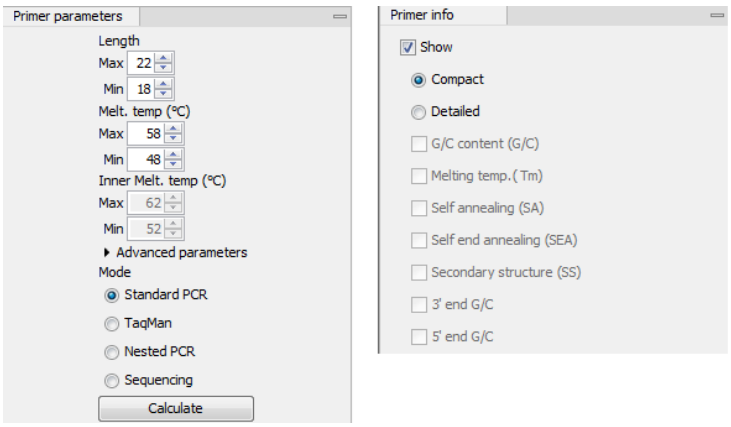


Figure 18.3: *The two groups of primer parameters (in the program, the Primer information group is listed below the other group).*

18.2.1 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 CG 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

18.3 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 18.3).

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.

18.3.1 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 18.4).

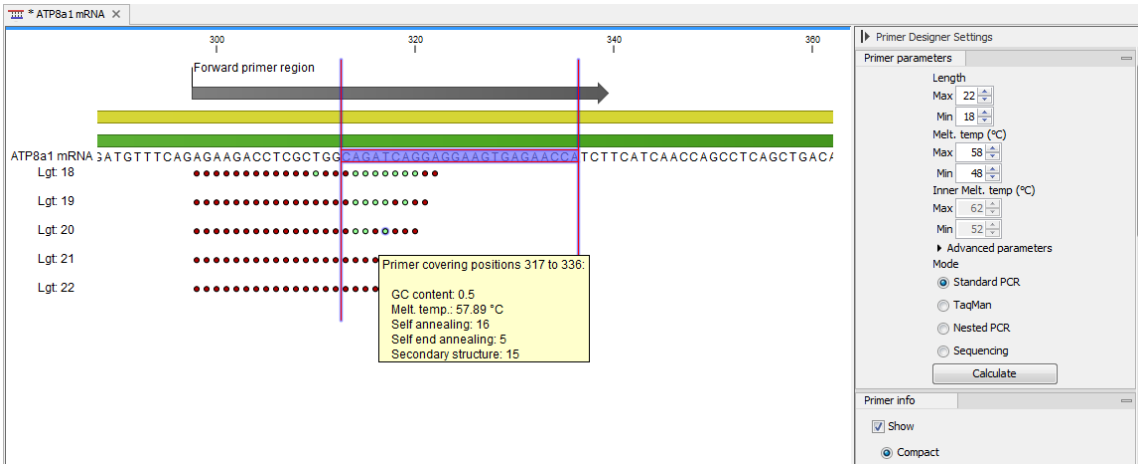


Figure 18.4: *Compact information mode.*

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 fulfils 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, rec circles will appear at the starting points of the primers which fail to meet this requirement.

18.3.2 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 18.5).

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

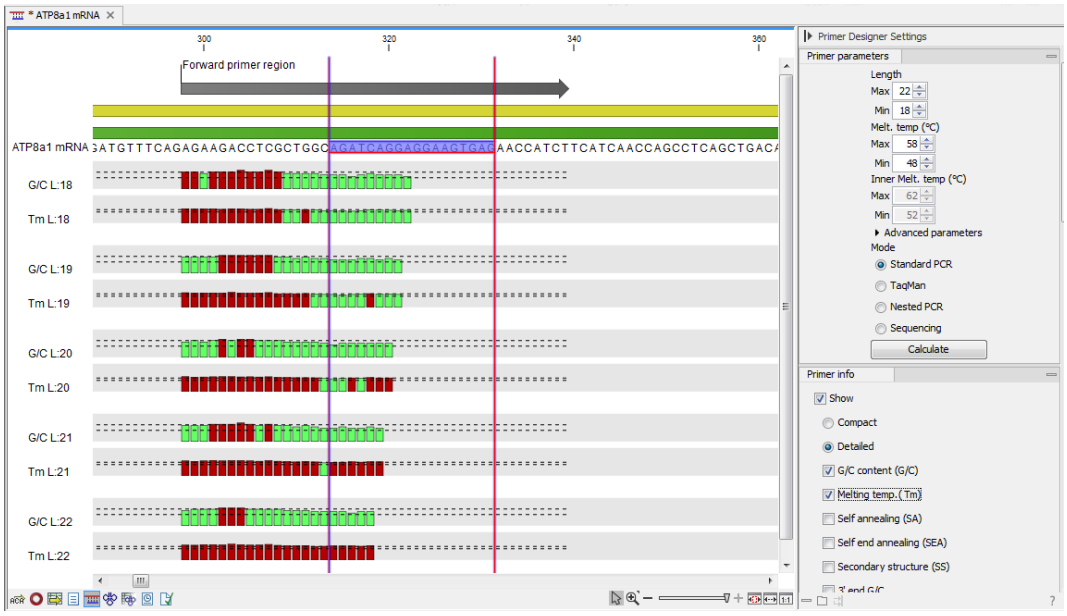


Figure 18.5: *Detailed information mode.*

- 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.

18.4 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 18.6).

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.

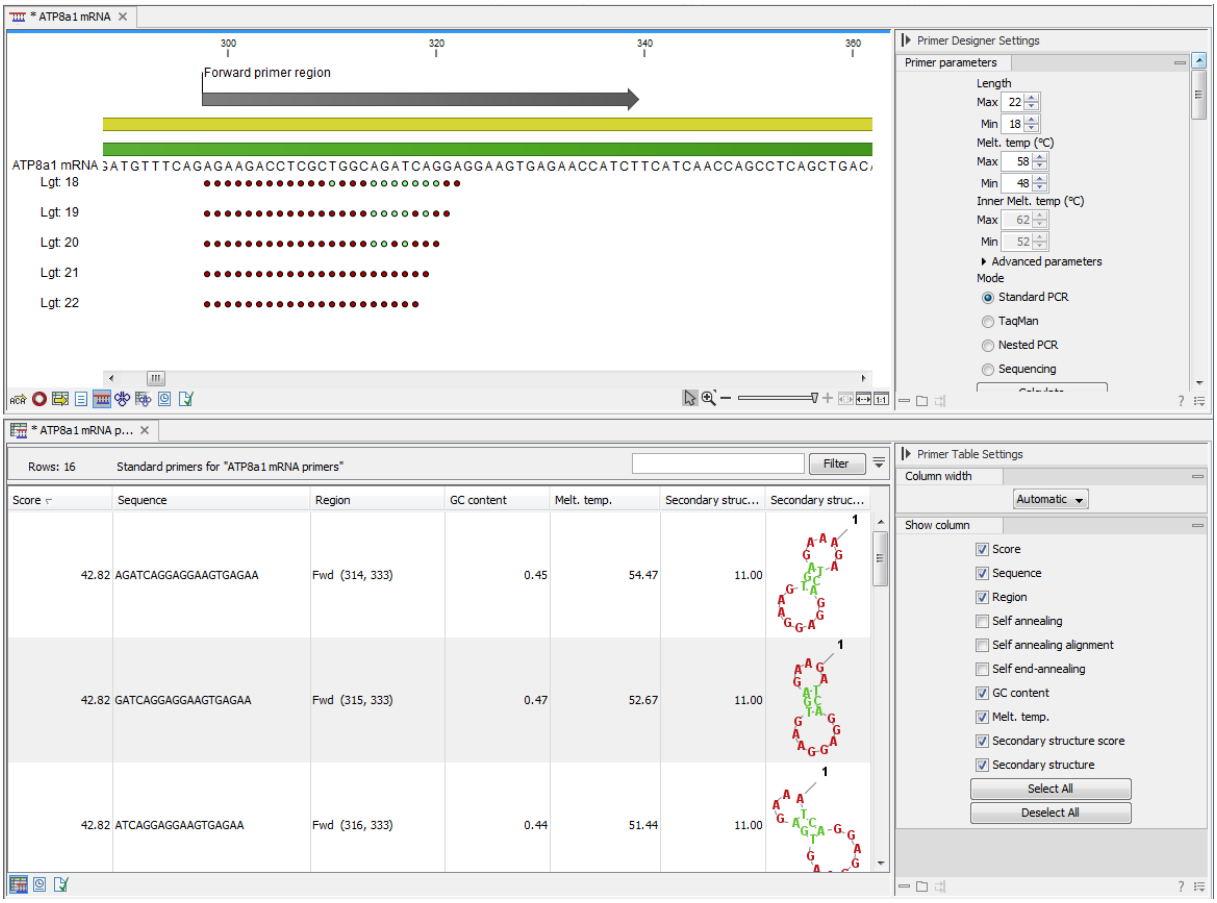


Figure 18.6: *Proposed primers.*

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**.

18.5 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 18.3) and setting the desired parameter values in the Primer Parameters preference group, the **Calculate** button will activate the primer design algorithm.

18.5.1 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 18.7).

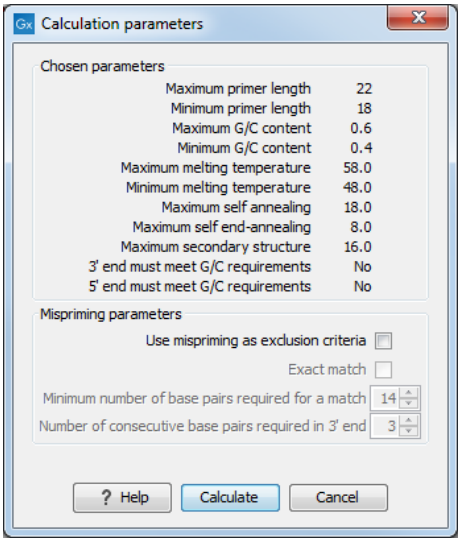


Figure 18.7: *Calculation dialog for PCR primers when only a single primer region has been defined.*

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.

18.5.2 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 18.8).

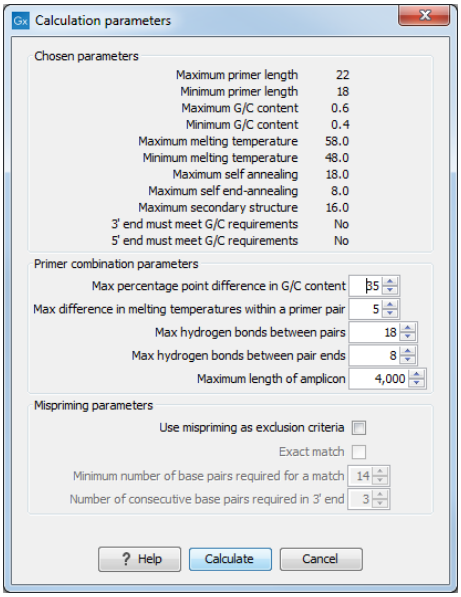


Figure 18.8: *Calculation dialog for PCR primers when two primer regions have been defined.*

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 18.5.1). 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.

18.5.3 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

18.6 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 18.3) 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 18.9).

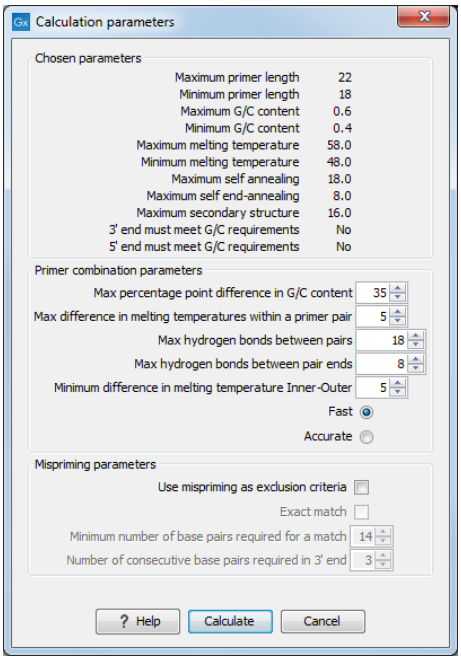


Figure 18.9: *Calculation dialog for nested primers.*

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.

18.7 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 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 18.3) 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 18.10) which is similar to the *Nested PCR* dialog described above (see section 18.6).

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).

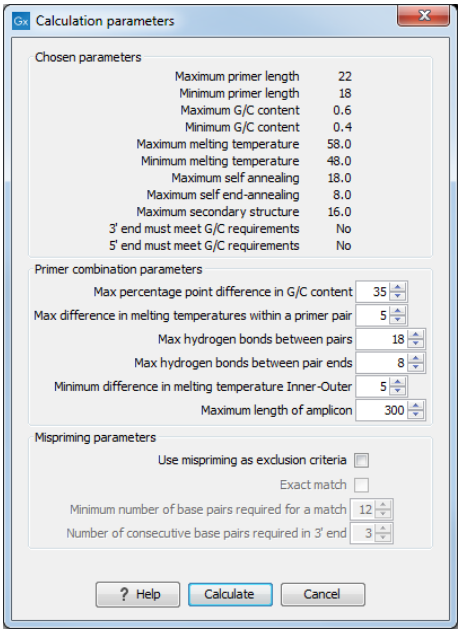


Figure 18.10: *Calculation dialog for taqman primers.*

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.

18.8 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 18.3) 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 18.11).

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 18.5 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.

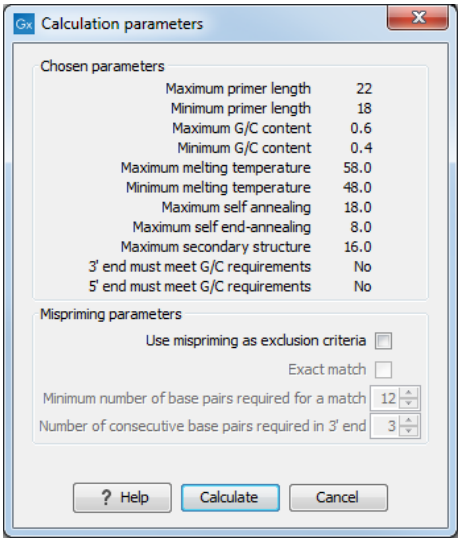


Figure 18.11: *Calculation dialog for sequencing primers.*

For each solution, the single primer information described under Standard PCR is available in the table.

18.9 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:

Toolbox | **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 18.12), the basic options for viewing the template alignment are the same as for the standard view of alignments (see section 21 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.

18.9.1 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 18.12):

- In the **Mode** submenu, specify either:

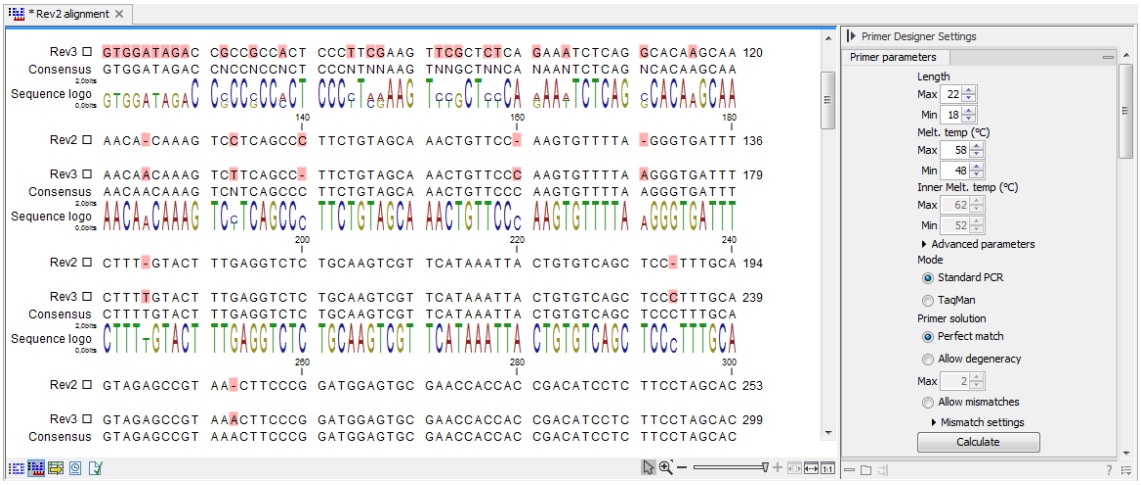


Figure 18.12: *The initial view of an alignment used for primer design.*

  - **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 18.13):

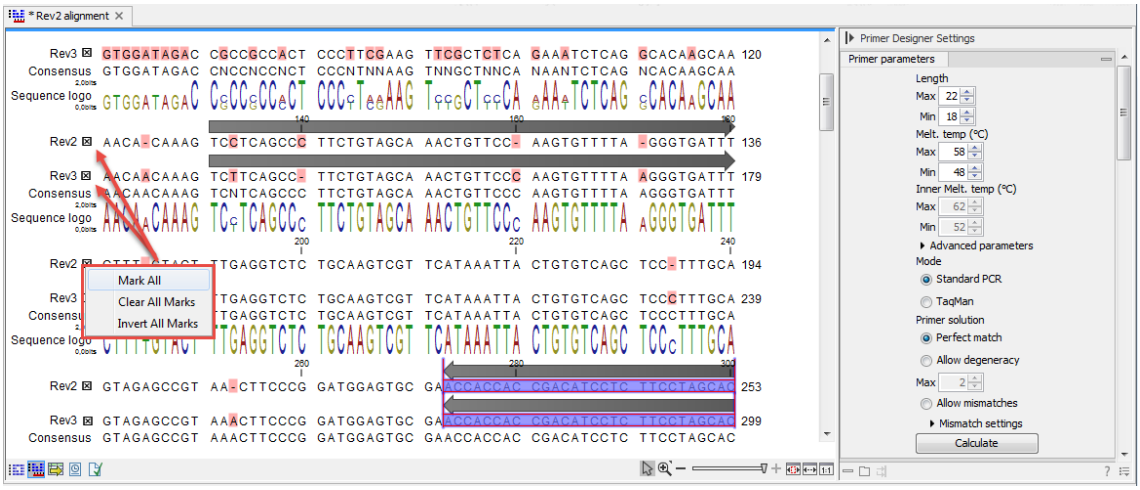


Figure 18.13: *The initial view of an alignment used for primer design.*

- 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.

18.9.2 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 18.14.

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.

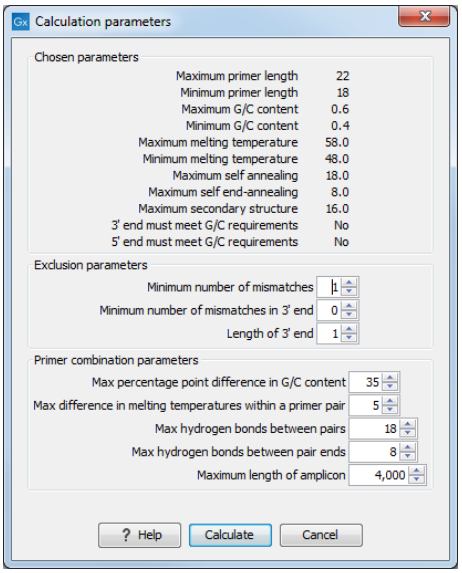


Figure 18.14: *Calculation dialog shown when designing alignment based PCR primers.*

18.9.3 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 18.15 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.

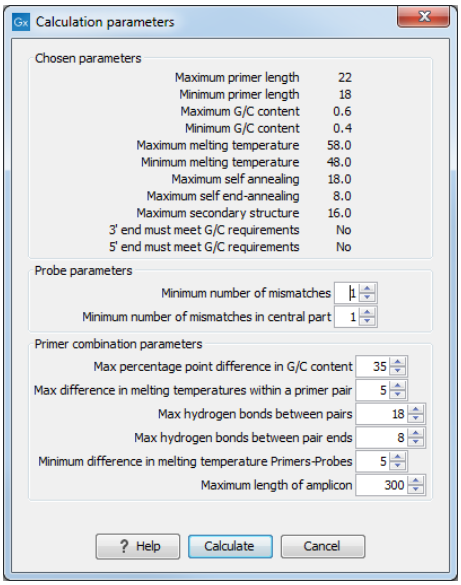


Figure 18.15: *Calculation dialog shown when designing alignment based TaqMan probes.*

18.10 Analyze primer properties

CLC Genomics Workbench can calculate and display the properties of predefined primers and probes:

Toolbox | **Molecular Biology Tools** ( ) | **Primers and Probes** ( )| **Analyze Primer Properties** ( )

If a sequence was selected before choosing the Toolbox action, this sequence is now 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 18.16:

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)

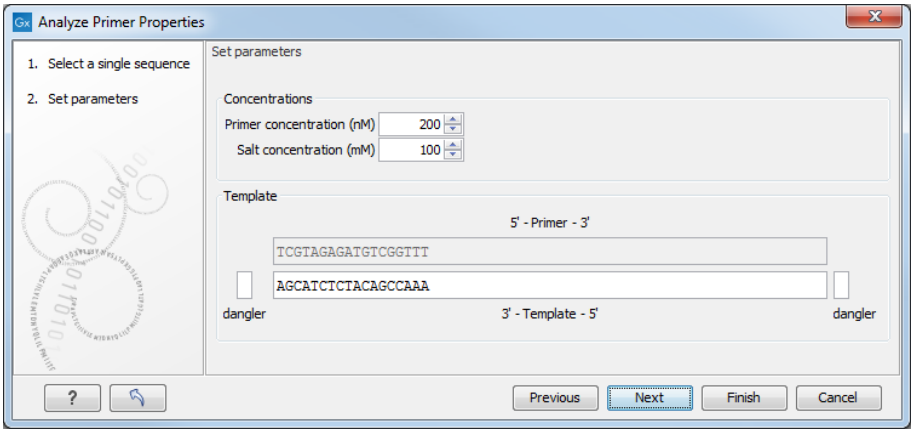


Figure 18.16: *The parameters for analyzing primer properties.*

- **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 18.17:

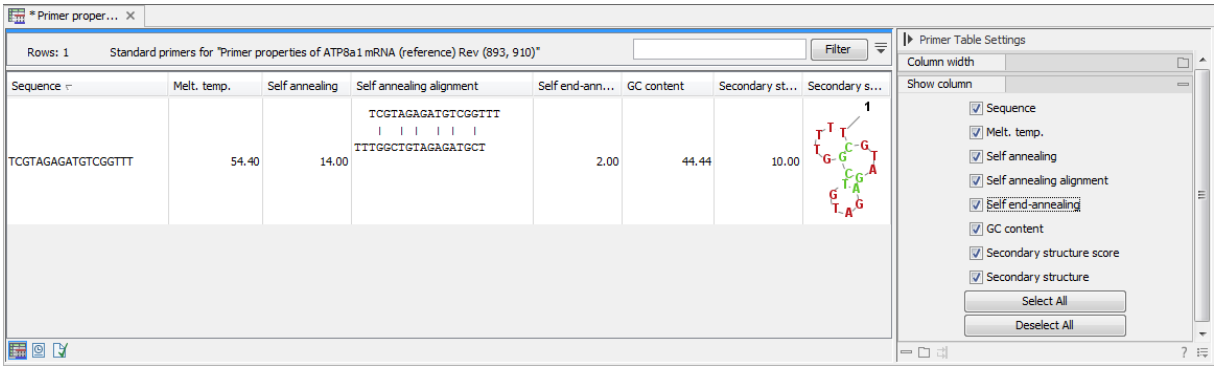


Figure 18.17: *Properties of a primer.*

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 18.5.3.

18.11 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:

Toolbox | 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.

18.11.1 Binding parameters

This opens the dialog displayed in figure 18.18:

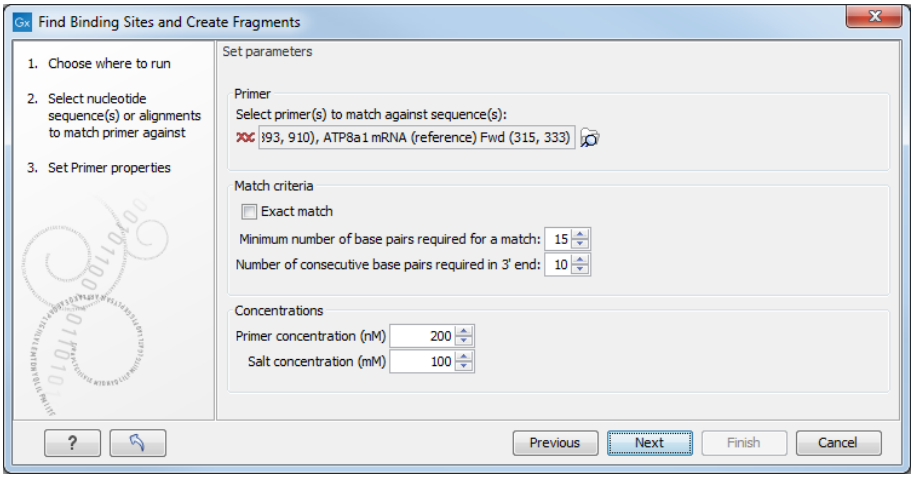


Figure 18.18: *Search parameters for finding primer binding sites.*

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)

18.11.2 Results - binding sites and fragments

Specify the output options as shown in figure 18.19:

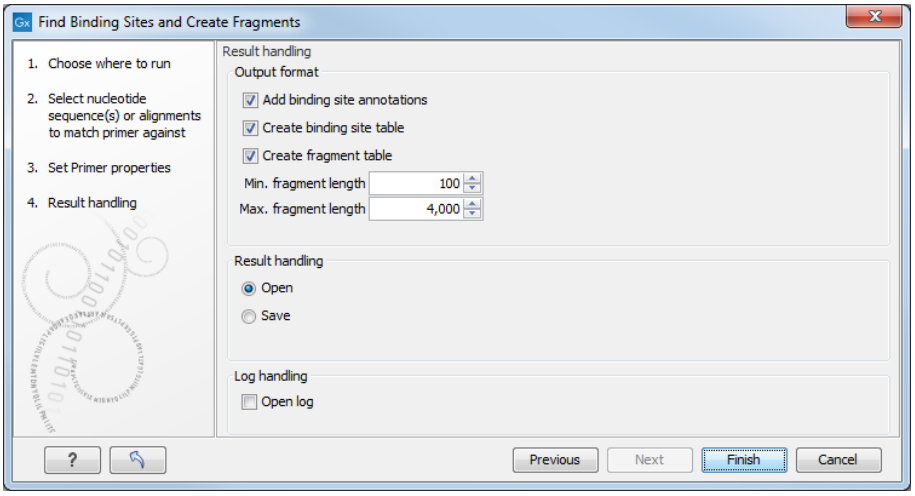


Figure 18.19: *Output options include reporting of binding sites and fragments.*

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 18.20.

The annotation has the following information:

- **Sequence of the primer**. Positions with mismatches will be in lower-case (see the fourth position in figure 18.20 where the primer has an `a` and the template sequence has a `T`).
- **Number of mismatches**.

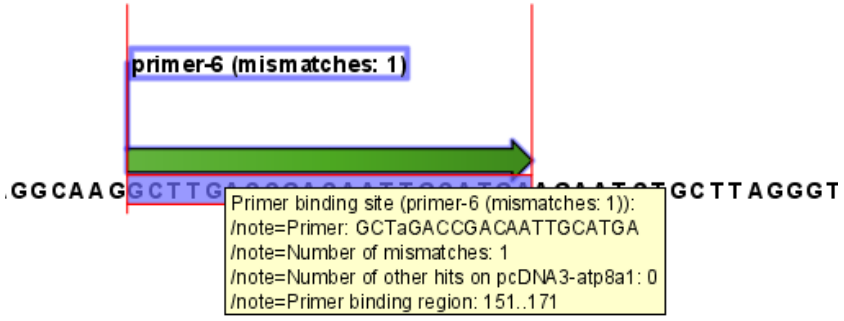


Figure 18.20: *Annotation showing a primer match.*

- **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 18.21.

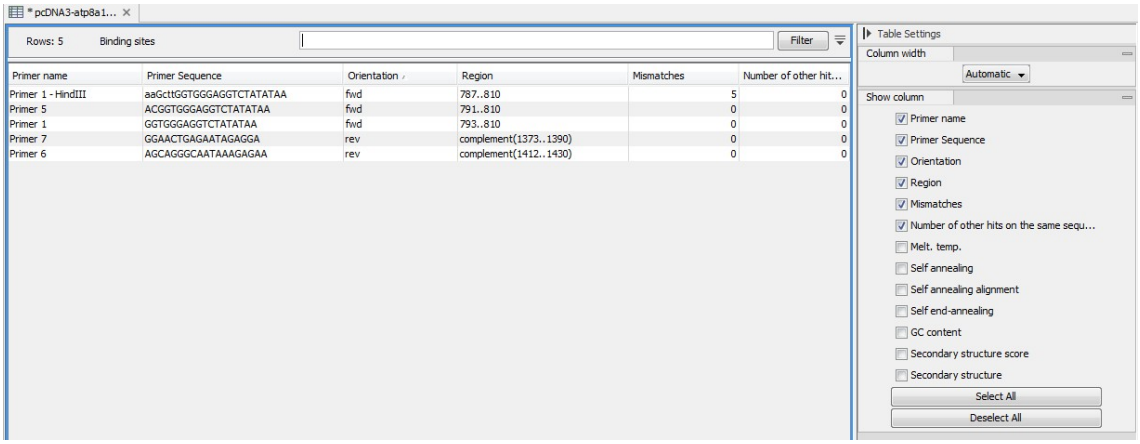


Figure 18.21: *A table showing all binding sites.*

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 18.5.3. You can use this table to browse the binding sites. If you make a split view of the table and the sequence (see section 2.1.5), 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 18.22.

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

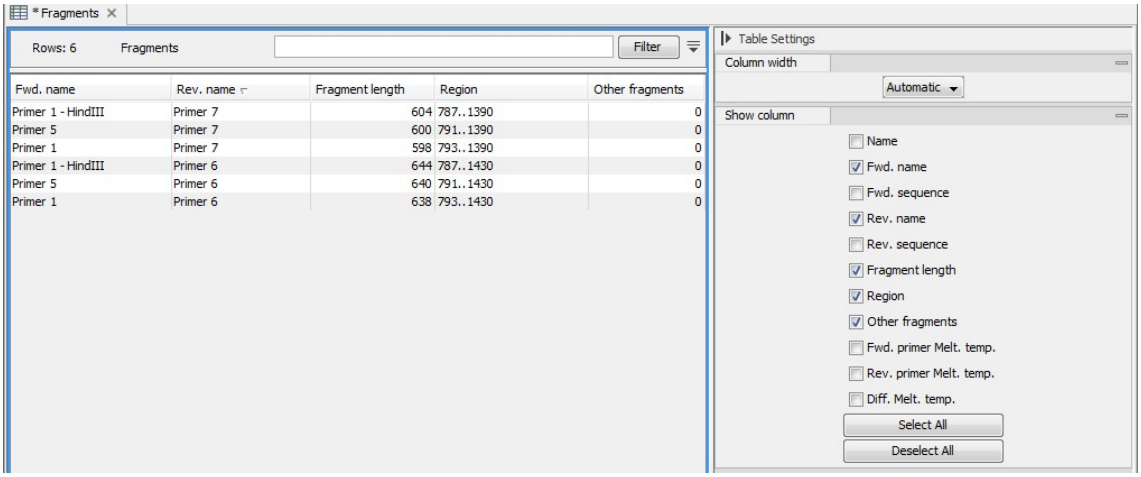


Figure 18.22: *A table showing all possible fragments of the specified size.*

the sequence (see section 2.1.5), 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 18.23.

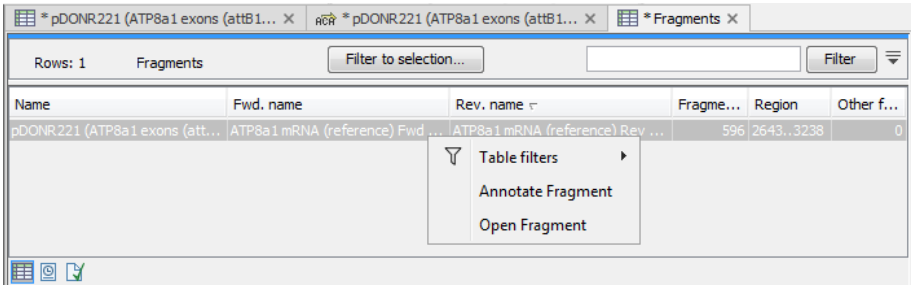


Figure 18.23: *Right-clicking a fragment allows you to annotate the region on the input sequence or open the fragment as a new sequence.*

This will put a *PCR fragment* annotations on the input sequence covering the region specified in the table. As you can see from figure 18.23, 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 20.3).

18.12 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:

Toolbox | **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 18.24).

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.

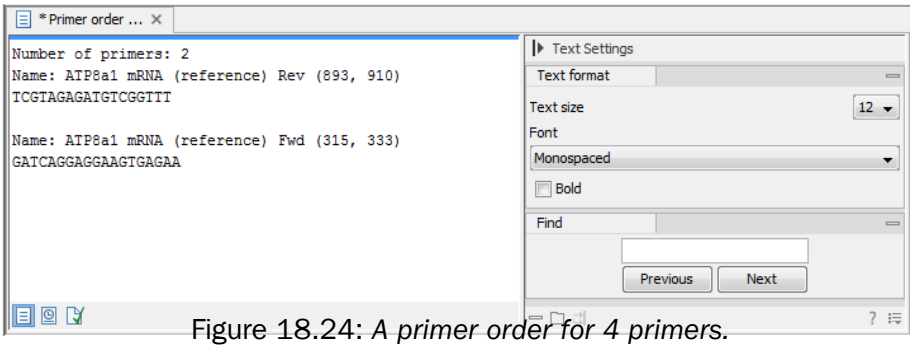


Figure 18.24: *A primer order for 4 primers.*

Chapter 19

Sequencing data analyses

Contents

19.1 Importing and viewing trace data . . . 434
19.1.1 Trace settings in the Side Panel . . . 434
19.2 Trim sequences . . . 435
19.2.1 Trimming using the Trim tool . . . 436
19.2.2 Manual trimming . . . 438
19.3 Assemble sequences . . . 438
19.4 Assemble sequences to reference . . . 440
19.5 Sort sequences by name . . . 443
19.6 Add sequences to an existing contig . . . 446
19.7 View and edit contigs and read mappings . . . 447
19.7.1 View settings in the Side Panel . . . 448
19.7.2 Editing a contig or read mapping . . . 451
19.7.3 Sorting reads . . . 452
19.7.4 Read conflicts . . . 452
19.7.5 Using the mapping . . . 452
19.7.6 Extract reads from a mapping . . . 453
19.7.7 Variance table . . . 455
19.8 Reassemble contig . . . 456
19.9 Secondary peak calling . . . 457
19.10 Extract Consensus Sequence . . . 458
19.11 Combine Reports . . . 462
19.11.1 Combine Reports output . . . 463

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.

19.1 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 6.1).

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 19.1.

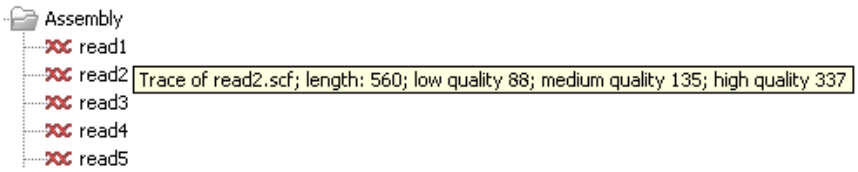


Figure 19.1: *A tooltip displaying information about the quality of the chromatogram.*

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 ( ).

The traces can be scaled by dragging the trace vertically as shown in figure figure 19.2. 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 insection 19.1.1.

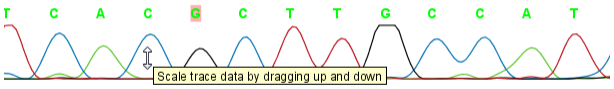


Figure 19.2: *Grab the traces to scale.*

19.1.1 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 19.3):

- **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 19.1.

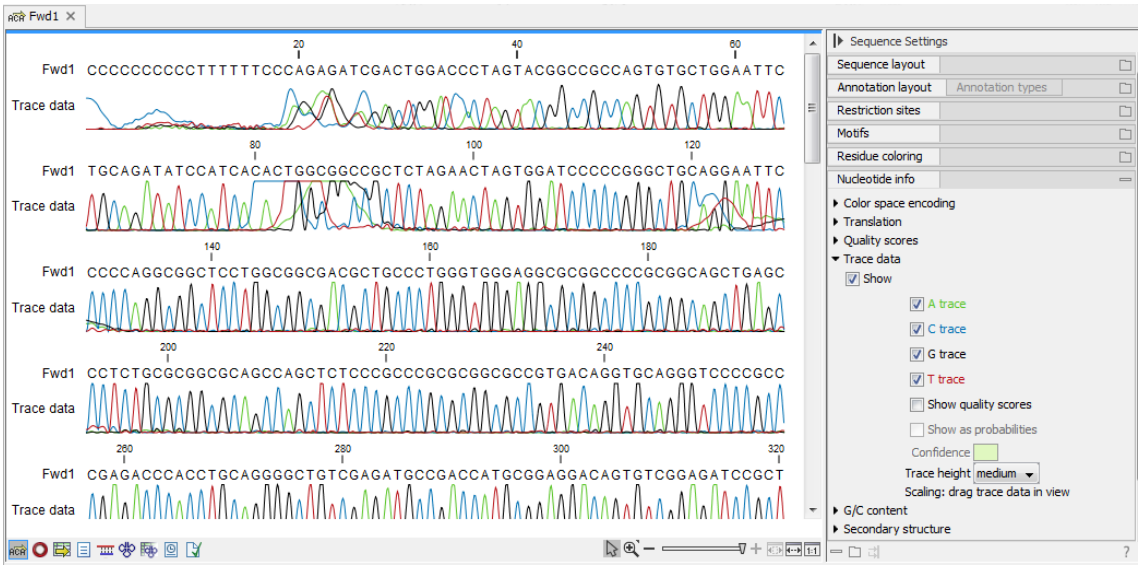


Figure 19.3: *A sequence with trace data. The preferences for viewing the trace are shown in the Side Panel.*

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 27.2.3.

19.2 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 19.4). Such annotated regions are then ignored when using downstream analysis tools located in the same section of the Workbench toolbox, for example Assembly (see section 19.2.2). 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 12.3.2).

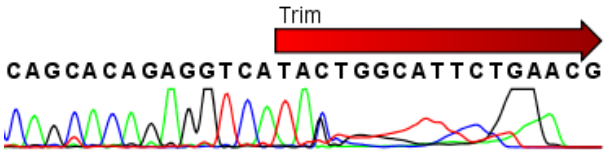


Figure 19.4: *Trimming creates annotations on the regions that will be ignored in the assembly process.*

Note! If you wish to remove regions that are trimmed, you should instead use the NGS Trim Reads tool (see section 25.2).

When exporting sequences in fasta format, there is an option to remove the parts of the sequence

covered by trim annotations.

19.2.1 Trimming using the Trim 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.
- 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:

Toolbox | 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 19.5.

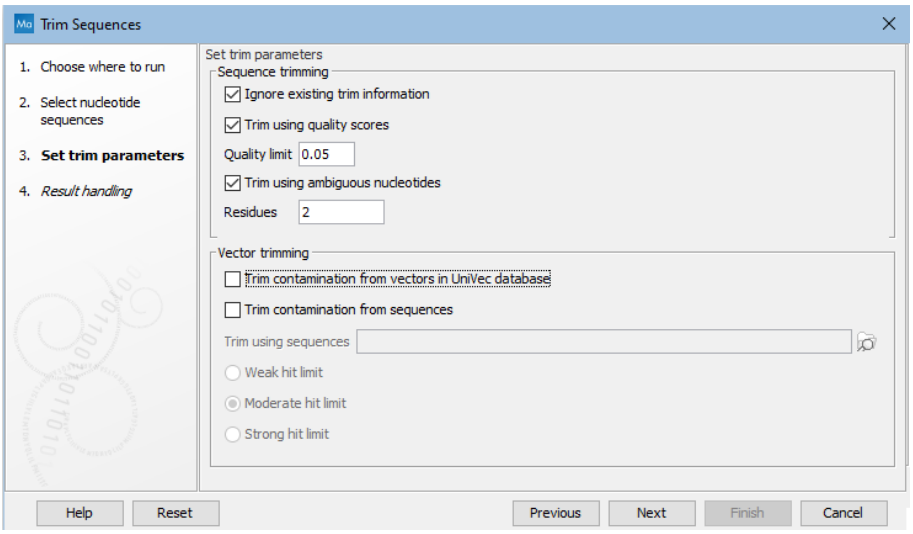


Figure 19.5: *Setting parameters for trimming.*

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: $Q = -10log10(P)$, 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: $p_{error} = 10^{\frac{Q}{-10}}$. (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 http://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 database is included when you install the *CLC Genomics Workbench*). A list of all the vectors in the UniVec database can be found at http://www.ncbi.nlm.nih.gov/VecScreen/replist.html.

  - **Hit limit for vector trimming.** Specifies how strictly vector contamination is trimmed. 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 (http://www.ncbi.nlm.nih.gov/VecScreen/VecScreen.html):
    - **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 $\geq$ 24.
      - Internal match with Score $\geq$ 30.

- **Trim contamination from saved sequences.** This option lets you select your own vector sequences that you have imported into the Workbench. If you select this option, you will be able to select one or more sequences when you click **Next**.

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.

19.2.2 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 Workbench Toolbox as the Trim Sequences tool) as regions to be ignored.

19.3 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 19.4).

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 ( )** in the **Toolbox** instead.

To perform the assembly:

> **Toolbox | 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 19.6

This dialog gives you the following options for assembly:

- **Minimum aligned read length.** The minimum number of nucleotides in a read which must

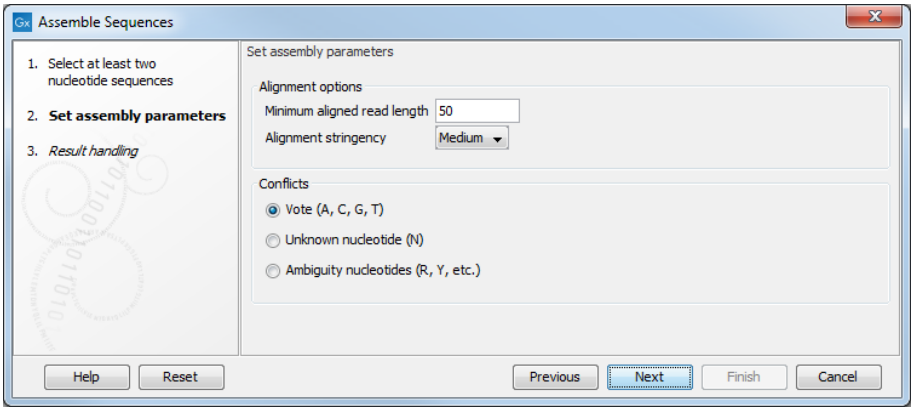


Figure 19.6: *Setting assembly parameters.*

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 19.7.4.

- **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 19.7.

- **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 19.7.7.

- **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 19.7 on how to use the resulting contigs.

19.4 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** ( ) under **Resequencing Analysis** in the **Toolbox**.

To start the assembly:

Toolbox | **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 19.7

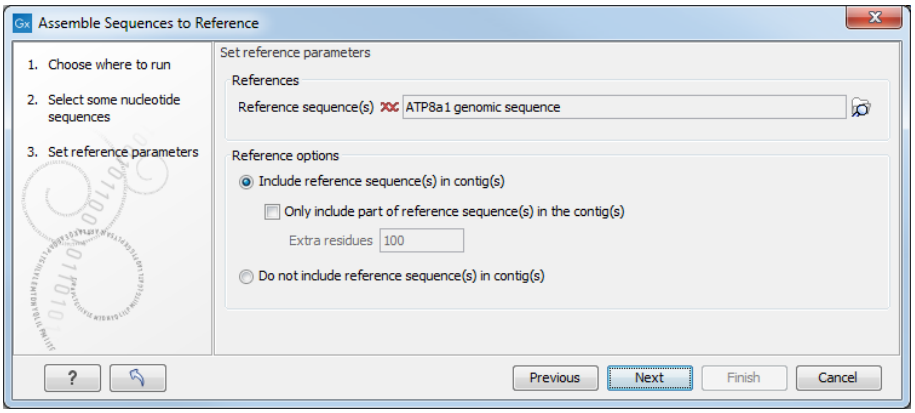


Figure 19.7: *Parameters for how the reference should be handled when assembling sequences to a reference sequence.*

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 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 19.8

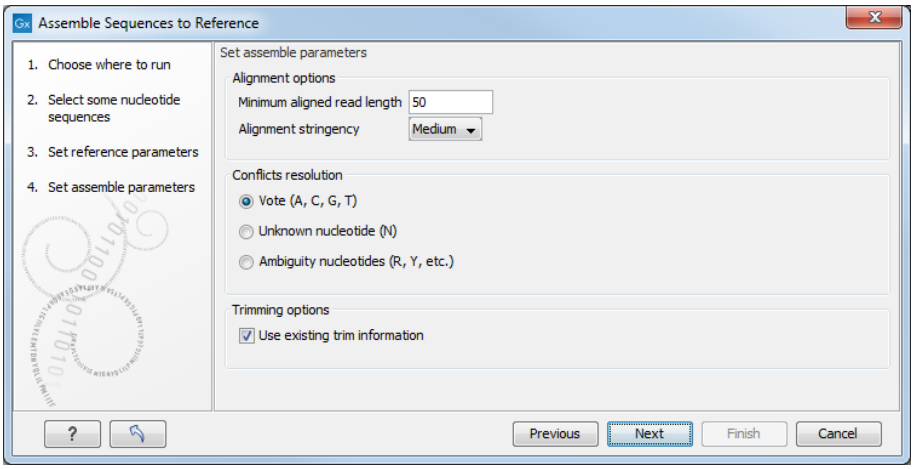


Figure 19.8: *Options for how the input sequences should be aligned and how nucleotide conflicts should be handled.*

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 19.2 for more information about trimming).

Click **Finish** to start the tool. This will start the assembly process. See section 19.7 on how to use the resulting contigs.

19.5 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:

> **Toolbox** | **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 19.9.

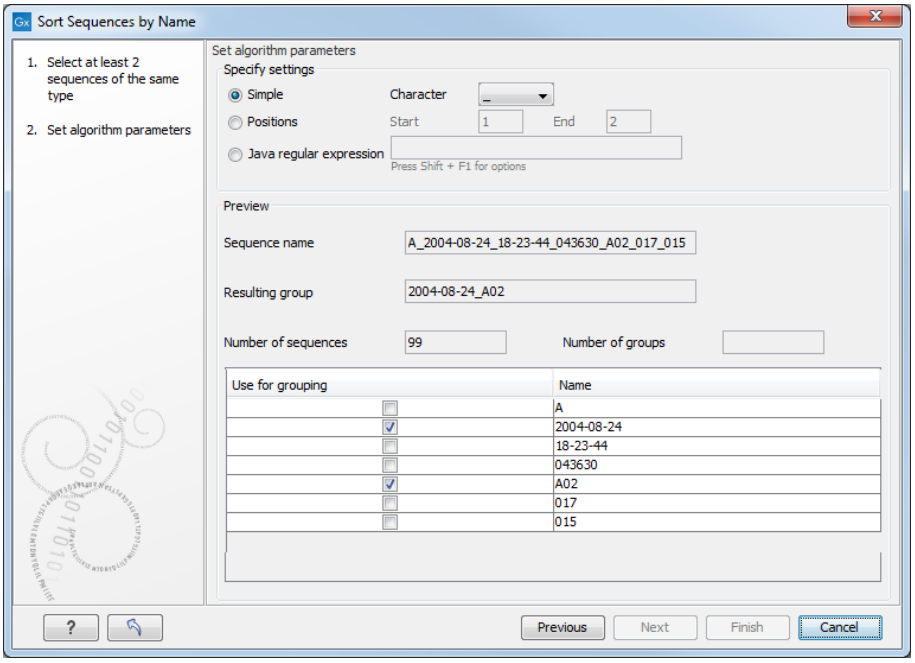


Figure 19.9: *Splitting up the name at every underscore (_) and using the date and analysis position for grouping.*

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 19.9.

Advanced splitting using regular expressions

You can see a more detail explanation of the regular expressions syntax in section 15.8.3.

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 19.9 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 19.10.

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.

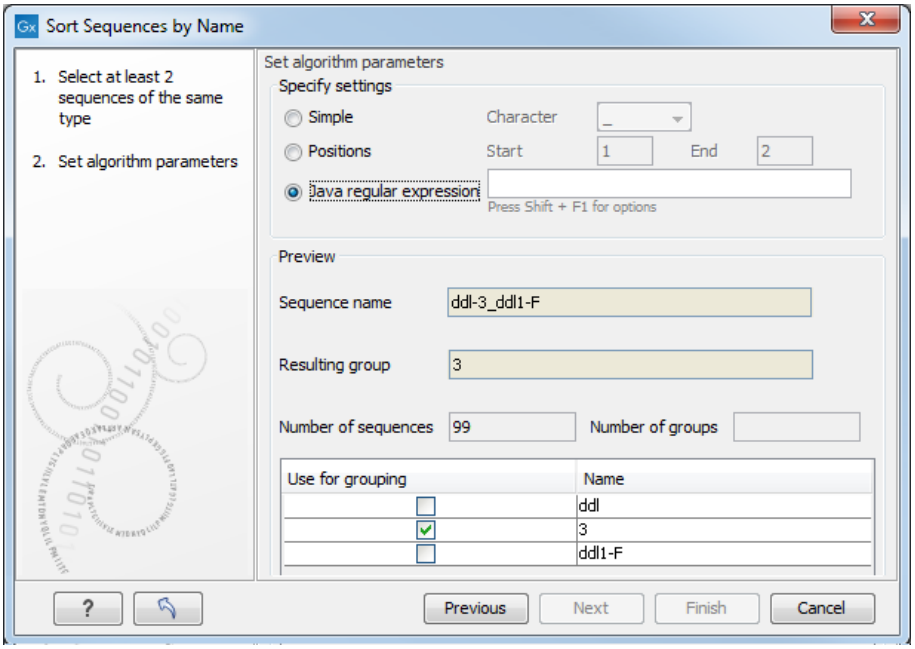


Figure 19.10: *Dividing the sequence into three groups based on the number in the middle of the name.*

19.6 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 start the assembly:

Toolbox in the Menu Bar | Molecular Biology Tools ( ) | Sanger Sequencing Analysis ( )| Add Sequences to Contig ( )

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 19.2.1).

When the elements are selected, click **Next**, and you will see the dialog shown in figure 19.11

The options in this dialog are similar to the options that are available when assembling to a reference sequence (see section 19.4).

Click **Finish** to start the tool. This will start the assembly process. See section 19.7 on how to use the resulting contig.

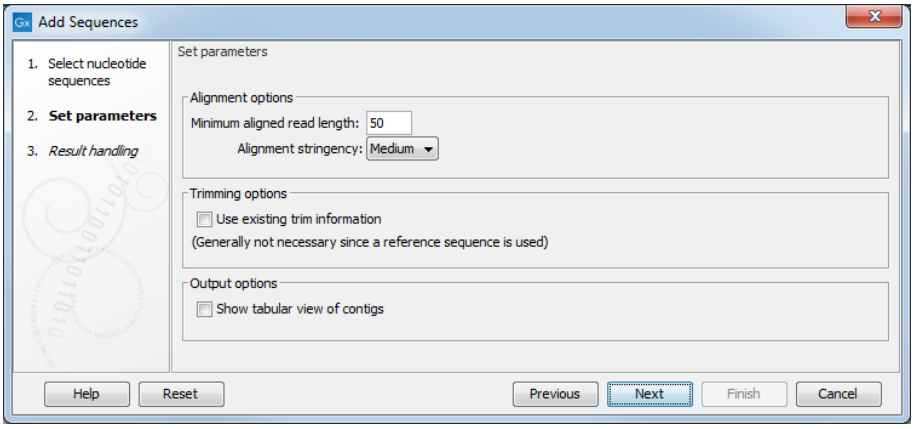


Figure 19.11: *Setting assembly parameters when assembling to an existing contig.*

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.

19.7 View and edit contigs and read mappings

The results of the assembly or mapping (assembly to a reference) are respectively contigs or a read mapping. In both cases the sequence reads have been aligned (see figure 19.12). If multiple reference sequences were used, this information will be in a table where the actual visual mapping can be opened by double-clicking.

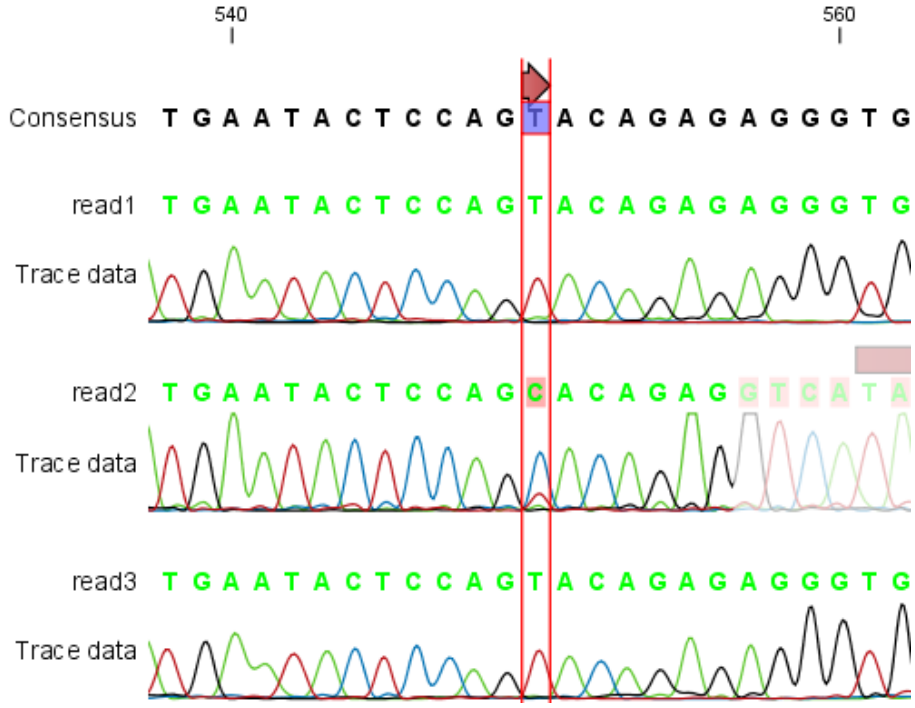


Figure 19.12: *The view of a contig. Note that you can zoom to a very detailed level.*

You can see that color of the residues and trace at the end of one of the reads has been faded. This indicates that this region has not contributed to the contig or mapping. This may be due to trimming before or during the assembly or to misalignment to the other reads.

You can easily adjust the trimmed area to include more of the read in the contig or mapping: simply drag the edge of the faded area as shown in figure 19.13.

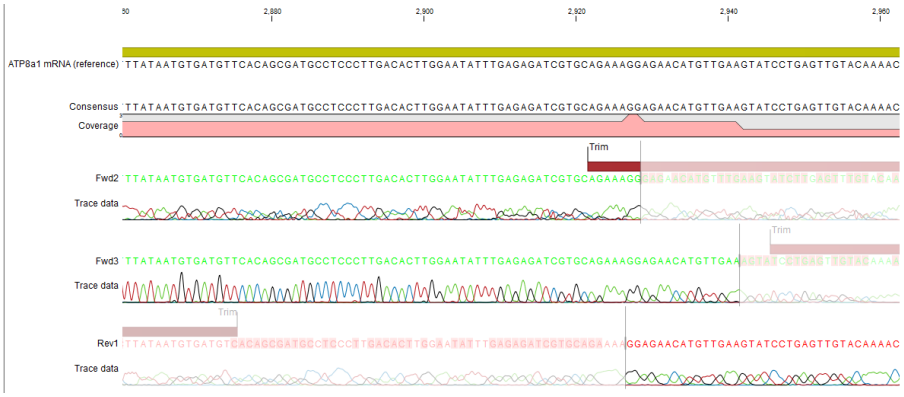


Figure 19.13: *Dragging the edge of the faded area.*

Note! The handles for dragging are only available at the zoom level where residues can be seen. This means that you need to have zoomed in to 100% or more and chosen **Compactness** levels "Not compact", "Low" or "Packed".

Residues are colored green unless they were reversed to map. in which case they will be red. The colors can be changed in the **Side Panel** as described in section 19.7.1

If you find out that the reversed reads should have been the forward reads and vice versa, you can reverse complement the whole contig or mapping. Right-click in the empty white area of the contig or mapping and choose to **Reverse Complement Sequence**.

19.7.1 View settings in the Side Panel

The View Settings panel for assemblies and read mappings with fewer than 2000 reads resembles that of alignments (see section 21.2) but has some extra preferences described below (figure 19.14).

- **Read layout.** This section appears at the top of the **Side Panel** when viewing a stand-alone read mapping:
  - **Compactness.** The compactness setting options let you control the level of detail to be displayed. This setting affects many of the other settings in the **Side Panel** as well as the general behavior of the view. For example: if the compactness is set to **Compact**, you will not be able to see quality scores or annotations on the reads, even if these are turned on via the "Nucleotide info" palette of the Side Panel. You can change the Compactness setting in the Side Panel directly, or you can use the shortcut: press and hold the Alt key while you scroll 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 this is relevant. To view such information, additional viewing options under the **Nucleotide info** view settings must also selected. For further details on these, please see section 19.1.1 and section 12.
    - **Low.** Hides trace data, quality scores and puts the reads' annotations on the sequence.

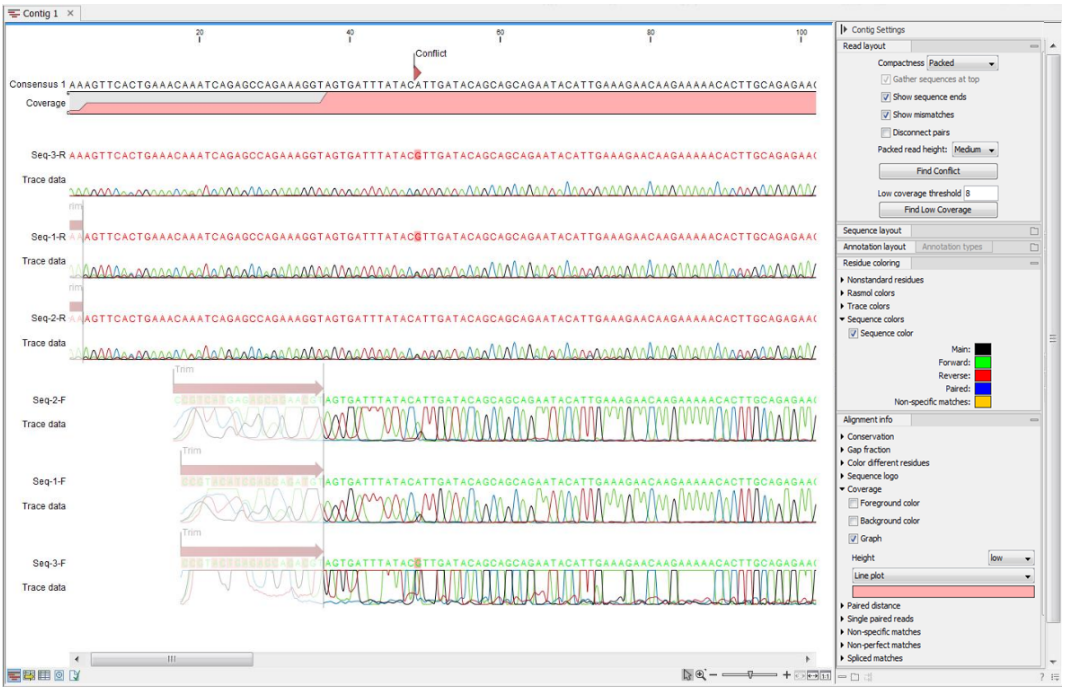


Figure 19.14: *An example of contig, result of an assembly with less than 2000 reads*

  * **Medium.** The labels of the reads and their annotations are hidden, and the residues of the reads cannot be seen.
  * **Compact.** Even less space between the reads.
  * **Packed.** All the other compactness settings will stack the reads on top of each other, but the packed setting will use all space available for displaying the reads. When zoomed in to 100%, you can see the residues but when zoomed out the reads will be represented as lines just as with the Compact setting. The packed mode is very useful when viewing large amounts of data. However certain functionality possible with other views are not available in packed view. For example, no editing of the read mapping or selections of it can be done and color coding changes are not possible.

- **Gather sequences at top.** Enabling this option affects the view that is shown when scrolling horizontally. If selected, the sequence reads which did not contribute to the visible part of the mapping will be omitted whereas the contributing sequence reads will automatically be placed right below the reference. This setting is not relevant when the compactness is packed.
- **Show sequence ends.** Regions that have been trimmed are shown with faded traces and residues. This illustrates that these regions have been ignored during the assembly.
- **Show mismatches.** When the compactness is packed, you can highlight mismatches which will get a color according to the Rasmol color scheme. A mismatch is whenever the base is different from the reference sequence at this position. This setting also causes the reads that have mismatches to be floated at the top of the view.
- **Disconnect pairs.** This option will break up the paired reads in the display (they are still marked as pairs - this just affects the visualization). The reads are marked with

colors for the direction (default red and green) instead of the color for pairs (default blue). This is particularly useful when investigating overlapping pairs in packed view and when the strand / read orientation is important.

- **Packed read height.** When the compactness is set to "packed", you can choose the height of the visible reads. When there are more reads than the height specified, an overflow graph will be displayed below the reads. The overflow graph is shown in the same colors as the sequences, and mismatches in reads are shown as narrow vertical lines. The colors of the small lines represent the mismatching residue. The color codes for the horizontal lines correspond to the color used for highlighting mismatches in the sequences (red = A, blue = C, yellow = G, and green = T), meaning that a red line with half the height of the blue part of the overflow graph will represent a mismatching "A" in half of the paired reads at this particular position.
- **Find Conflict.** Residues that are different from the reference are colored (as default), providing an overview of the conflicts. Clicking this button selects the next position where there is an conflict between the sequence reads. You can also use the Space key to find the next conflict.
- **Low coverage threshold.** All regions with coverage up to and including this value are considered low coverage. When clicking the 'Find low coverage' button the next region in the read mapping with low coverage will be selected.

- **Alignment info.** There is one additional parameter:
  - **Coverage**: Shows how many sequence reads that are contributing information to a given position in the mapping. The level of coverage is relative to the overall number of sequence reads.
    - **Foreground color.** Colors the letters 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 letters using a gradient, where the left side color is used for low coverage and the right side is used for maximum coverage
    - **Graph.** The coverage is displayed as a graph (Learn how to export the data behind the graph in section 6.8).
      - **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.

- **Residue coloring.** There is one additional parameter:
  - **Sequence colors.** This option lets you use different colors for the reads.
    - **Main**. The color of the consensus and reference sequence. Black per default.
    - **Forward**. The color of forward reads (single reads). Green per default.
    - **Reverse**. The color of reverse reads (single reads). Red per default.
    - **Paired**. The color of paired reads. Blue per default. Note that reads from **broken pairs** are colored according to their Forward/Reverse orientation or as a Non-specific match, but with a darker nuance than ordinary single reads.

    - **Non-specific matches**. When a read would have matched equally well another place in the mapping, it is considered a non-specific match. This color will "overrule" the other colors. Note that if you are mapping with several reference sequences, a read is considered a double match when it matches more than once *across all the contigs/references*. A non-specific match is yellow per default.

- **Sequence layout.**
  - **Matching residues as dots** Matching residues will be presented as dots. Only the top sequence will be preserved in its original format.

There are many other viewing options available, both general and aimed as specific elements of a contig or a mapping, which can be adjusted in the View settings. Those covered here were the key ones relevant for standard review of the results.

19.7.2 Editing a contig or read mapping

When editing contigs and read mappings, you are typically interested in confirming or changing single bases, and this can be done simply by:

selecting the base | **typing the right base**

Some users prefer to use lower-case letters in order to be able to see which bases were altered when they use the results later on. In *CLC Genomics Workbench* all changes are recorded in the history log (see section 2.1.1), allowing the user to quickly reconstruct the actions performed in the editing session.

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 19.12). This is useful for discerning the trace curves.

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 (**  **)**

This will show a warning dialog, but you can choose never to see this dialog again by clicking the checkbox at the bottom of the dialog.

Note that for contigs or mappings with more than 1,000 reads, you can only do single-residue replacements (you can't delete or edit a selection). When the compactness is **Packed**, you cannot edit any of the reads.

19.7.3 Sorting reads

If you wish 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.

19.7.4 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**.

19.7.5 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 27.6).

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**.

19.7.6 Extract reads from a mapping

Note that the functionalities described in this page are valid for read mappings. For similar functionalities on tracks, see section 24.2.5.

Extract from Selection Sometimes it is useful to extract part of a mapping for in-depth analysis. This could be the case if you have performed an analysis of a whole genome data set and have found a region that you are particularly interested in analyzing further. Rather than running all further analysis on your full data, you may prefer to run only on a subset of the data. You can extract a subset of your mapping data by running the **Extract from Selection** tool on a selected region in your mapping. The result of running this tool is a new mapping which contains only the reads (and optionally only those that are of a particular type) in your selected region.

To select a region, use the **Selection mode** ( ) (see Section 2.2.3 for a detailed description of the different modes) and select you region of interest in your mapping, then right-click on the reference sequence or on the consensus sequence of the mapping (figure 19.15).

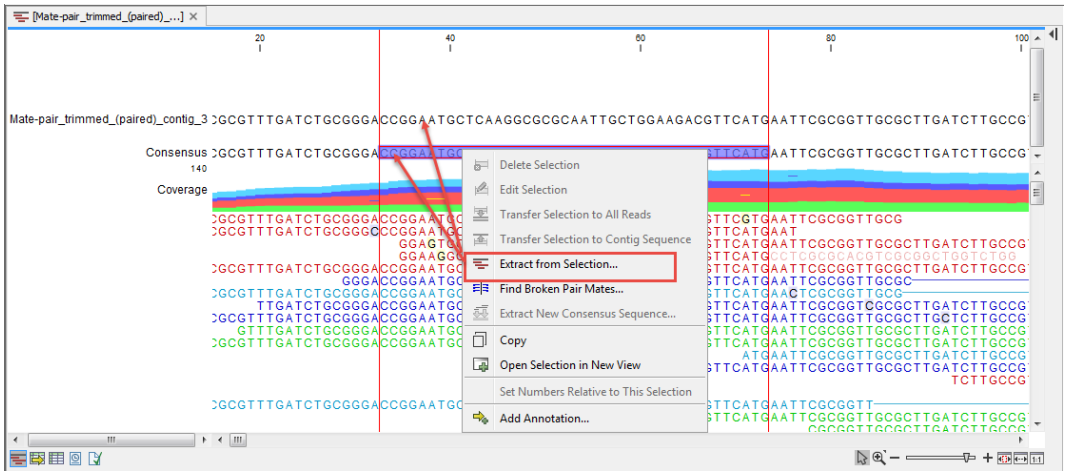


Figure 19.15: *Extracting parts of a mapping using the right-click menu available when clicking on a selected portion of the consensus sequence.*

When you choose the **Extract from Selection** option you are presented by the dialog shown in figure 19.16.

The purpose of this dialog is to let you specify what kind of reads you want to include. Per default all reads are included. The options are:

Paired status **Include intact paired reads** When paired reads are placed within the paired distance specified, they will fall into this category. Per default, these reads are colored in blue.

Include paired reads from broken pairs When a pair is broken, either because only one read in the pair matches, or because the distance or relative orientation is wrong, the reads are placed and colored as single reads, but you can still extract them by checking this box.

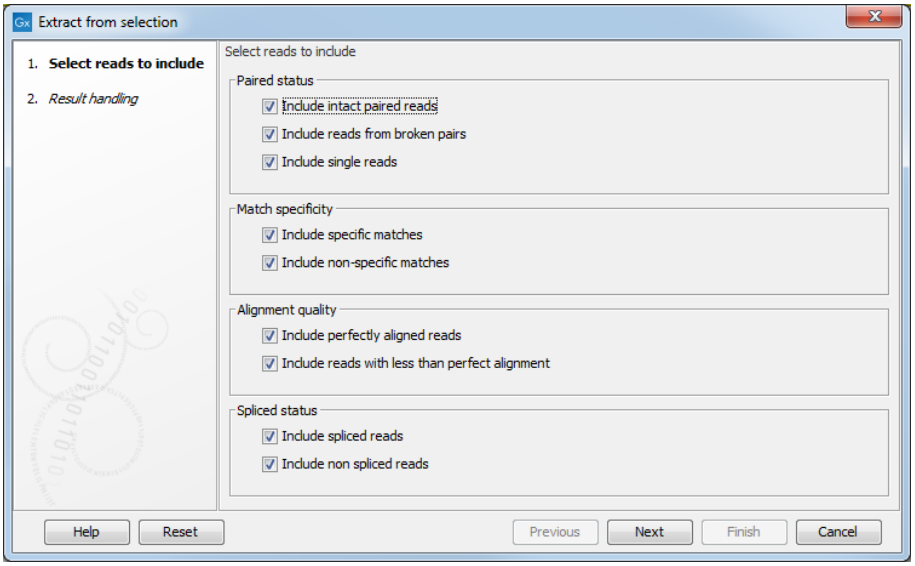


Figure 19.16: *Selecting the reads to include.*

Include single reads This will include reads that are marked as single reads (as opposed to paired reads). Note that paired reads that have been broken during assembly are not included in this category. Single reads that come from trimming paired sequence lists are included in this category.

Match specificity **Include specific matches** Reads that only are mapped to one position.

Include non-specific matches Reads that have multiple equally good alignments to the reference. These reads are colored yellow per default.

Alignment quality **Include perfectly aligned reads** Reads where *the full read* is perfectly aligned to the reference sequence (or consensus sequence for de novo assemblies). Note that at the end of the contig, reads may extend beyond the contig (this is not visible unless you make a selection on the read and observe the position numbering in the status bar). Such reads are not considered perfectly aligned reads because they don't align in their entire length.

Include reads with less than perfect alignment Reads with mismatches, insertions or deletions, or with unaligned nucleotides at the ends (the faded part of a read).

Spliced status **Include spliced reads** Reads that are across an intron.

Include non spliced reads Reads that are not across an intron.

Note that only reads that are completely covered by the selection will be part of the new contig.

One of the benefits of this is that you can actually use this tool to extract subset of reads from a contig. An example work flow could look like this:

1. Select the whole reference sequence
2. Right-click and **Extract from Selection**
3. Choose to include only paired matches
4. Extract the reads from the new file (see section 15.1)

You will now have all paired reads from the original mapping in a list.

Extract Sequences When right-clicking on the sequences (as opposed to the consensus sequence), the menu to the right of figure 19.17 is available, and allows you to Extract Sequences from the mapping as explained in section 15.1. As opposed to the Extract from Selection tool, the Extract sequences will include all reads, not only the ones covered by the selection.

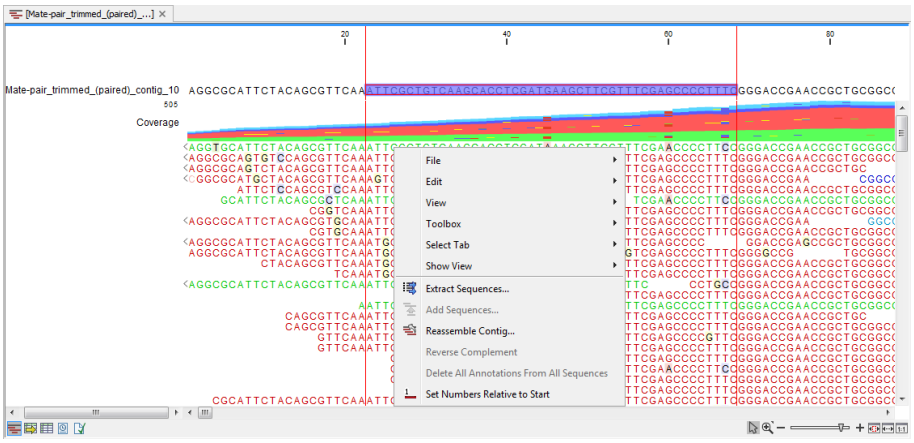


Figure 19.17: *Selecting the reads to include.*

19.7.7 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 19.18.

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 19.18, 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 19.18: *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 19.18). The comments are saved when you **Save** ( ).

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.

19.8 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:

Toolbox | Molecular Biology Tools ( ) | Sanger Sequencing Analysis ( )| Reassemble Contig ( )

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 19.19

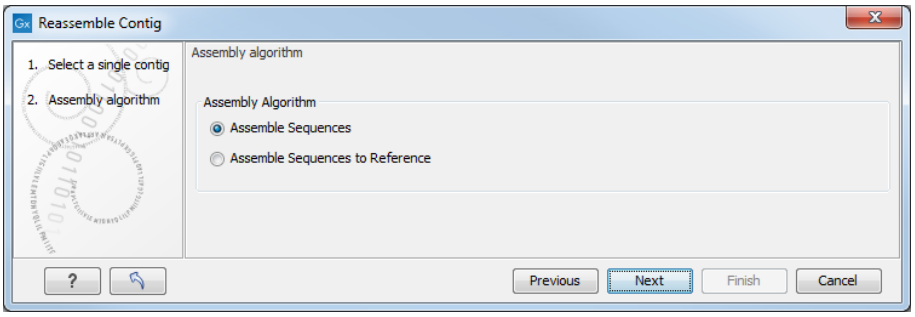


Figure 19.19: *Re-assembling a contig.*

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 19.2.2. 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 19.4.

When you click **Finish**, a new contig is created, so you do not lose the information in the old contig.

19.9 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 19.2).

When a secondary peak is called, the residue is change 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:

Toolbox | **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 19.20.

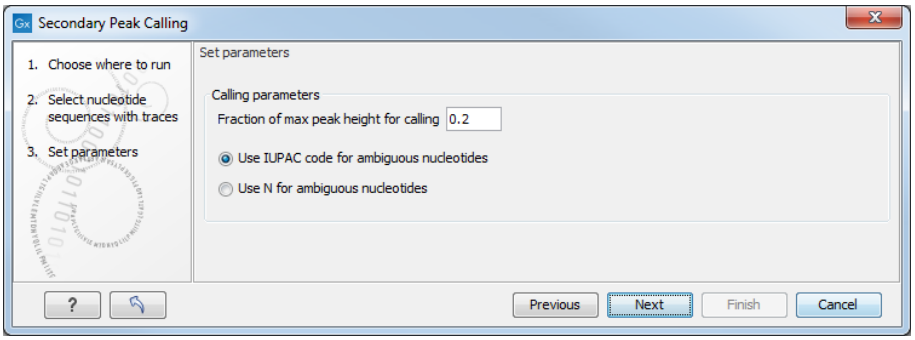


Figure 19.20: *Setting parameters secondary peak calling.*

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.
- **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.

19.10 Extract Consensus Sequence

Using the **Extract Consensus Sequence** tool, a consensus sequence can be extracted from all kinds of read mappings, including those generated from *de novo* assembly or RNA-seq analyses. In addition, you can extract a consensus sequence from nucleotide BLAST results.

Note: Consensus sequences can also be extracted when viewing a read mapping by right-clicking on the name of the consensus or reference sequence, or a selection of the reference sequence, and selecting the option **Extract New Consensus Sequence** ( ) from the menu that appears. The same option is available from the graphical view of BLAST results when right-clicking on a selection of the subject sequence.

To start the **Extract Consensus Sequence** tool, go to:

Toolbox | **Resequencing Analysis** ( ) | **Extract Consensus Sequence** ( )

In the first step, select the read mappings or nucleotide BLAST results to work with.

In the next step, options affecting how the consensus sequence is determined are configured (see figure 27.46).

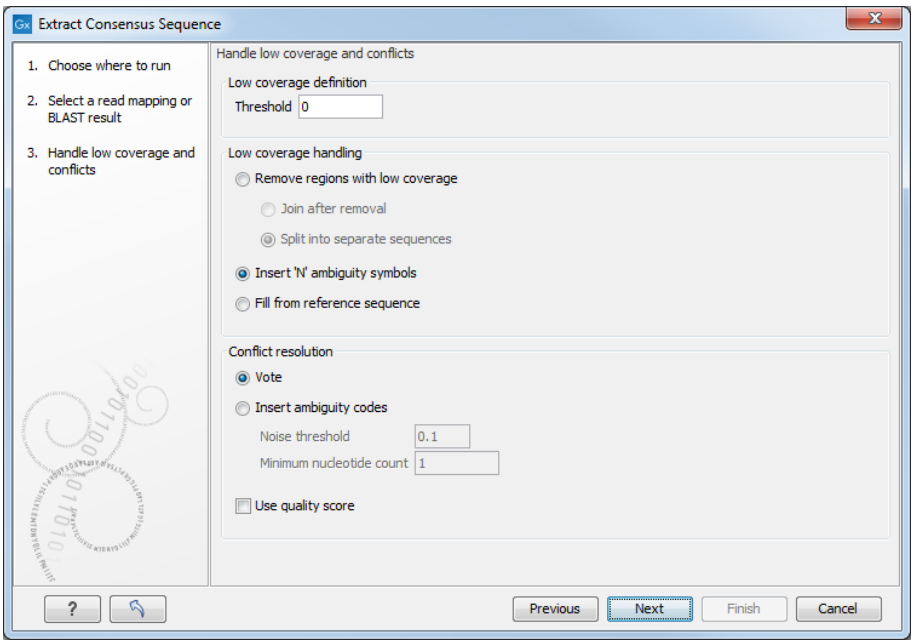


Figure 19.21: *Specifying how the consensus sequence should be extracted.*

Handling low coverage regions

The first step is to define a **low coverage threshold**. Consensus sequence is not generated for reference positions with coverage at or below the threshold specified.

The default value is 0, which means that a reference base is considered to have low coverage when no reads cover this position. Using this threshold, if just a single read covered a particular position, only that read would contribute to the consensus at that position. Setting a higher threshold gives more confidence in the consensus sequence produced.

There are several options for how low coverage regions should be handled:

- **Remove regions with low coverage**. When using this option, no consensus sequence is created for the low coverage regions. There are two ways of creating the consensus sequence from the remaining contiguous stretches of high coverage: either the consensus sequence is **split** into separate sequences when there is a low coverage region, or the low coverage region is simply ignored, and the high-coverage regions are directly **joined**. In this case, an annotation is added at the position where a low coverage region is removed in the consensus sequence produced (see below).
- **Insert 'N' ambiguity symbols**. This simply adds Ns for each base in the low coverage region. An annotation is added for the low coverage region in the consensus sequence produced (see below).
- **Fill from reference sequence**. This option uses the sequence from the reference to construct the consensus sequence for low coverage regions. An annotation is added for the low coverage region in the consensus sequence produced (see below).

Handling conflicts

Settings are provided in the lower part of the wizard for configuring how conflicts or disagreement between the reads should be handled when building a consensus sequence in regions with adequate coverage.

- **Vote** When reads disagree at a given position, the base present in the majority of the reads at that position is used for the consensus.
  - If the most common symbol is a gap, then the consensus symbol is a gap.
  - If there are equal numbers of gaps and non-gaps, then it is one of the other symbols.
  - When choosing between symbols, we choose in the order A - C - G - T.
  - Ambiguous symbols cannot be chosen.

  If the **Use quality score** option is also selected, quality scores are used to decide the base to use for the consensus sequence, rather than the number of reads. The quality scores for each base at a given position in the mapping are summed, and the base with the highest total quality score at a given position is used in the consensus. If two bases have the same total quality score at a location, we follow the order of preference listed above.

  Information about biological heterozygous variation in the data is lost when the **Vote** option is used. For example, in a diploid genome, if two different alleles are present in an almost even number of reads, only one will be represented in the consensus sequence.

- **Insert ambiguity codes** When reads disagree at a given position, an ambiguity code representing the bases at that position is used in the consensus. (The IUPAC ambiguity codes used can be found in Appendix H and G.)

  Unlike the Vote option, some level of information about biological heterozygous variation in the data is retained using this option.

  To avoid the situation where a different base in a single read could lead to an ambiguity code in the consensus sequence, the following options can be configured:

  - **Noise threshold** The percentage of reads where a base must be present at given position for that base to contribute to an ambiguity code. The default value is 0.1, i.e. for a base to contribute to an ambiguity code, it must be present in at least 10 % of the reads at that position.
  - **Minimum nucleotide count** The minimum number of reads a particular base must be present in, at a given position, for that base to contribute to the consensus.

  If no nucleotide passes these two thresholds at a given position, that position is omitted from the consensus sequence.

  If the **Use quality score** option is also selected, summed quality scores are used, instead of numbers of reads for conflict handling. To contribute to an ambiguity code, the summed quality scores for bases at a given position must pass the noise threshold.

In the next step, output options are configured (figure 27.47).

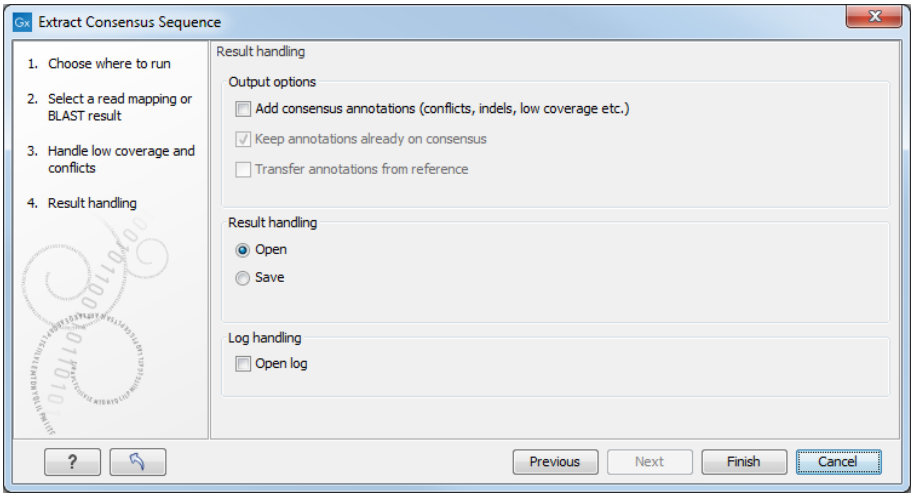


Figure 19.22: *Choose to add annotations to the consensus sequence.*

Consensus annotations

The annotations that can be added to the consensus sequence provide information about conflicts that have been resolved and about low coverage regions (unless you have chosen to split the consensus sequence). Note that for large data sets, a very high number of annotations may be generated, which will cause the tool to take longer to complete. The results will also take up much more disk space.

For stand-alone read mappings, it is possible to transfer existing annotations to the consensus sequence. Since the consensus sequence produced may be broken up, the annotations will also be broken up, and thus may not have the same length as before. In some cases, gaps and low-coverage regions will lead to differences in the sequence coordinates between the input data and the new consensus sequence. The annotations copied will be placed in the region on the consensus that corresponds to the region on the input data, but the actual coordinates might have changed.

Track-based read mappings do not themselves contain annotations and thus the options related to transferring annotations, "Transfer annotations from the reference sequence" and "Keep annotations already on consensus", cannot be selected for this type of input.

Copied/transferred annotations will contain the same qualifier text as the original. That is, 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, not the new sequence, which may differ.

Quality scores on the consensus sequence

The resulting consensus sequence (or sequences) will have quality scores assigned if quality scores were found in the reads used to call the consensus. For a given consensus symbol X we compute its quality score from the "column" in the read mapping. Let Y be the sum of all quality scores corresponding to the "column" below X, and let Z be the sum of all quality scores from that column that supported X[1]. Let $Q = Z - (Y - Z)$, then we will assign X the quality score of q where

[1]By supporting a consensus symbol, we understand the following: when conflicts are resolved using voting, then only the reads having the symbol that is eventually called are said to support the consensus. When ambiguity codes

$$q = \begin{cases} 64 & \text{if } Q > 64 \\ 0 & \text{if } Q < 0 \\ Q & \text{otherwise} \end{cases}$$

19.11 Combine Reports

The **Combine Reports** tool makes it easy to get a cross-sample overview by summarizing reports from multiple samples. The tool takes in multiple reports and generates a single report containing summaries and other selected information from the original reports with outliers highlighted.

To create a sample report for a single sample, use the tool **Create Sample Report** described in section 26.5. The individual sample reports, which combine information from multiple report types for a single sample, can then be used as input to the **Combine Reports** tool to generate a combined report that provides a comprehensive overview of results.

Reports produced by the *CLC Genomics Workbench* tools listed in section 26.4.2 can be used as input, as can reports generated by some tools delivered by plugins developed by QIAGEN.

Creating a combined report

To create a combined report, go to:

Toolbox |**Quality Control** ( ) | **Combine Reports** ( )

In the dialog that opens, select the reports to be combined (figure 26.26).

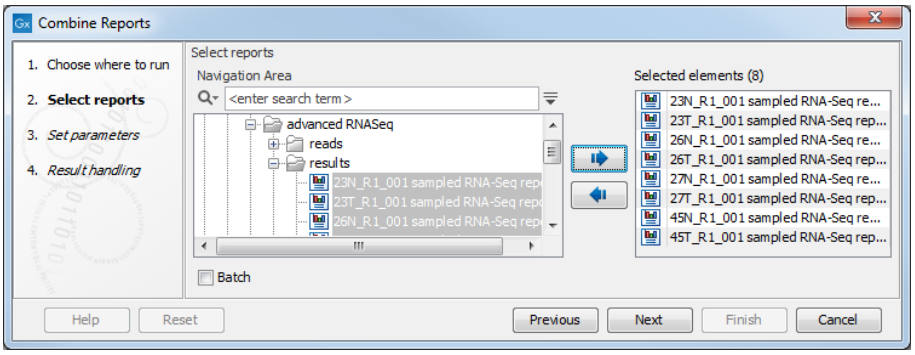


Figure 19.23: *The reports to be combined are selected as input.*

In the next dialog, configuration options are presented, as shown in figure 26.27:

The configuration options are:

- **Show summary information as plots** Enable this to display information as box plots instead of tables, if possible for a given data value.
- **Include tables for outliers** Enable this to add a table after each summary table or box plot, containing the samples that are outliers for that data type, if possible for a given data value.

  See section 26.4.1 for further details.

are used instead, all reads contribute to the called consensus and thus $Y = Z$.

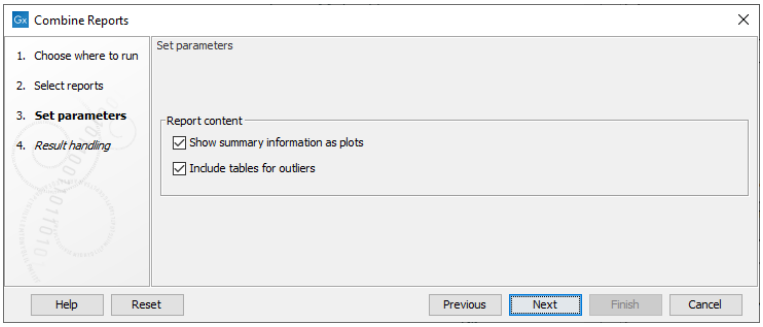


Figure 19.24: *Configuration options for combining reports.*

19.11.1 Combine Reports output

The **Combine Reports** tool generates a single report, ( ), containing summaries and other selected information from the reports provided as input.

The report is divided into one or more sections depending on the input report type.

By default, summary information is displayed in table format. Each table presents a summary of the corresponding information from the input reports.

The tables will contain one row per input report, the first column indicating the sample same taken from the corresponding input report, and additional rows, shaded in pale gray, which report the minimum, median, maximum, mean and standard deviation for all numeric columns (figure 26.28).

If the **Show summary information as plots** option was enabled, tables will be displayed as box plots, wherever possible. Each numeric column will be presented as one box in the plot (figure 26.28).

Combined reports contain only data from report types supported by the **Combined Reports** tool. Where some of the reports supplied as inputs are supported and some are not, the combined report will contain information only from the supported ones. Supported report types are listed in section 26.4.2. If none of the reports provided are of supported types, the report will contain a statement saying this.

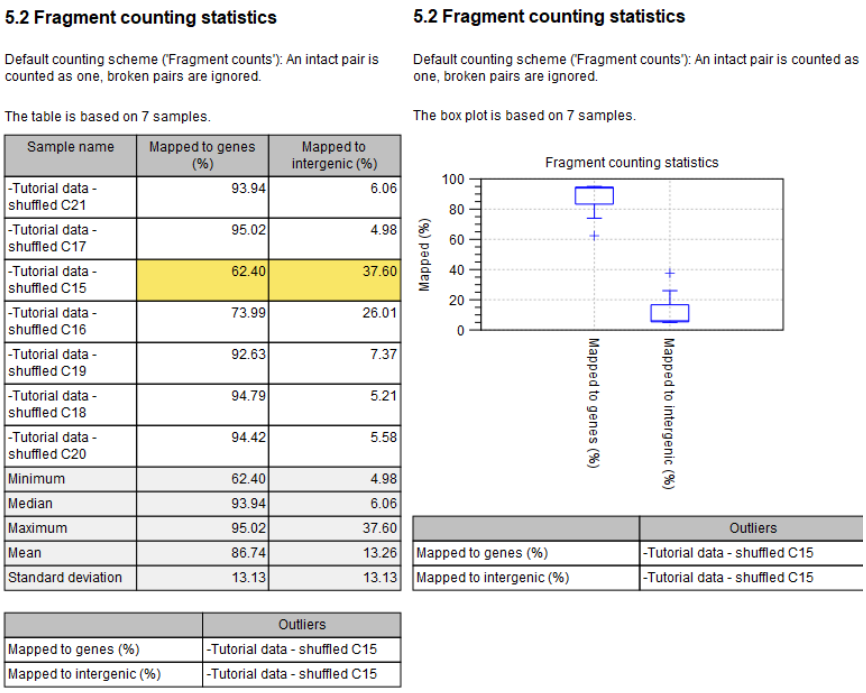
Outliers and other highlighted entries in combined reports

Table cells are colored if the data they contain is considered an outlier or problematic.

Cells containing outliers are highlighted in yellow. Outliers are those outside the range: lower quartile - 1.5 IQR, upper quartile + 1.5 IQR.

If the **Include tables for outliers** option was enabled, any table or plot containing outliers will be followed by an additional table containing the names of the outliers. A row is provided for each column containing outliers (figure 26.28).

Cells highlighted in pink indicate that a problem has been identified. This is explained underneath the table for each column that contains pink cells. If a cell is both an outlier and problematic, it will be highlighted in red (figure 26.29).



5.2 Fragment counting statistics

Default counting scheme ('Fragment counts'): An intact pair is counted as one, broken pairs are ignored.

The table is based on 7 samples.

Sample name	Mapped to genes (%)	Mapped to intergenic (%)
-Tutorial data - shuffled C21	93.94	6.06
-Tutorial data - shuffled C17	95.02	4.98
-Tutorial data - shuffled C15	62.40	37.60
-Tutorial data - shuffled C16	73.99	26.01
-Tutorial data - shuffled C19	92.63	7.37
-Tutorial data - shuffled C18	94.79	5.21
-Tutorial data - shuffled C20	94.42	5.58
Minimum	62.40	4.98
Median	93.94	6.06
Maximum	95.02	37.60
Mean	86.74	13.26
Standard deviation	13.13	13.13

	Outliers
Mapped to genes (%)	-Tutorial data - shuffled C15
Mapped to intergenic (%)	-Tutorial data - shuffled C15

5.2 Fragment counting statistics

Default counting scheme ('Fragment counts'): An intact pair is counted as one, broken pairs are ignored.

The box plot is based on 7 samples.

	Outliers
Mapped to genes (%)	-Tutorial data - shuffled C15
Mapped to intergenic (%)	-Tutorial data - shuffled C15

Figure 19.25: *By default, summary information is reported in table format (left). Tables of numerical data in cross-sample reports contain summary rows, shaded in pale gray. Cells containing values identifed as outliers are highlighted in yellow. Box plots are generated instead of tables for numerical data if the "Show summary information as plots" option was enabled (right). The tables at the bottom, containing summary information about outliers, are present because the reports were generated with the "Include tables for outliers" option enabled.*

5.4 Strand specificity

The table is based on 7 samples.

Sample name	Strand specific setting	Forward reads mapped (%)	Reverse reads mapped (%)	Ignored reads (wrong strand) (%)
-Tutorial data - shuffled C21	Forward	100	0	5.83
-Tutorial data - shuffled C17	Forward	100	0	4.01
-Tutorial data - shuffled C15	Forward	100	0	39.39
-Tutorial data - shuffled C16	Forward	100	0	25.46
-Tutorial data - shuffled C19	Forward	100	0	4.05
-Tutorial data - shuffled C18	Forward	100	0	6.27
-Tutorial data - shuffled C20	Forward	100	0	6.24
Minimum	-	100.00	0.00	4.01
Median	-	100.00	0.00	6.24
Maximum	-	100.00	0.00	39.39
Mean	-	100.00	0.00	13.04
Standard deviation	-	0.00	0.00	13.88

Ignored reads (wrong strand) (%): > 25% of reads were filtered away due to the strand specific setting.
- If a strand-specific protocol has not been used, re-run the tool with strand specific setting "Both".
- If a strand-specific protocol has been used, library preparation may have failed.

	Outliers
Ignored reads (wrong strand) (%)	-Tutorial data - shuffled C15

Figure 19.26: *Table with both problematic cells and outliers. The table is followed by an explanation for the column "Ignored reads (wrong strand) (%)" containing problematic cells.*

Chapter 20

Cutting and cloning

Contents

20.1 Restriction site analyses 465
20.1.1 Dynamic restriction sites 466
20.1.2 Restriction Site Analysis 471
20.2 Restriction enzyme lists 474
20.3 Molecular cloning 475
20.3.1 Introduction to the cloning editor 476
20.3.2 The cloning workflow 477
20.3.3 Manual cloning 479
20.3.4 Insert restriction site 484
20.4 Gateway cloning 484
20.4.1 Add attB sites 485
20.4.2 Create entry clones (BP) 488
20.4.3 Create expression clones (LR) 489
20.5 Gel electrophoresis 491
20.5.1 Gel view 491

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.

20.1 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.
- In the **Toolbox** you will find the Cloning and Restriction Sites tool that provides more control on the analysis, and gives you 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.

20.1.1 Dynamic restriction sites

If you open a sequence, a sequence list etc, you will find a **Restriction Sites** section in the Side Panel.

Restriction sites can be shown on the sequence as colored triangles and lines (figure 20.1): check the "Show" option on top of the Restriction sites section, then specify the enzymes that should be displayed.



Figure 20.1: *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 name flags** above the list of restriction enzymes.

There is also an option to specify how the **Labels** shown 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 20.2). 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.

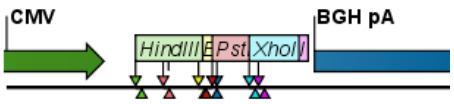


Figure 20.2: *Restriction site labels shown as flags.*

- **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 20.3).

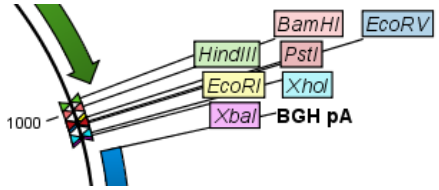


Figure 20.3: *Restriction site labels in radial layout.*

- **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 20.4).

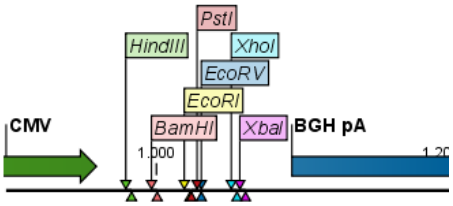


Figure 20.4: *Restriction site labels stacked.*

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 20.5).

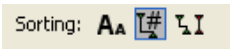


Figure 20.5: *Buttons to sort restriction enzymes.*

- **Sort enzymes alphabetically** (). Clicking this button will sort the list of enzymes alphabetically.
- **Sort enzymes by number of restriction sites** (). 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 20.6.

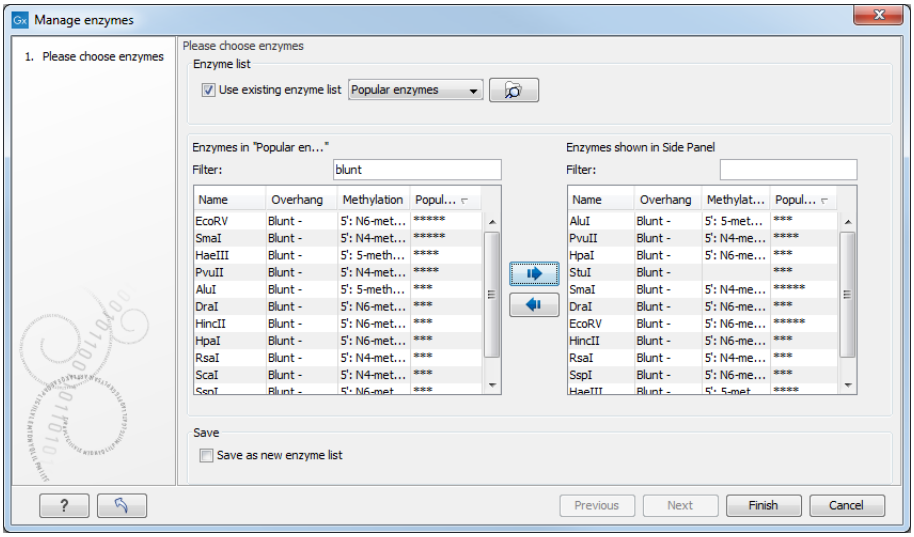


Figure 20.6: *Adding or removing enzymes from the Side Panel.*

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 20.7). You can also open a view of an enzyme list saved in the Navigation Area.

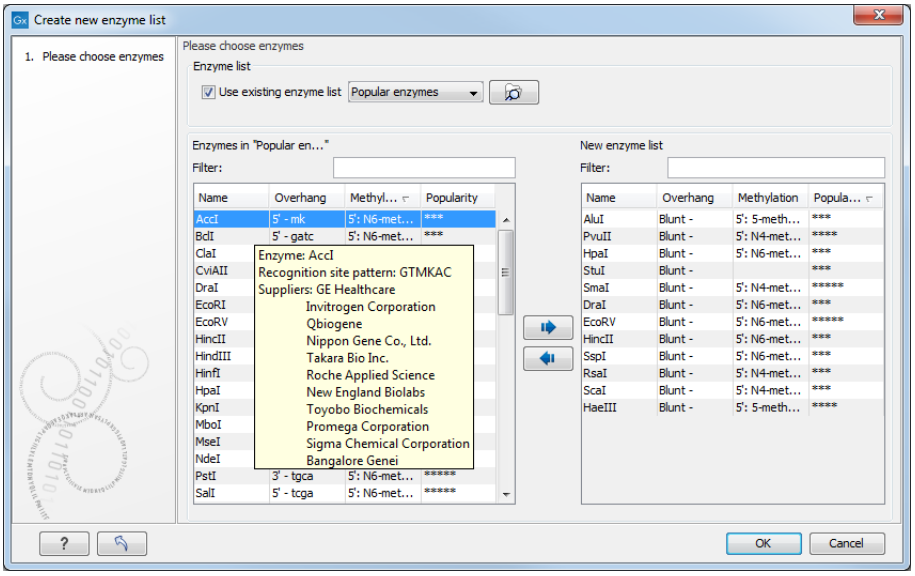


Figure 20.7: *Showing additional information about an enzyme like recognition sequence or a list of commercial vendors.*

At the bottom of the dialog, you can select to save the updated list of enzymes as a new file. When you click **Finish**, the enzymes are added to the Side Panel and the cut sites are shown on the sequence. If you have specified a set of enzymes which you always use, it will probably be a good idea to save the settings in the Side Panel (see section 4.6) for future use.

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 20.1.1). 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 20.8), you can define the terms of your search.

At the top of the dialog, you see the selected region, and below are two panels:

- **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

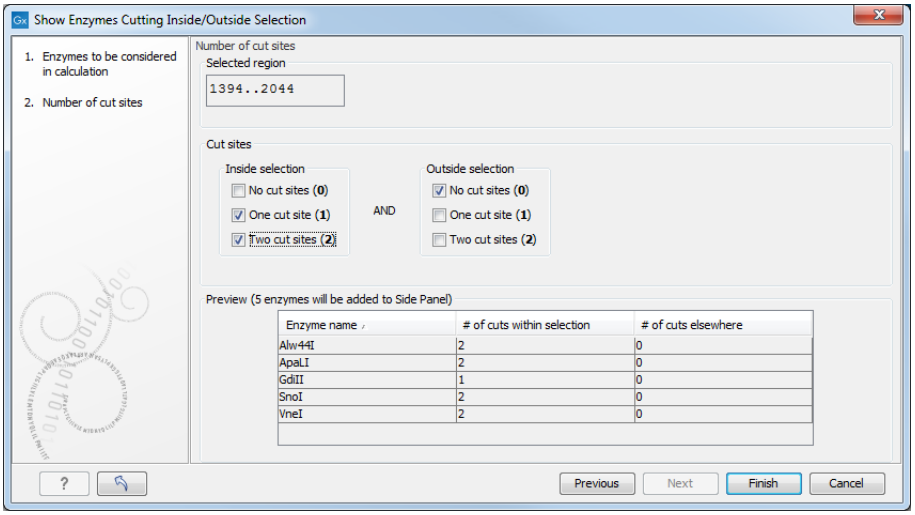


Figure 20.8: *Deciding number of cut sites inside and outside the selection.*

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 20.9).

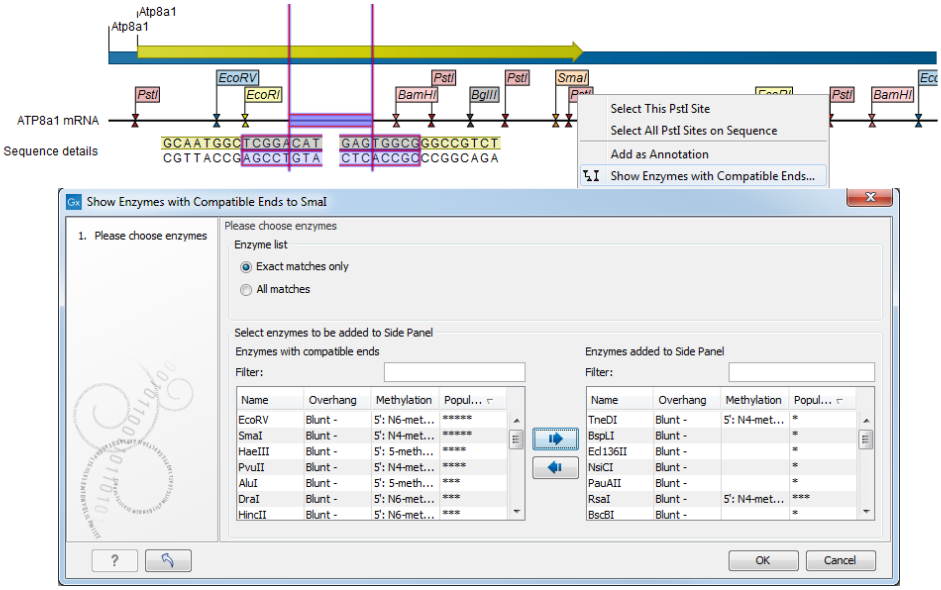


Figure 20.9: *Enzymes with compatible ends.*

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 Side Panel and their cut sites displayed on the sequence.

20.1.2 Restriction Site Analysis

Besides the dynamic restriction sites, you can do a more elaborate restriction map analysis with more output format using the Toolbox:

> **Toolbox | Molecular Biology Tools ( ) | Cloning and Restriction Sites ( )| Restriction Site Analysis ( )**

You first specify which sequence should be used for the analysis. Then define which enzymes to use as basis for finding restriction sites on the sequence (see section 20.1.1).

In the next dialog, you can use the checkboxes so that the output of the restriction map analysis only include restriction enzymes which cut the sequence a specific number of times (figure 20.10).

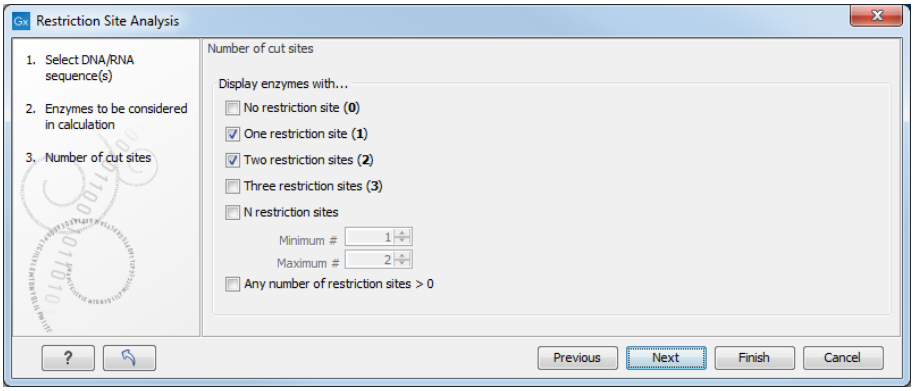


Figure 20.10: *Selecting number of cut sites.*

The default setting is to include the enzymes which cut the sequence one or two times, but you can use the checkboxes to perform very specific searches for restriction sites, for example to find enzymes which do not cut the sequence, or enzymes cutting exactly twice.

The Result handling dialog (figure 20.11) lets you specify how the result of the restriction map analysis should be presented.

Add restriction sites as annotations to sequence(s) . This option makes it possible to see the restriction sites on the sequence (see figure 20.12) and save the annotations for later use.

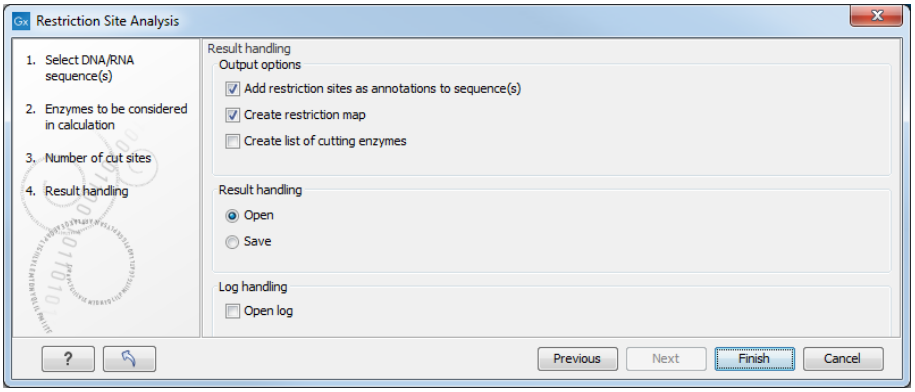


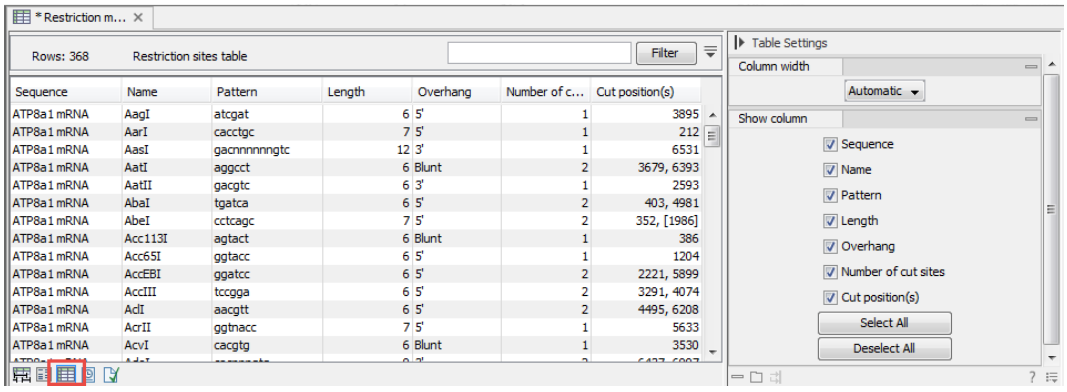
Figure 20.11: *Choosing to add restriction sites as annotations or creating a restriction map.*



Figure 20.12: *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 20.13. 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	gacnnnnnngtc	12	3'	1	6531
ATP8a1 mRNA	AatI	aggcct	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	ggtacc	6	5'	1	1204
ATP8a1 mRNA	AccEBI	ggatcc	6	5'	2	2221, 5899
ATP8a1 mRNA	AccIII	tccgga	6	5'	2	3291, 4074
ATP8a1 mRNA	AclI	aacgtt	6	5'	2	4495, 6208
ATP8a1 mRNA	AcrII	ggtnacc	7	5'	1	5633
ATP8a1 mRNA	AcvI	cacgtg	6	Blunt	1	3530

Figure 20.13: *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 figure20.14). Click the Fragments button ( ) at the bottom of the view.

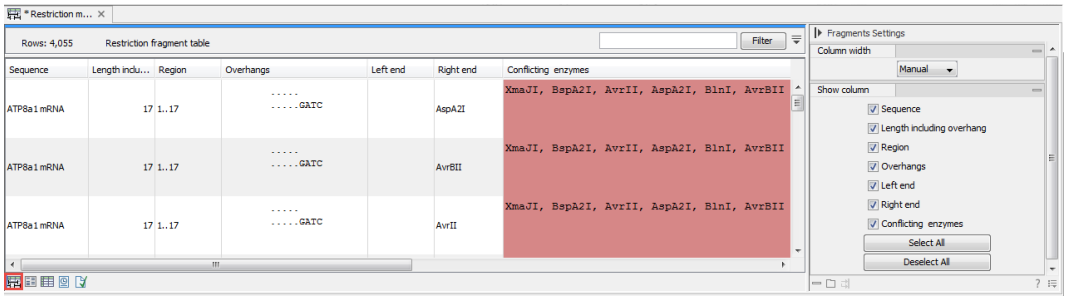


Figure 20.14: *The result of the restriction analysis shown as table of fragments.*

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 20.40). For more information about gel electrophoresis, see section 20.5.

20.2 Restriction enzyme lists

CLC Genomics Workbench includes all the 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 form 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 20.15

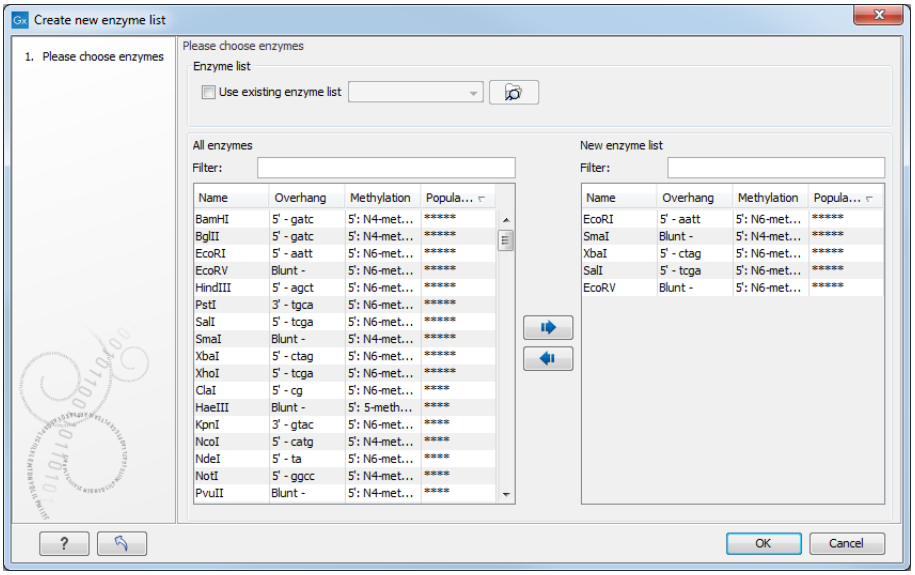


Figure 20.15: *Choosing enzymes for the new enzyme list.*

Choose which enzyme you want to include in the new enzyme list (see section 20.1.1), and click **Finish** to open the enzyme list.

View and modify enzyme list An enzyme list is shown in figure 20.16. 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.

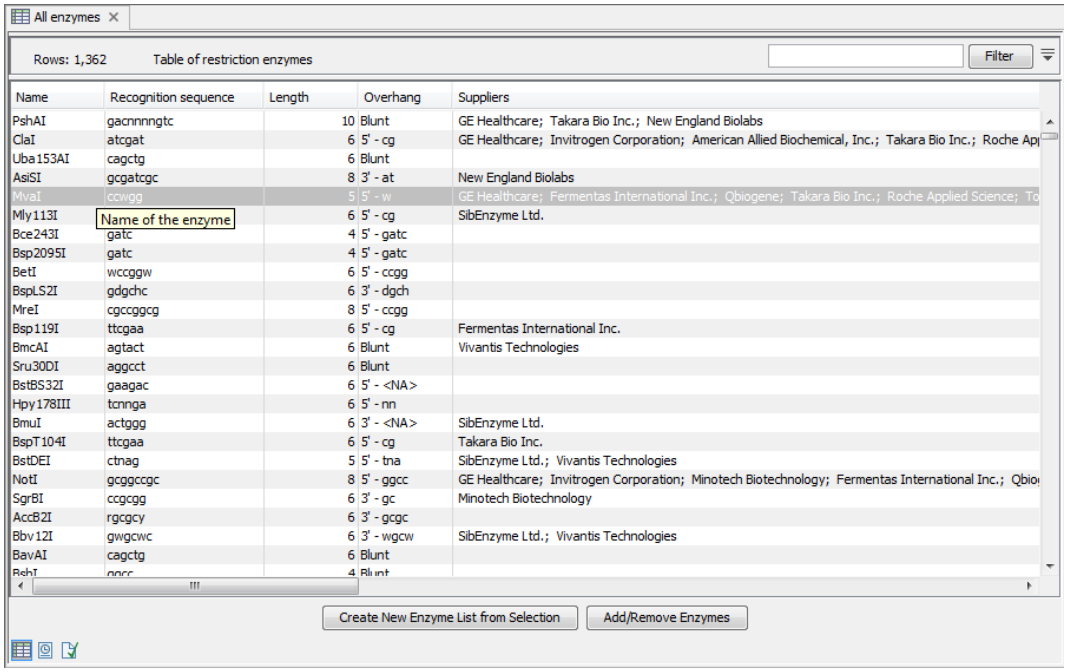


Figure 20.16: *An enzyme list.*

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 20.15 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 distributor, type the name of the distributor into the filter and select and create a new enzyme list from the selection.

20.3 Molecular cloning

The *in silico* cloning process in *CLC Genomics Workbench* begins with the Cloning tool:

Molecular Biology Tools ( ) | Cloning and Restriction Sites ( )| 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 20.17).

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 20.18).

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".

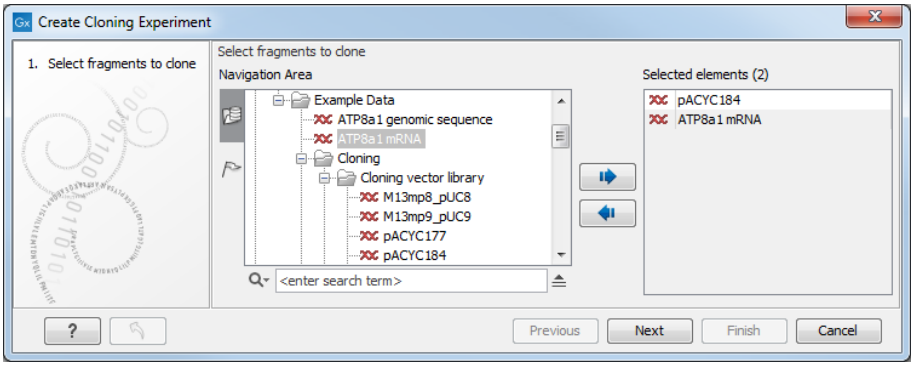


Figure 20.17: *Selecting the sequences containing the fragments you want to clone and the vector.*

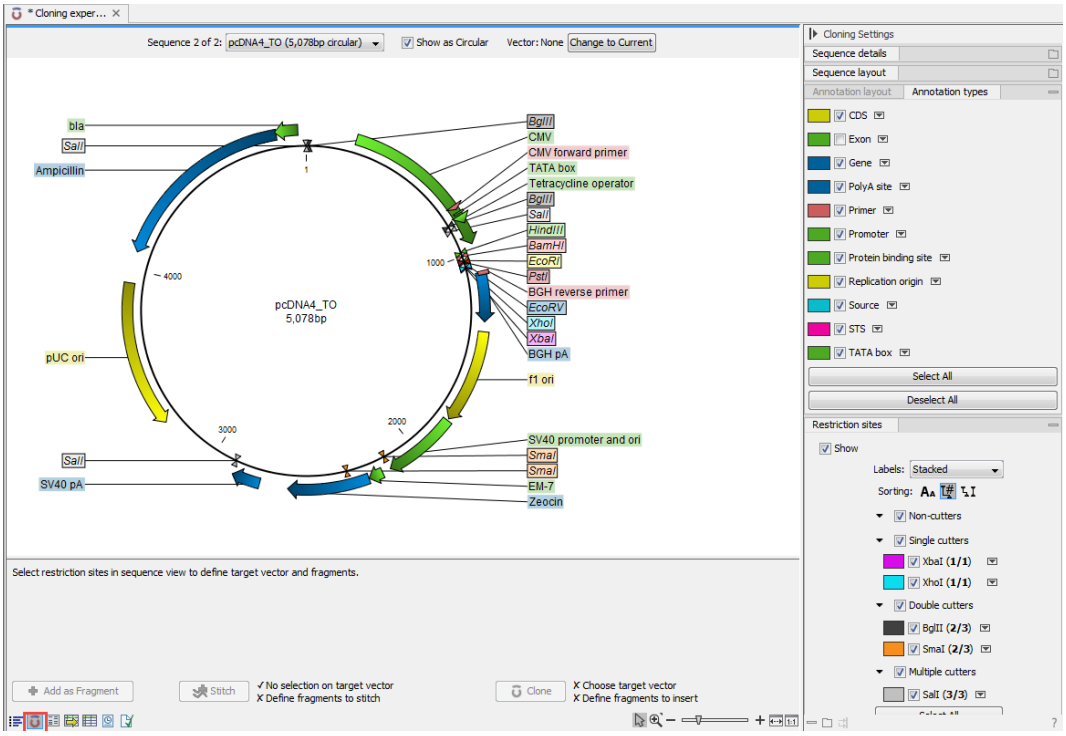


Figure 20.18: *Cloning editor view of the sequence list. Choose which sequence to display from the drop down menu.*

20.3.1 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 12). 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 activated in different ways. One way is to click on the **Cloning Editor** icon ( ) in the view area when a sequence list has been opened in the sequence list editor. Another way is to create a new cloning experiment (the actual data object will still be a sequence list) using the **Cloning** ( ) action from the toolbox. Using this action the user collects a set of existing sequences and creates a new sequence list.

The cloning editor can be used in two different ways:

- **The cloning mode**, when the user has selected one of the sequences as 'Vector'. In the cloning mode, the user opens up the vector by applying 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 user can switch the order of the inserted fragments and rotate them prior to adjusting the overhangs to match the cloning conditions.
- **The stitch mode**, when the user has not selected a sequence as 'Vector'. In stitch mode, the user can select a number of fragments (either full sequences or cuttings) from the cloning experiment. These fragments can then be stitched together into one single new and longer sequence. In the stitching adapter dialog, the user can switch order and rotate the fragments prior to adjusting the overhangs to match the stitch conditions.

20.3.2 The cloning workflow

The *cloning workflow* is designed to support restriction cloning workflows through 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 20.1.1). 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 20.19).

   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.

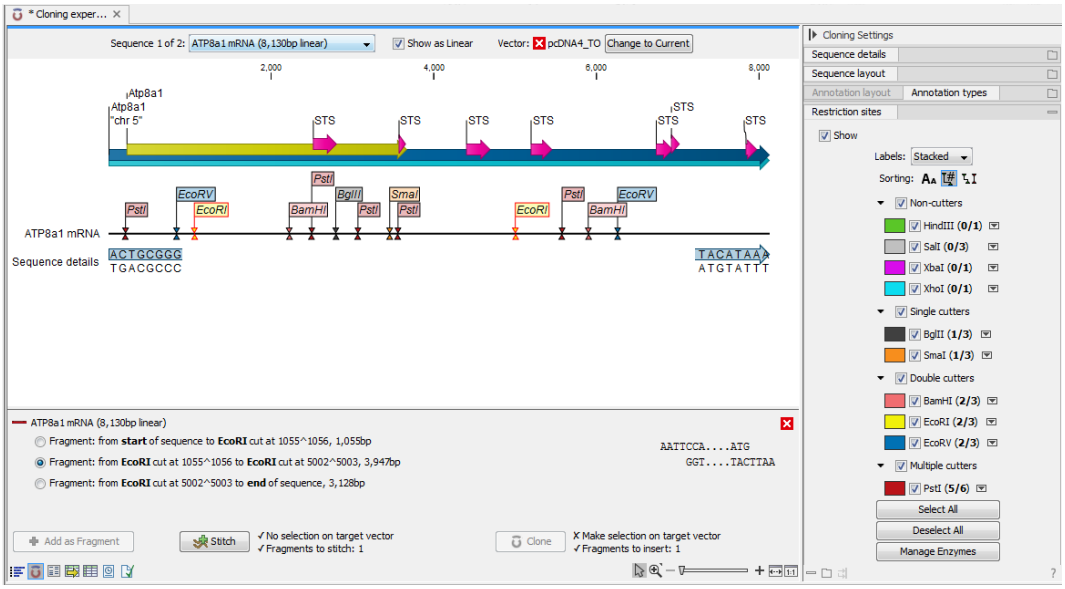


Figure 20.19: *EcoRI cut sites selected to cut out fragment.*

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 20.20).

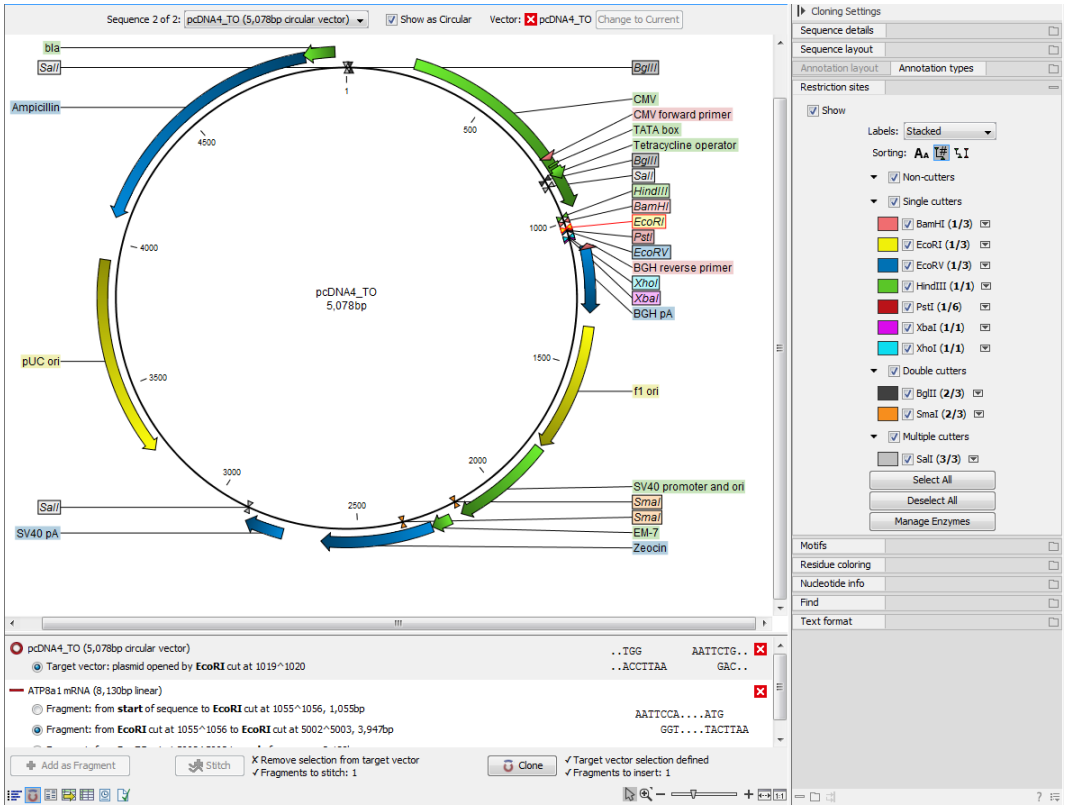


Figure 20.20: *EcoRI site used to open the vector. Note that the "Cloning" button has now been enabled as both criteria ("Target vector selection defined" and "Fragments to insert:...") have been defined.*

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 20.21)

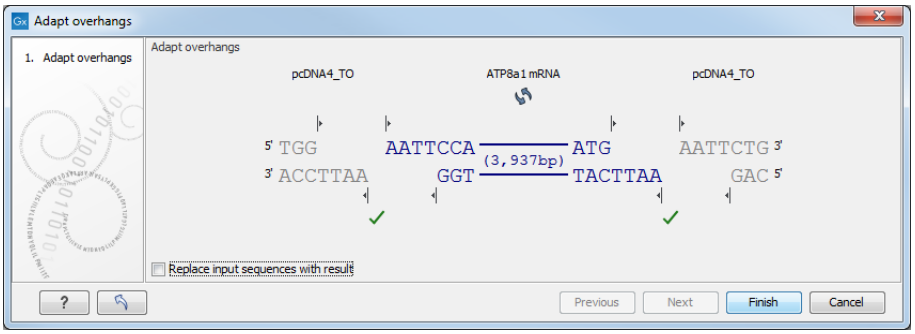


Figure 20.21: *Showing the insertion point of the vector.*

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.

20.3.3 Manual cloning

If you wish to use the manual way of cloning, you still create a sequence list with the Cloning tool, but can skip the "Perform cloning" step of the cloning workflow explained in section 20.3.2. Instead, all manipulations of sequences are done manually, using right-click menus. These menus have two different appearances depending on where you click, as visualized in figure 20.22.

Manipulate the whole sequence

Right-click the sequence label to the left to see the menu shown in figure 20.23.

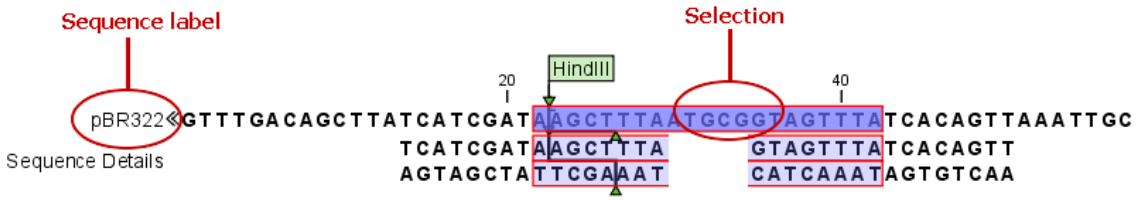


Figure 20.22: *The red circles mark the two places you can use for manipulating the sequences.*

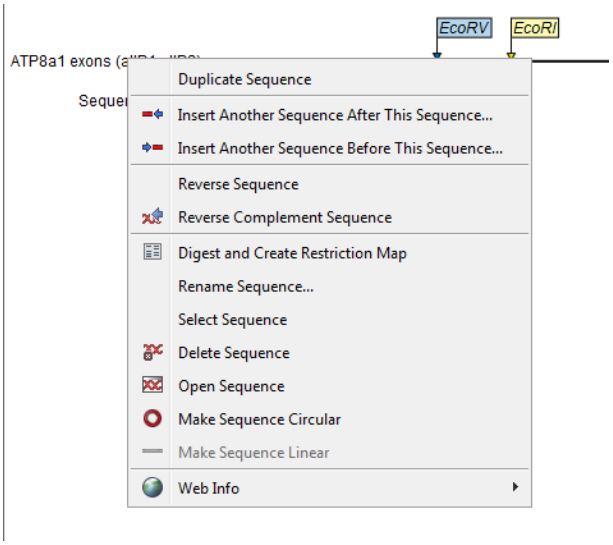


Figure 20.23: *Right click on the sequence in the cloning view.*

- **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 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 20.5
- **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 20.24.

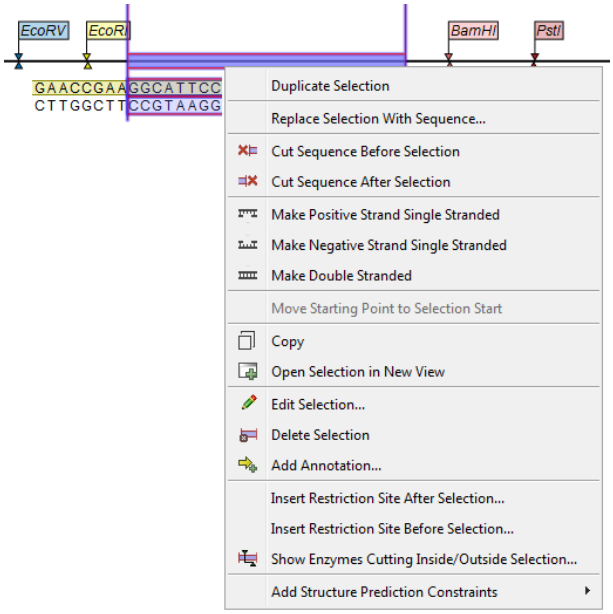


Figure 20.24: *Right click on a sequence selection in the cloning view.*

- **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 is it 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 20.3.4).
- **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 20.25).

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 20.3.2.

Furthermore, t he 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).

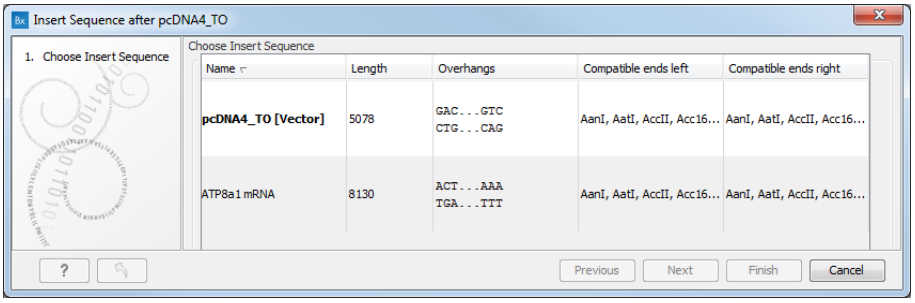


Figure 20.25: *Select a sequence for insertion.*

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.

Select the sequence you wish to insert and click **Next** to adapt insert sequence to vector dialog (figure 20.26).

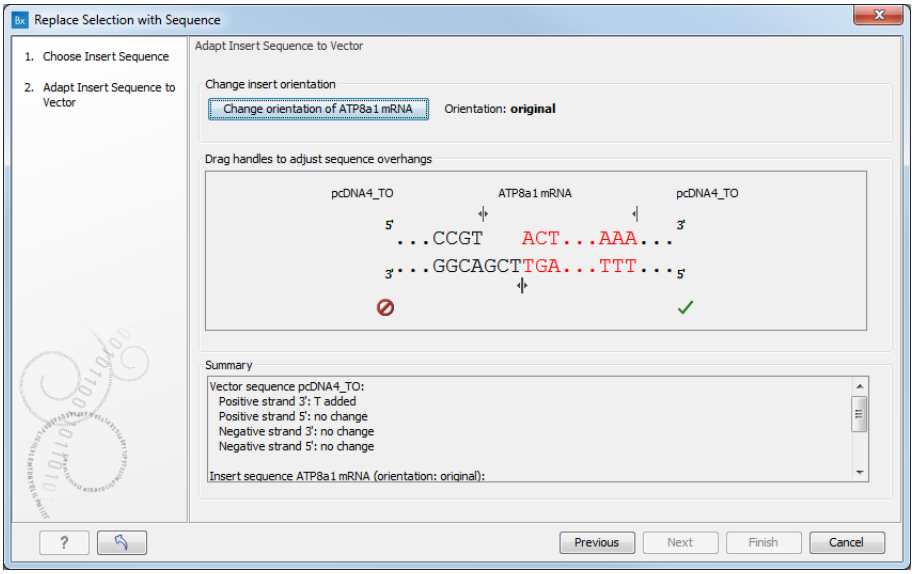


Figure 20.26: *Drag the handles to adjust overhangs.*

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.

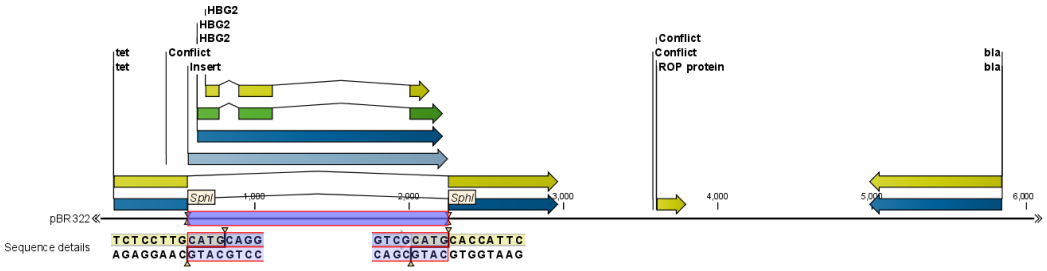


Figure 20.27: *One sequence is now inserted into the cloning vector. The sequence inserted is automatically selected.*

20.3.4 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 20.28.

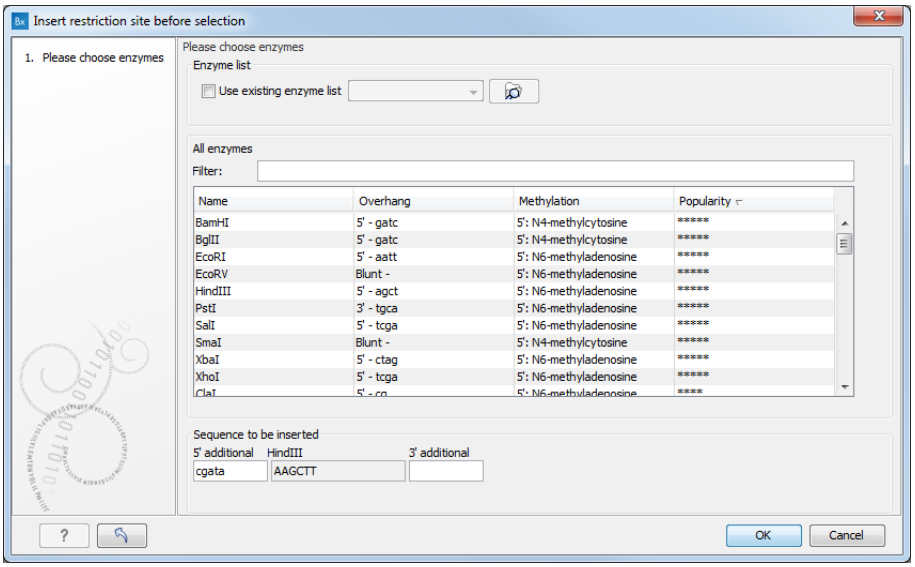


Figure 20.28: *Inserting a restriction site and potentially a recognition sequence.*

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.

20.4 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 http://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.

20.4.1 Add attB sites

The first step in the Gateway cloning process is to amplify the target sequence with primers including so-called attB sites:

Toolbox | **Molecular Biology Tools** ( ) | **Cloning and Restriction Sites** ( )| **Gateway Cloning** ( ) | **Add attB Sites** ( )

This will open a dialog where you can select one or more sequences. Note that if your 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 20.29.

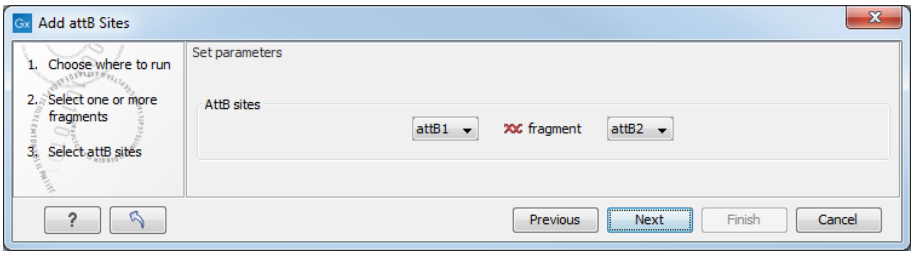


Figure 20.29: *Selecting which attB sites to add.*

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 20.30). 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 20.4.1.

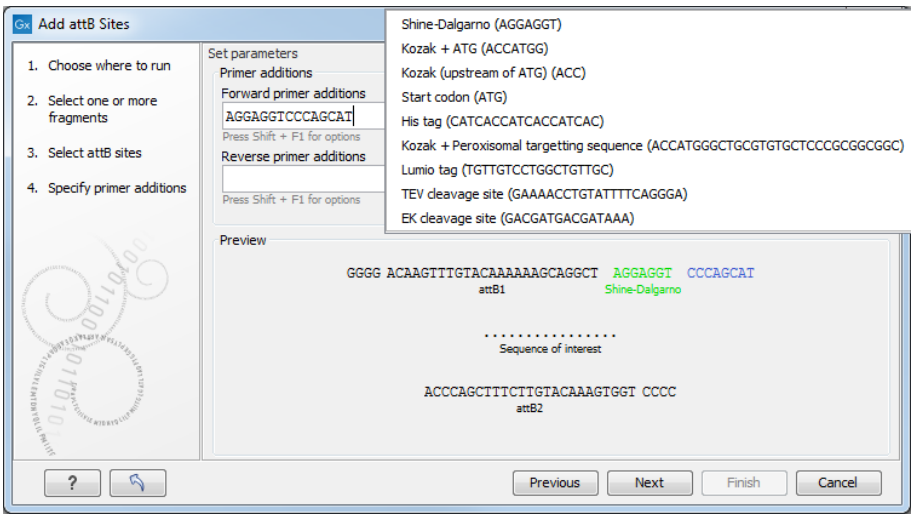


Figure 20.30: *Primer additions 5' of the template-specific part of the primer where a Shine-Dalgarno site has been added between the attB site and the gene of interest.*

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 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 20.31.

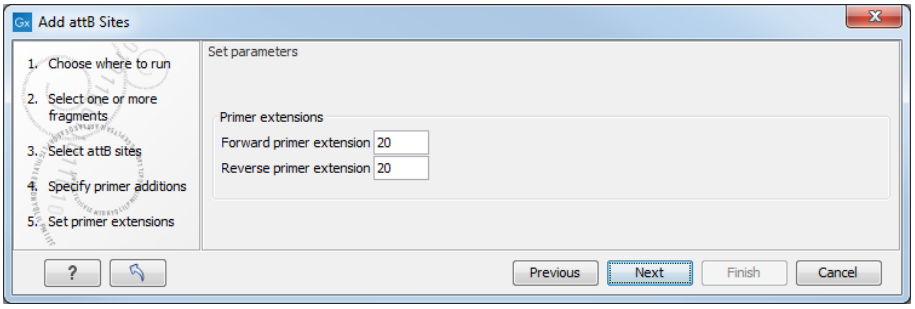


Figure 20.31: *Specifying the length of the template-specific part of the primers.*

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 20.32).

The attB sites, the primer additions and the primer regions are annotated in the final result as shown in figure 20.33 (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

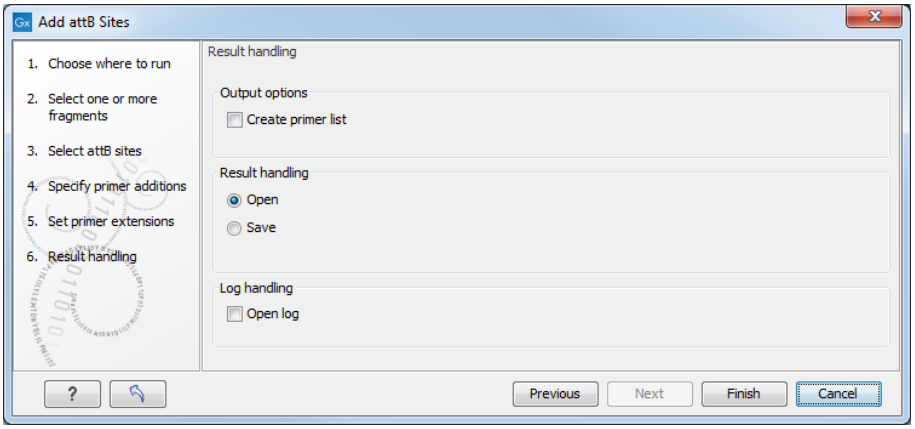


Figure 20.32: *Besides the main output which is a copy of the input sequence(s) now including attB sites and primer additions, you can get a list of primers as output.*

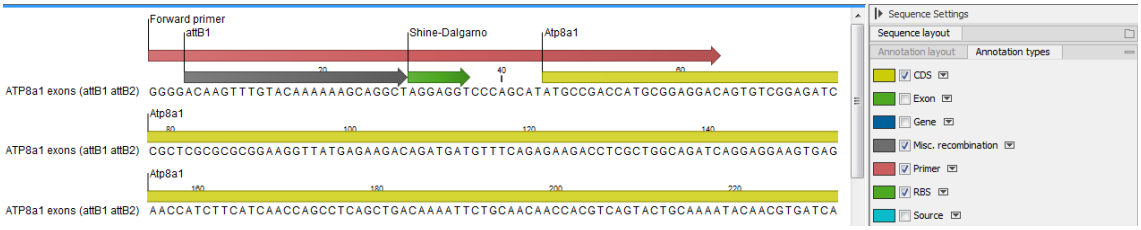


Figure 20.33: *the attB site plus the Shine-Dalgarno primer addition is annotated.*

workflow (see section 20.4.2).

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 20.30 can be configured and extended. If there is a tag that you use a lot, you 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 20.34), 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.

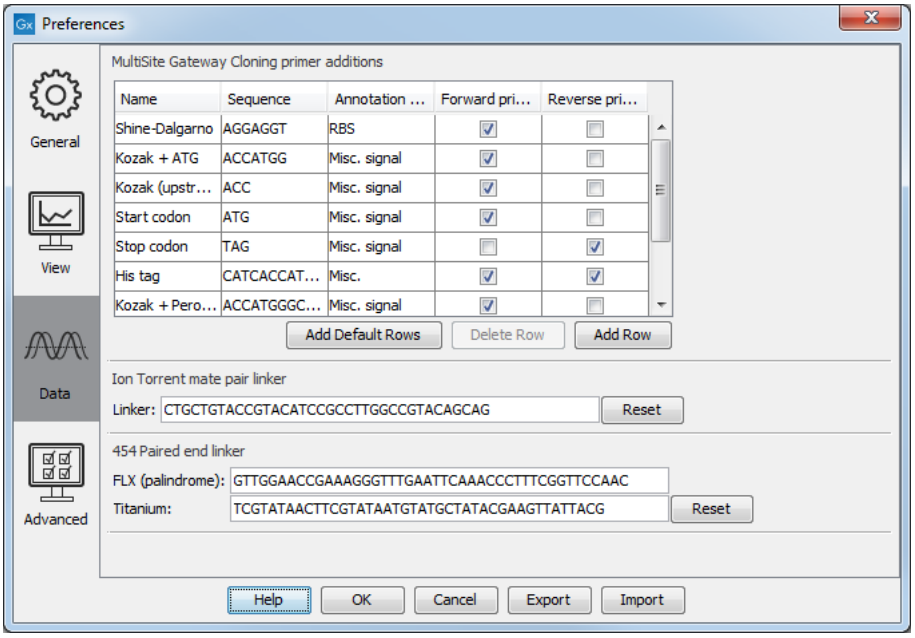


Figure 20.34: *Configuring the list of primer additions available when adding attB sites.*

Reverse primer addition Whether this addition should be visible in the list of additions for the reverse primer.

20.4.2 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 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.

Toolbox | Molecular Biology Tools ( ) | Cloning and Restriction Sites ( )| 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 20.4.1). You can select more than one sequence as input, and the corresponding number of entry clones will be created.

In the following dialog (figure 20.35), you can specify a donor vector.

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

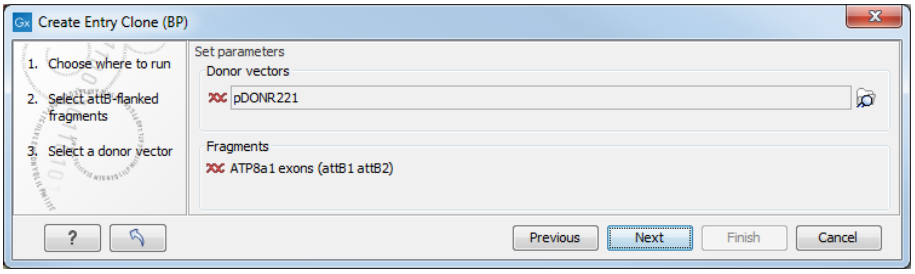


Figure 20.35: *Selecting one or more donor vectors.*

the recombination, and the entry clone is now equipped with attL sites as shown in figure 20.36.

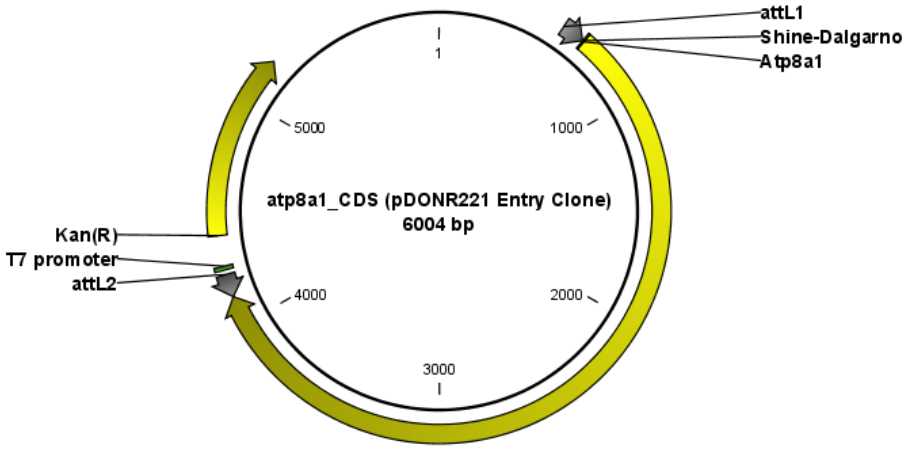


Figure 20.36: *The resulting entry vector opened in a circular view.*

Note that the bi-product of the recombination is not part of the output.

20.4.3 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
ATTGACACGCCCGGGCGACGGATGGTGATCCCCCTGGCCAGTGCACGTCTGCTGTCAGATAAAGTCTCC
CGTGAACTTTACCCGGTGGTGCATATCGGGGATGAAAGCTGGCGCATGATGACCACCGATATGGCCAGT
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:

Toolbox | **Molecular Biology Tools ( )** | **Cloning and Restriction Sites ( )** | **Gateway Cloning ( )** | **Create Expression Clone ( )**

In the first step, select one or more entry clones (see how to create an entry clone in section 20.4.2). If you wish to perform separate LR reactions with multiple entry clones, you should run the **Create Expression Clone** in batch mode (see section 9.3).

In the second step, select the destination vector that was previously saved in the Navigation Area (fig 20.37).

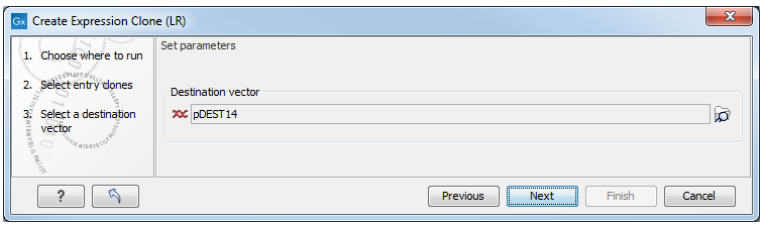


Figure 20.37: *Selecting one or more destination vectors.*

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 20.38.

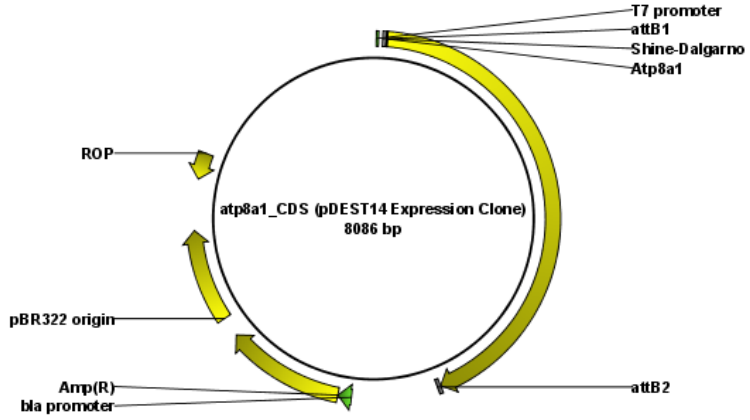


Figure 20.38: *The resulting expression clone opened in a circular view.*

You can choose to create a sequence list with the bi-products as well.

20.5 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 performing the **Restriction Site Analysis** from the Toolbox, you can choose to create a restriction map which can be shown as a gel (see section 20.1.2).
- 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 12.2).
  - Ordinary sequence view (see section 12).
  - Graphical view of sequence lists (see section 12.6).
  - Cloning editor (see section 20.3).
  - Primer designer (see section 18.3).
- **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 20.39).

20.5.1 Gel view

In figure 20.40 you can see a simulation of a gel with its Side Panel to the right.

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.

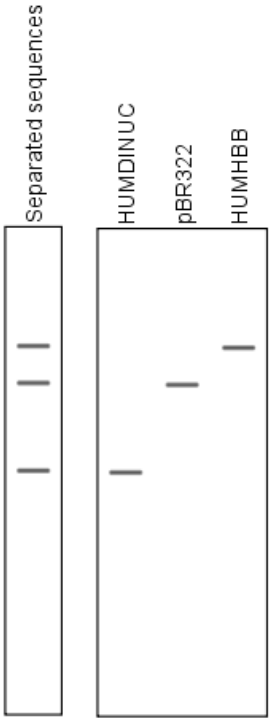


Figure 20.39: *A sequence list shown as a gel.*

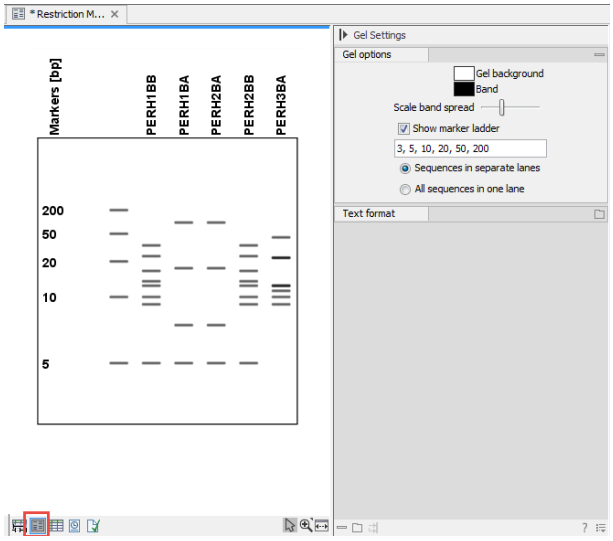


Figure 20.40: *Five lanes showing fragments of five sequences cut with restriction enzymes.*

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 21

Sequence alignment

Contents

21.1 Create an alignment 494
21.1.1 Gap costs 495
21.1.2 Fast or accurate alignment algorithm 496
21.1.3 Aligning alignments 497
21.1.4 Fixpoints 497
21.2 View alignments 499
21.2.1 Bioinformatics explained: Sequence logo 502
21.3 Edit alignments 503
21.3.1 Realignment 505
21.4 Join alignments 507
21.5 Pairwise comparison 509
21.5.1 The pairwise comparison table 510
21.5.2 Bioinformatics explained: Multiple alignments 511

CLC Genomics Workbench can align nucleotides and proteins using a *progressive alignment* algorithm (see section 21.5.2.

This chapter describes how to use the program to align sequences, and alignment algorithms in more general terms.

21.1 Create an alignment

Alignments can be created from sequences, sequence lists (see section 12.6), existing alignments and from any combination of the three.

To create an alignment in *CLC Genomics Workbench*:

> **Toolbox** | **Classical Sequence Analysis** ( ) | **Alignments and Trees** ( )| **Create Alignment** ( )

This opens the dialog shown in figure 21.1.

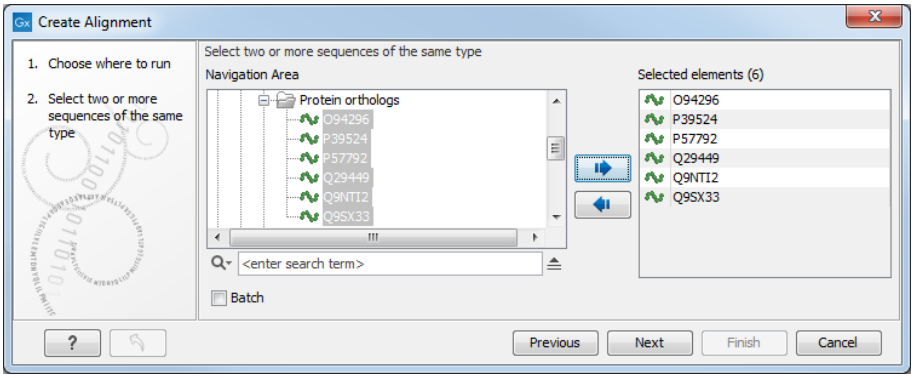


Figure 21.1: *Creating an alignment.*

If you have selected some elements before choosing the Toolbox action, they are now listed in the **Selected Elements** window of the dialog. Use the arrows to add or remove sequences, sequence lists or alignments from the selected elements. Click **Next** to adjust alignment algorithm parameters. Clicking **Next** opens the dialog shown in figure 21.2.

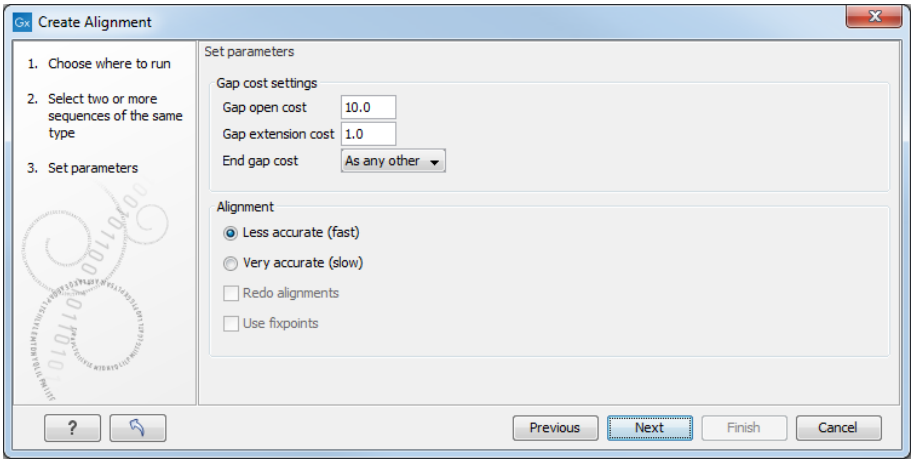


Figure 21.2: *Adjusting alignment algorithm parameters.*

21.1.1 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 21.3 and 21.4 illustrate the differences between the different gap scores at the sequence ends.

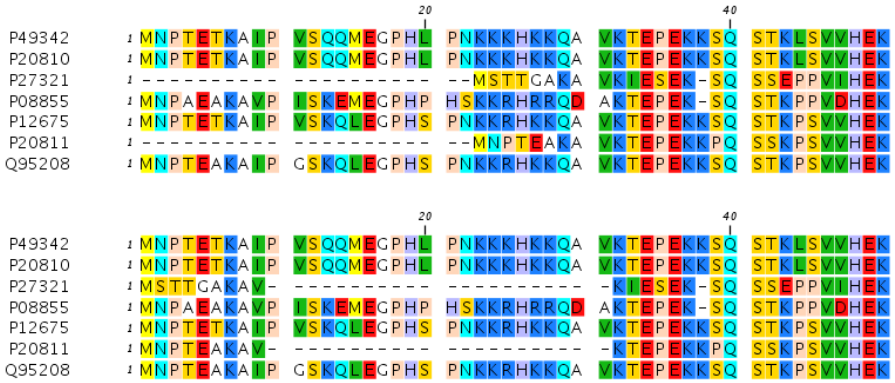


Figure 21.3: *The first 50 positions of two different alignments of seven calpastatin sequences. The top alignment is made with cheap end gaps, while the bottom alignment is made with end gaps having the same price as any other gaps. In this case it seems that the latter scoring scheme gives the best result.*

21.1.2 Fast or accurate alignment algorithm

CLC Genomics Workbench has two algorithms for calculating alignments:

- **Fast (less accurate).** This allows for use of an optimized alignment algorithm which is very fast. The fast option is particularly useful for data sets with very long sequences.
- **Slow (very accurate).** This is the recommended choice unless you find the processing time too long.

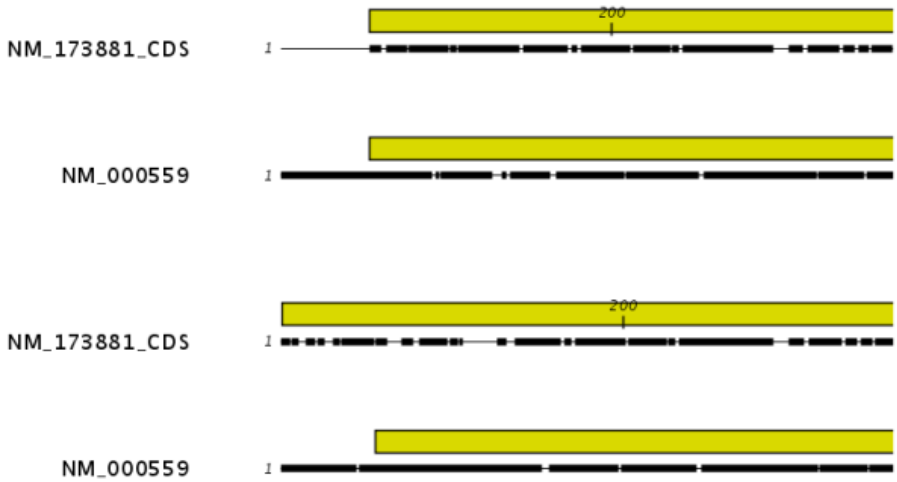


Figure 21.4: *The alignment of the coding sequence of bovine myoglobin with the full mRNA of human gamma globin. The top alignment is made with free end gaps, while the bottom alignment is made with end gaps treated as any other. The yellow annotation is the coding sequence in both sequences. It is evident 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.*

Both algorithms use progressive alignment. The faster algorithm builds the initial tree by doing more approximate pairwise alignments than the slower option.

21.1.3 Aligning alignments

If you have selected an existing alignment in the first step (21.1), you have to decide how this alignment should be treated.

- **Redo alignment.** The original alignment will be realigned if this checkbox is checked. Otherwise, the original alignment is kept in its original form except for possible extra equally sized gaps in all sequences of the original alignment. This is visualized in figure 21.5.

This feature is useful if you wish to add extra sequences to an existing alignment, in which case you just select the alignment and the extra sequences and choose not to redo the alignment.

It is also useful if you have created an alignment where the gaps are not placed correctly. In this case, you can realign the alignment with different gap cost parameters.

21.1.4 Fixpoints

With fixpoints, you can get full control over the alignment algorithm. The fixpoints are points on the sequences that are forced to align to each other.

To add a fixpoint, open the sequence or alignment and:

> **Select the region you want to use as a fixpoint | right-click the selection | Set alignment fixpoint here**

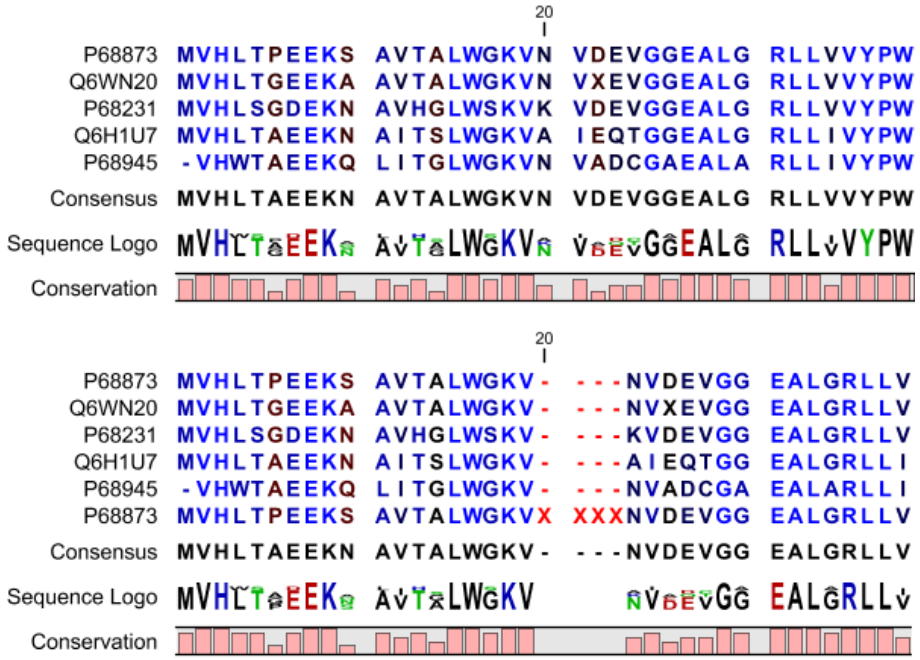


Figure 21.5: *The top figures shows the original alignment. In the bottom panel a single sequence with four inserted X's are aligned to the original alignment. This introduces gaps in all sequences of the original alignment. All other positions in the original alignment are fixed.*

This will add an annotation labeled "Fixpoint" to the sequence (see figure 21.6). Use this procedure to add fixpoints to the other sequence(s) that should be forced to align to each other.



Figure 21.6: *Adding a fixpoint to a sequence in an existing alignment. At the top you can see a fixpoint that has already been added.*

When you click "Create alignment" and go to **Step 2**, check **Use fixpoints** in order to force the alignment algorithm to align the fixpoints in the selected sequences to each other.

In figure 21.7 the result of an alignment using fixpoints is illustrated.

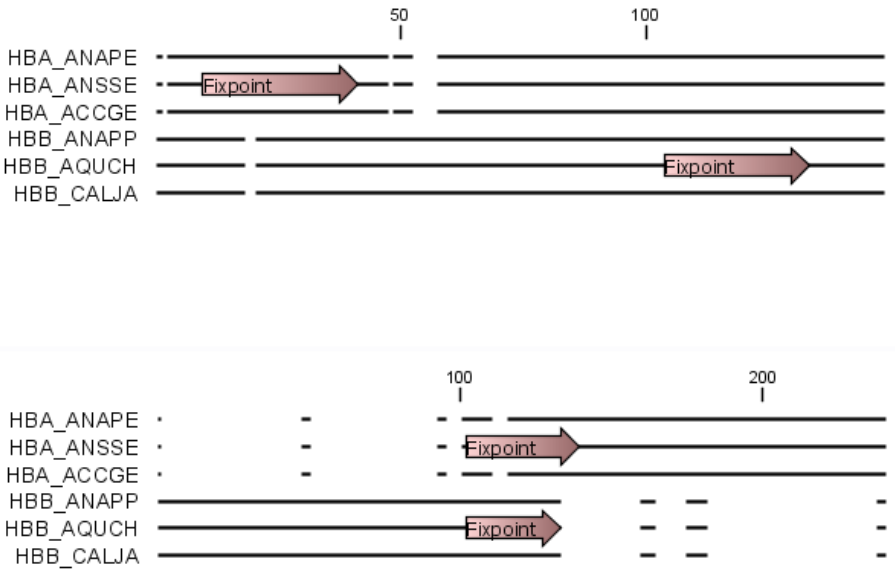


Figure 21.7: *Realigning using fixpoints. In the top view, fixpoints have been added to two of the sequences. In the view below, the alignment has been realigned using the fixpoints. The three top sequences are very similar, and therefore they follow the one sequence (number two from the top) that has a fixpoint.*

You can add multiple fixpoints, e.g. adding two fixpoints to the sequences that are aligned will force their first fixpoints to be aligned to each other, and their second fixpoints will also be aligned to each other.

Advanced use of fixpoints Fixpoints with the same names will be aligned to each other, which gives the opportunity for great control over the alignment process. It is only necessary to change any fixpoint names in very special cases.

One example would be three sequences A, B and C where sequences A and B has one copy of a domain while sequence C has two copies of the domain. You can now force sequence A to align to the first copy and sequence B to align to the second copy of the domains in sequence C. This is done by inserting fixpoints in sequence C for each domain, and naming them 'fp1' and 'fp2' (for example). Now, you can insert a fixpoint in each of sequences A and B, naming them 'fp1' and 'fp2', respectively. Now, when aligning the three sequences using fixpoints, sequence A will align to the first copy of the domain in sequence C, while sequence B would align to the second copy of the domain in sequence C.

You can name fixpoints by:

> **right-click the Fixpoint annotation | Edit Annotation ( ) | type the name in the 'Name' field**

21.2 View alignments

Since an alignment is a display of several sequences arranged in rows, the basic options for viewing alignments are the same as for viewing sequences. Therefore we refer to section 12 for an explanation of these basic options.

However, there are a number of alignment-specific view options in the **Alignment info** and the **Nucleotide info**

in the Side Panel to the right of the view. Below is more information on these view options.

Under **Translation** in the **Nucleotide info**, 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.

The options in the **Alignment info** relate to each column in the alignment.

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. Learn how to export the data behind the graph in section 6.8.

  - **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 (Learn how to export the data behind the graph in section 6.8).
  - **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 21.2.1 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.

21.2.1 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 21.8). 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 21.8). 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 21.8 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}),

$$R_{seq} = S_{max} - S_{obs} = \log_2 N - \left(-\sum_{n=1}^{N} p_n \log_2 p_n \right)$$

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.

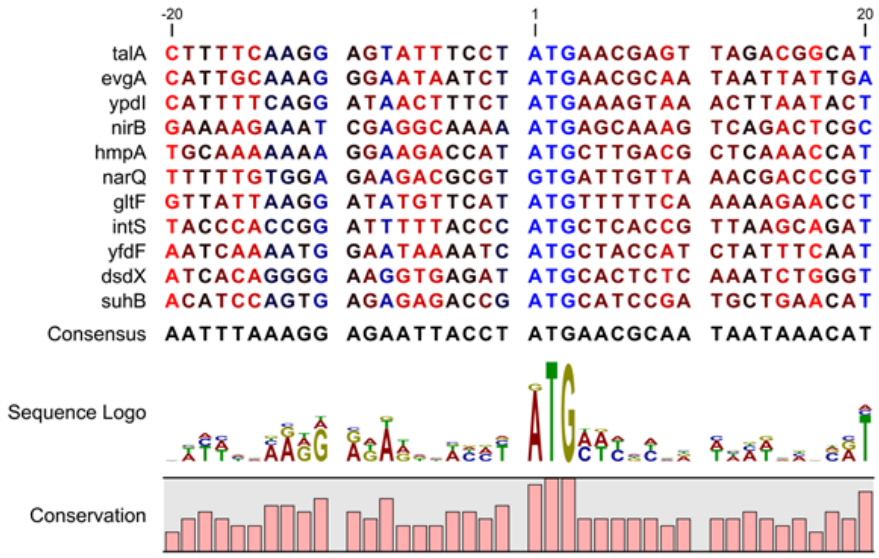


Figure 21.8: *Ungapped sequence alignment of eleven* E. coli *sequences defining a start codon. The start codons start at position 1. Below the alignment is shown the corresponding sequence logo. As seen, a GTG start codon and the usual ATG start codons are present in the alignment. This can also be visualized in the logo at position 1.*

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]

21.3 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 21.1). 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 21.9).

Note! Residues can only be moved when they are next to a gap.

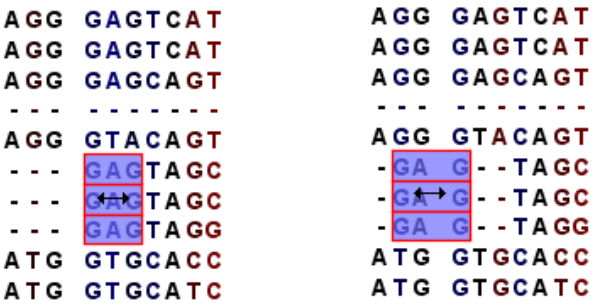


Figure 21.9: *Moving a part of an alignment. Notice the change of mouse pointer to a horizontal arrow.*

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 21.1).

The same procedure can be used for joining two alignments.

21.3.1 Realignment

Realigning a section of an alignment

If you have created an alignment, it is possible to realign a part of it, leaving the rest of the

alignment unchanged:

> **Select a part of the alignment to realign | 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 21.10).

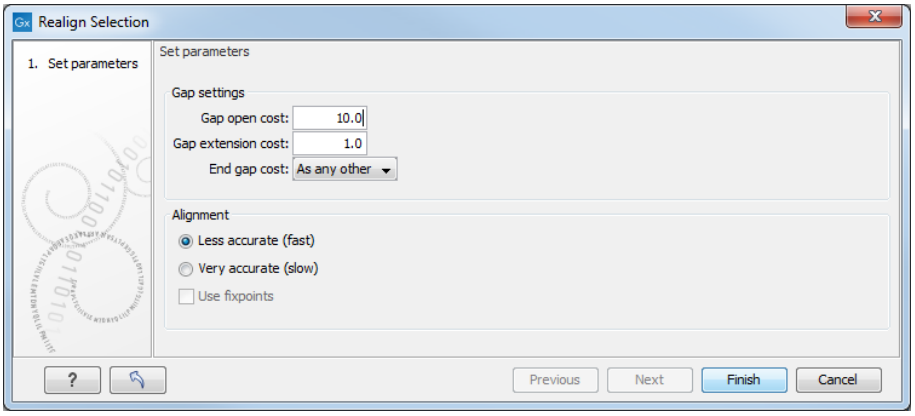


Figure 21.10: *Realigning a section of an alignment.*

Learn more about the options available to you in the sections 21.1.1 and 21.1.2.

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.

Realigning a selection is a very powerful tool for editing alignments in several situations:

- **Removing changes.** If you change the alignment in a specific region manually but decide to undo all the edits you just made, you can easily select the region that was edited and realign it.
- **Adjusting the number of gaps.** If you have a region in an alignment which has too many gaps, you can select the region and realign it with 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 the two residues, select the region and realign it using the fixpoints. Now, the two residues are aligned with each other and everything in the selected region around them is adjusted to accommodate this change.

Realigning a subset of aligned sequences

To realign only a subset of the sequences of an alignment, you have to select the sequences you want to realign by click-and-drag on their entire length, and open the selection in a new view. This means that the sequences you want to select need to be situated on top of each other in a single stack.

A small stack is easily obtainable by dragging the sequences by their names to the top of the alignment. But when a large quantity of sequences needs to be moved to sit together, we recommend using the selection boxes that are available when checking the option "Show selection boxes" from the Alignment settings section in the right-hand side panel (figure 21.11).

Select all the sequences you want to realign independently of the rest of the alignment, and right-click on the name of one of the sequences to choose the option "Sort Sequences by Marked Status". This will bring all checkbox-selected items to the top of the alignment.

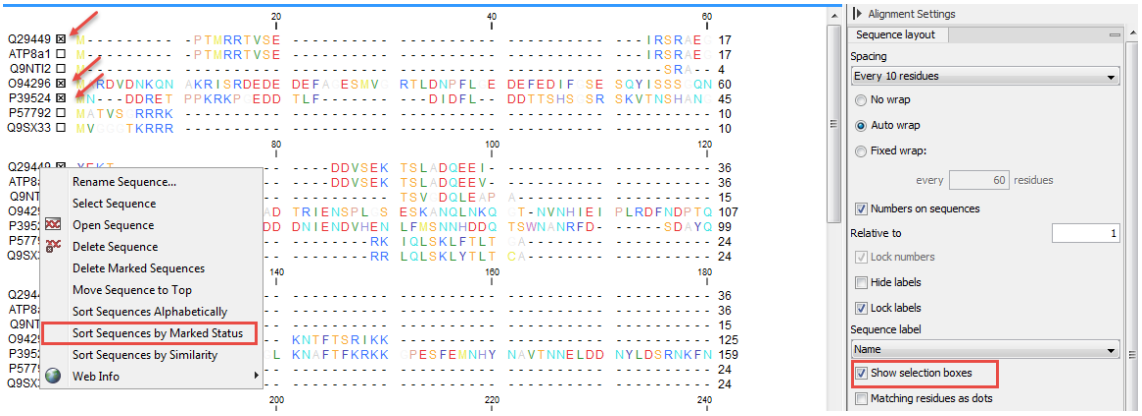


Figure 21.11: *Realigning a subset of the sequences that are part of an alignment. Note that we can see here the menu available when right clicking on the names of the sequences.*

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 21.12), you can choose the option "Open selection in a new view", with the ability to run any relevant tool on that sub-alignment.

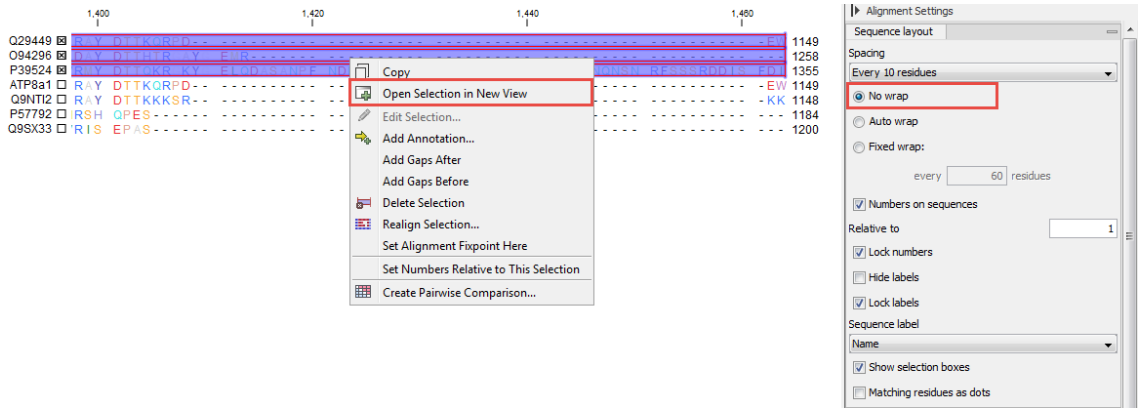


Figure 21.12: *Open the selected sequences in a new window to realign them.*

21.4 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 by:

Toolbox | Classical Sequence Analysis ( ) | Alignments and Trees ( )| Join Alignments ( )

This opens the dialog shown in figure 21.13.

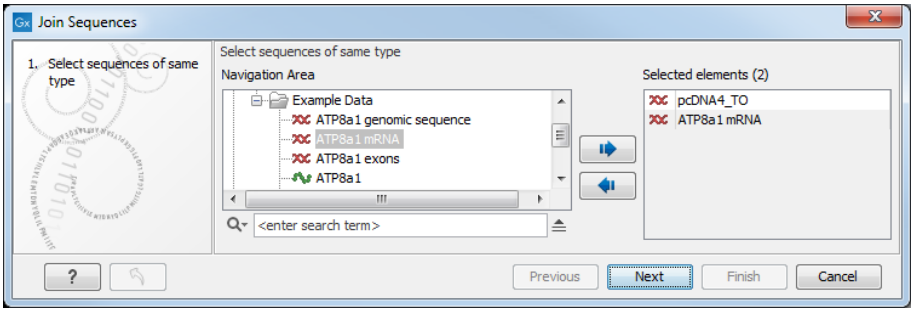


Figure 21.13: *Selecting two alignments to be joined.*

If you have selected some alignments before choosing the Toolbox action, they are now listed 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 21.14.

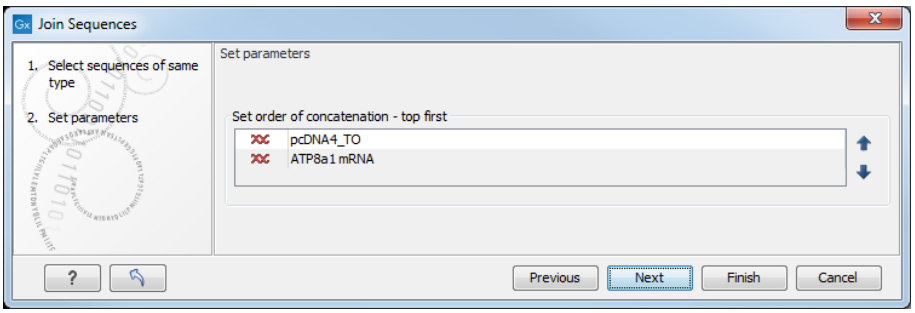


Figure 21.14: *Selecting order of concatenation.*

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 21.15.

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 considered to have the same origin and are thus joined. Consider the joining of the alignments shown in figure 21.15 "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.

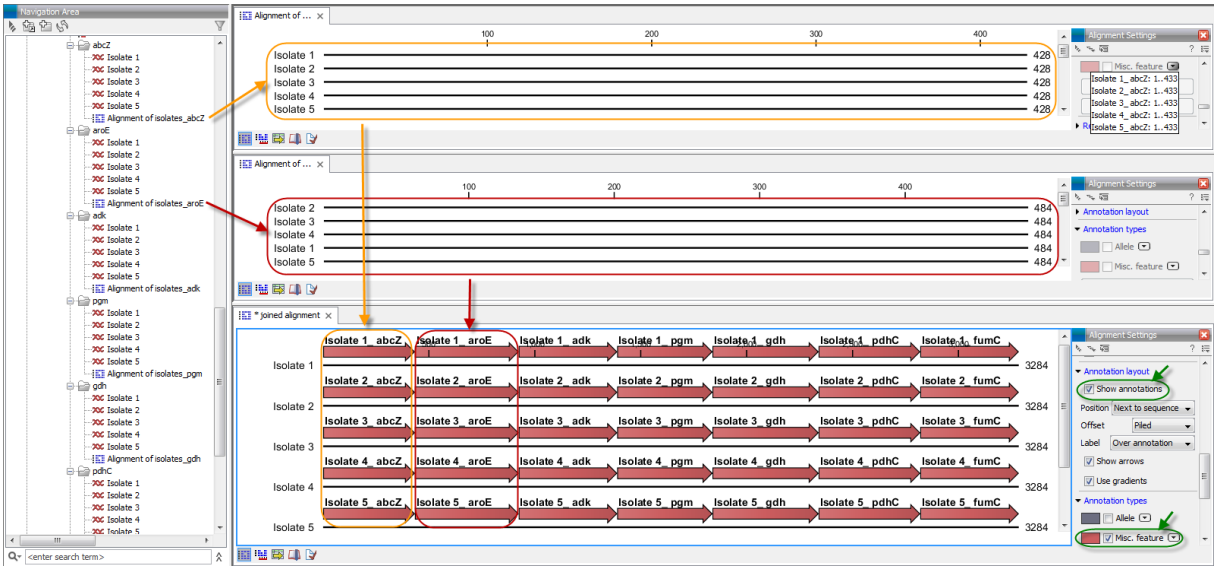


Figure 21.15: *The upper part of the figure shows two of the seven alignments for the genes "abcZ" and "aroE" respectively. Each alignment consists of sequences from one gene from five different isolates. The lower part of the figure shows the result of "Join Alignments". Seven genes have been joined 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.*

21.5 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:

Toolbox | **Classical Sequence Analysis** ( ) | **Alignments and Trees** ( )| **Create Pairwise Comparison** ( )

This opens the dialog displayed in figure 21.16:

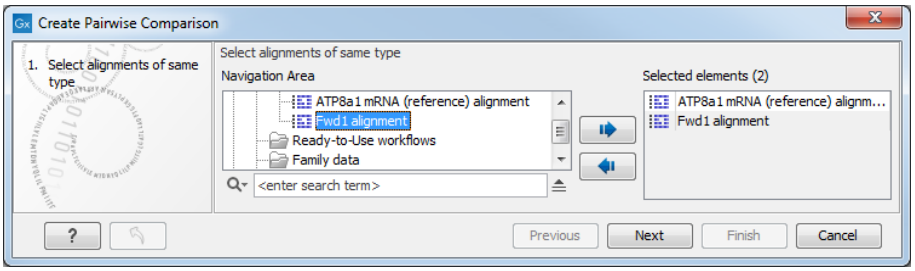


Figure 21.16: *Creating a pairwise comparison table.*

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 21.17.

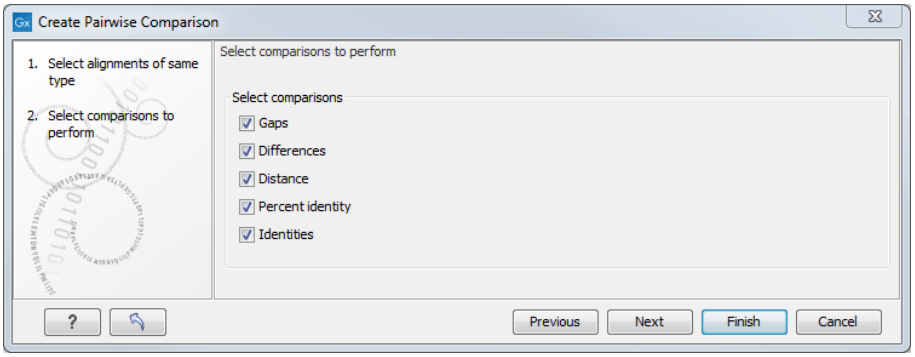


Figure 21.17: *Adjusting parameters for pairwise comparison.*

- **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.

21.5.1 The pairwise comparison table

The table shows the results of selected comparisons (see an example in figure 21.18). 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.

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.

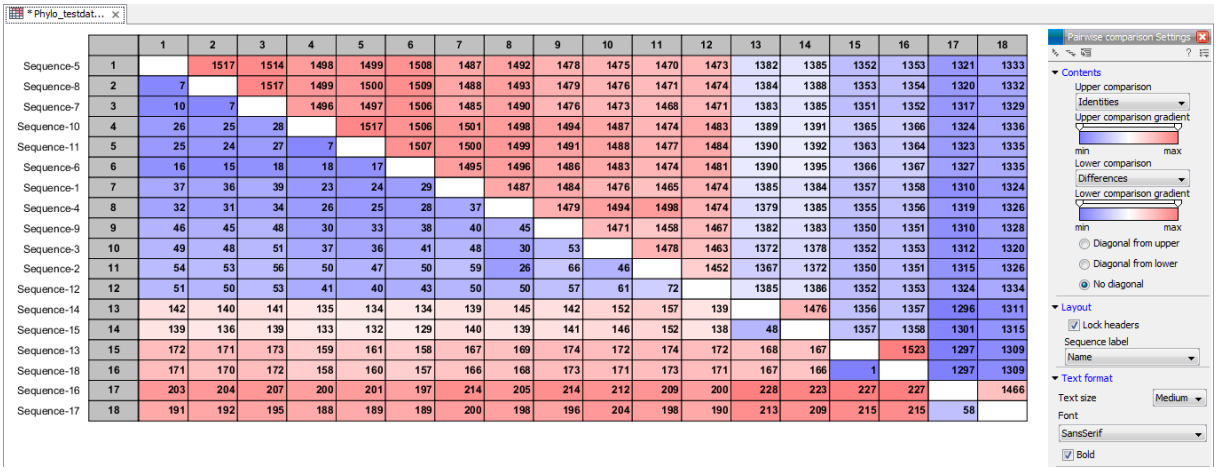


Figure 21.18: *A pairwise comparison table.*

- **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.

21.5.2 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 21.19) 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.

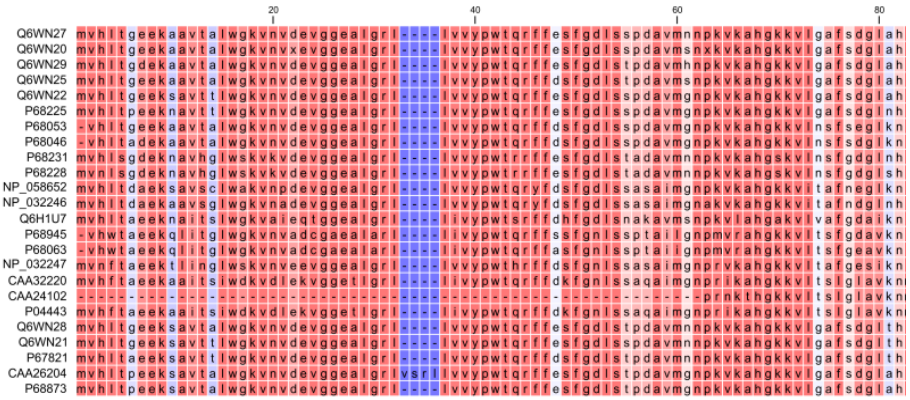


Figure 21.19: *The tabular format of a multiple alignment of 24 Hemoglobin protein sequences. Sequence names appear at the beginning of each row and the residue position is indicated by the numbers at the top of the alignment columns. The level of sequence conservation is shown on a color scale with blue residues being the least conserved and red residues being the most conserved.*

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.

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 22

Phylogenetic trees

Contents

22.1 K-mer Based Tree Construction . . . 516
22.2 Create tree . . . 517
22.3 Model Testing . . . 518
22.4 Maximum Likelihood Phylogeny . . . 520
22.4.1 Bioinformatics explained . . . 523
22.5 Tree Settings . . . 528
22.5.1 Minimap . . . 529
22.5.2 Tree layout . . . 529
22.5.3 Node settings . . . 531
22.5.4 Label settings . . . 531
22.5.5 Background settings . . . 533
22.5.6 Branch layout . . . 534
22.5.7 Bootstrap settings . . . 534
22.5.8 Visualizing metadata . . . 535
22.5.9 Node right click menu . . . 535
22.6 Metadata and phylogenetic trees . . . 537
22.6.1 Table Settings and Filtering . . . 539
22.6.2 Add or modify metadata on a tree . . . 539
22.6.3 Undefined metadata values on a tree . . . 540
22.6.4 Selection of specific nodes . . . 541

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 22.4.1 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 21.1) 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 22.4.1). 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 22.4.1). 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 releated, 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 22.4 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 22.3) in order to identify the best suitable models for creating a tree.

- **Create Tree** ( ) Is a tool that uses distance estimates computed from multiple alignments to create trees. 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 22.4.1).

22.1 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:

Toolbox | Classical Sequence Analysis ( ) | Alignments and Trees ( )| K-mer Based Tree Construction ( )

Select sequences or a sequence list (figure 22.1):

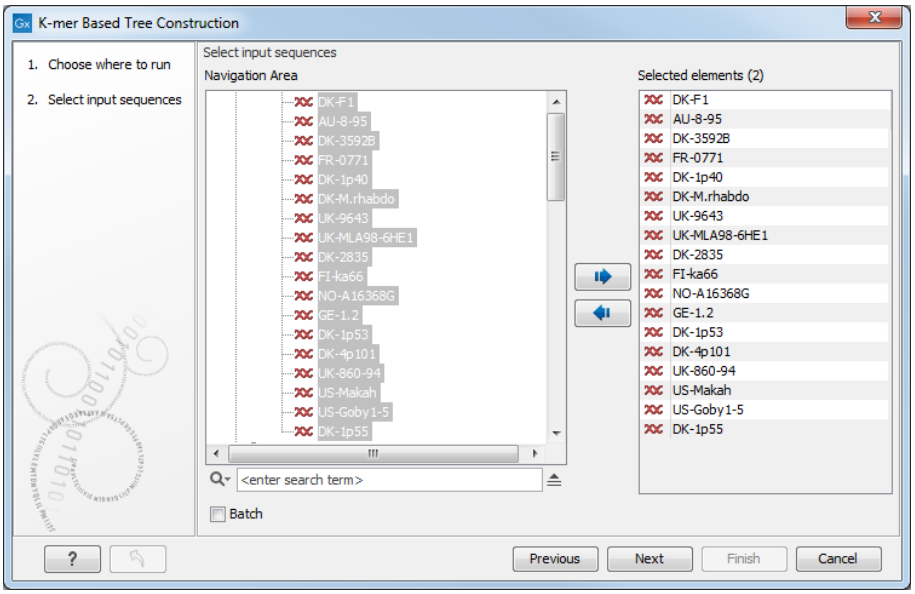


Figure 22.1: *Select sequences needed for creating a tree with K-mer based tree construction.*

Next, select the construction method, specify the k-mer length and select a distance measure for tree construction (figure 22.2):

- **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 22.4.1):
    - The **UPGMA** method. Assumes constant rate of evolution.
    - The **Neighbor Joining** method. Well suited for trees with varying rates of evolution.
- **K-mer settings**

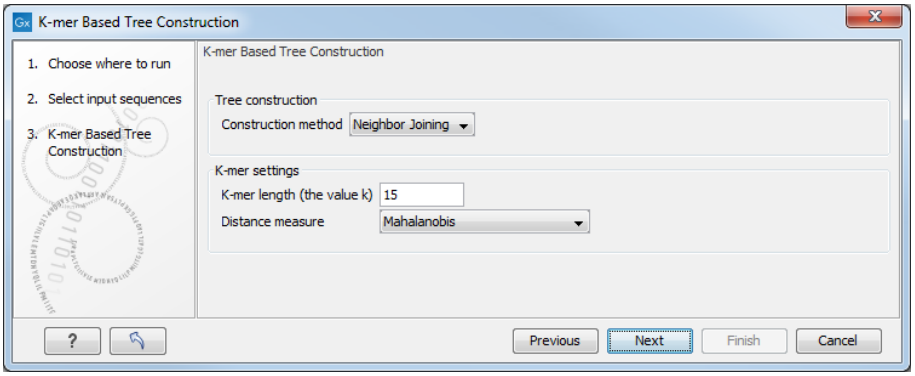


Figure 22.2: *Select the construction method, and specify the k-mer length and a distance measure.*

- **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 22.4.1 for further details.

22.2 Create tree

The Create tree tool can be used to generate a distance-based phylogenetic tree with multiple alignments as input:

Toolbox | Classical Sequence Analysis ( ) | Alignments and Trees ( )| Create Tree ( )

This will open the dialog displayed in figure 22.3:

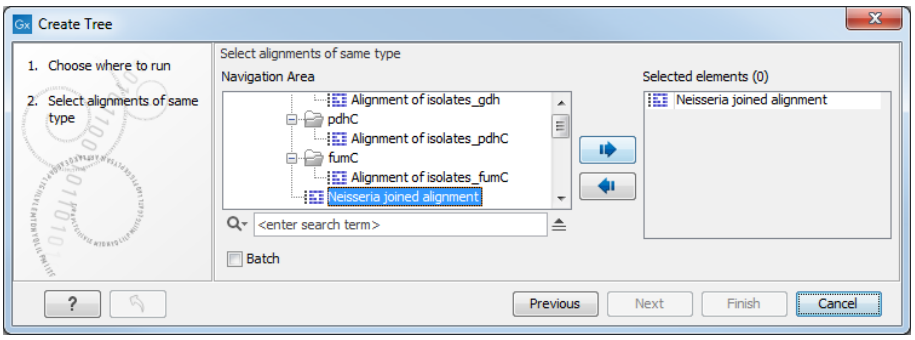


Figure 22.3: *Creating a tree.*

If an alignment was selected before choosing the Toolbox action, this alignment is now 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 22.4 shows the parameters that can be set for this distance-based tree creation:

- Tree construction (see section 22.4.1)
  - Tree construction method
    - The **UPGMA** method. Assumes constant rate of evolution.

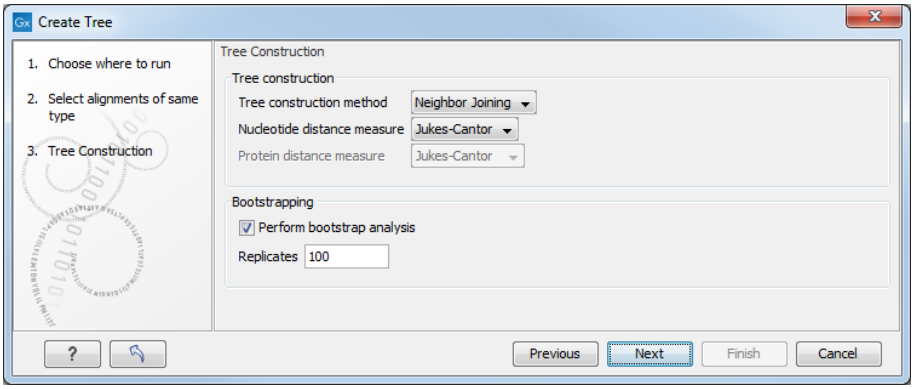


Figure 22.4: *Adjusting parameters for distance-based methods.*

    - 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 22.4.1). 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 22.4.1.

22.3 Model Testing

As the Model Testing tool can help identify the best substitution model (22.4.1) 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:

> **Toolbox | Classical Sequence Analysis ( ) | Alignments and Trees ( )| Model Testing (:)**

Select the alignment that you wish to use for the tree construction (figure 22.5):

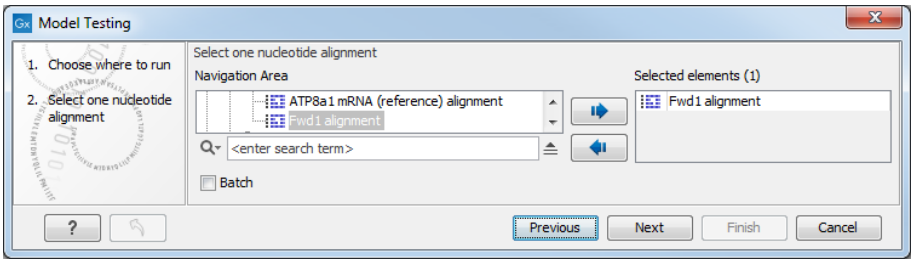


Figure 22.5: *Select alignment for model testing.*

Specify the parameters to be used for model testing (figure 22.6):

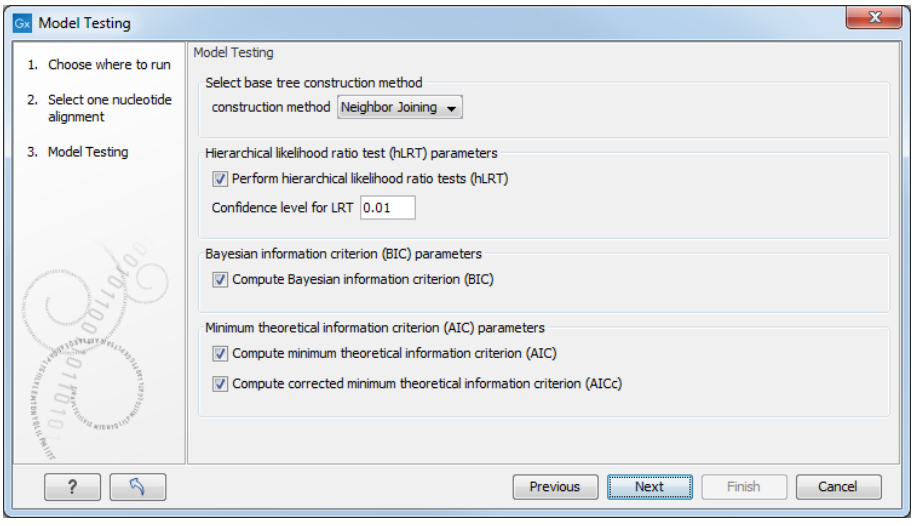


Figure 22.6: *Specify parameters for model testing.*

- **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 = -2ln(L)+Kln(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 = -2ln(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 (AIC)** Rank substitution models based on minimum corrected theoretical information criterion (AICc). Formula used is AICc = -2ln(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.

22.4 Maximum Likelihood Phylogeny

To generate a maximum likelihood based phylogenetic tree, go to:

Toolbox | Classical Sequence Analysis ( ) | Alignments and Trees ( )| Maximum Likelihood Phylogeny ( )

First, select the alignment to be used for the reconstruction (figure 22.7).

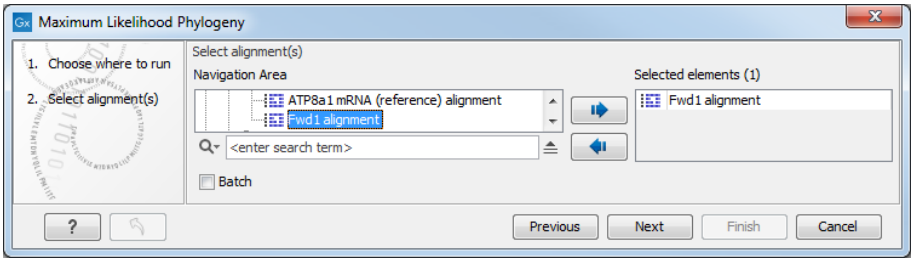


Figure 22.7: *Select the alignment for tree construction.*

You can then set up the following parameters (figure 22.8):

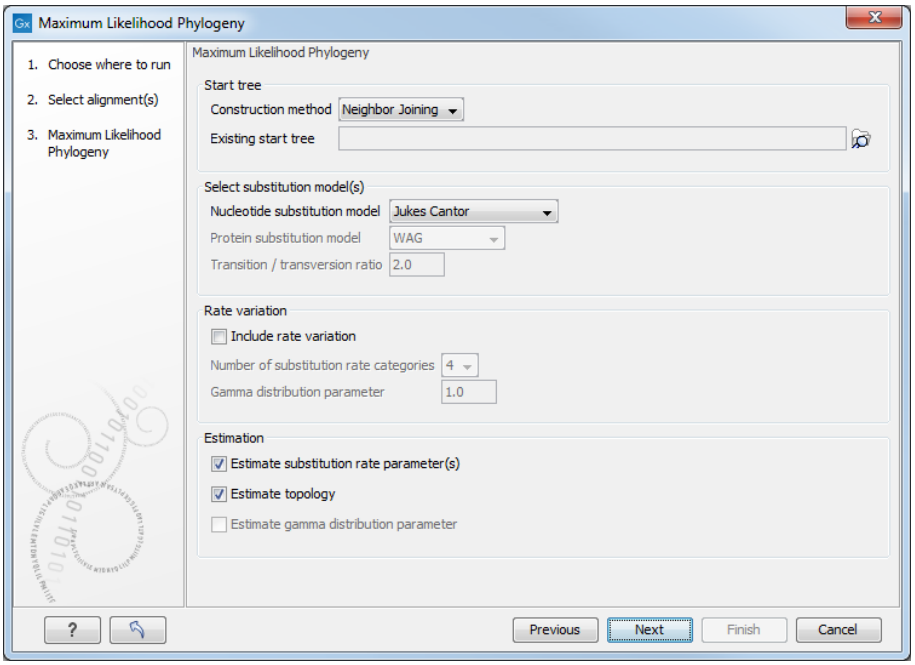


Figure 22.8: *Adjusting parameters for maximum likelihood phylogeny*

- **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**
  - **Nucleotice 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 transtion/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 22.4.1.

– **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 22.9).

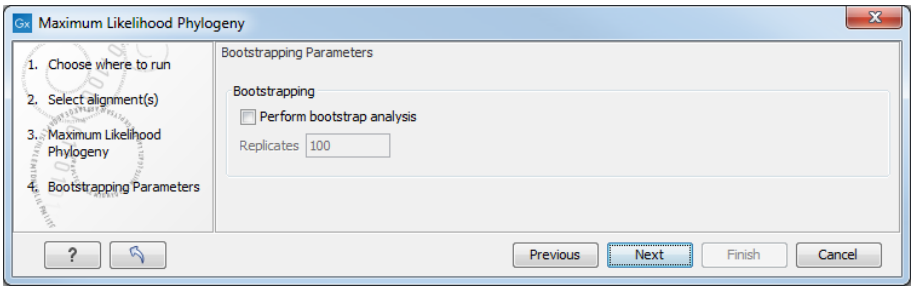


Figure 22.9: *Adjusting parameters for ML phylogeny*

To evaluate the reliability of the inferred trees, *CLC Genomics Workbench* allows the option of doing a **bootstrap** analysis (see section 22.4.1). 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.

22.4.1 Bioinformatics explained

The phylogenetic tree

The evolutionary hypothesis of a phylogeny can be graphically represented by a phylogenetic tree.

Figure 22.10 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 22.10 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.

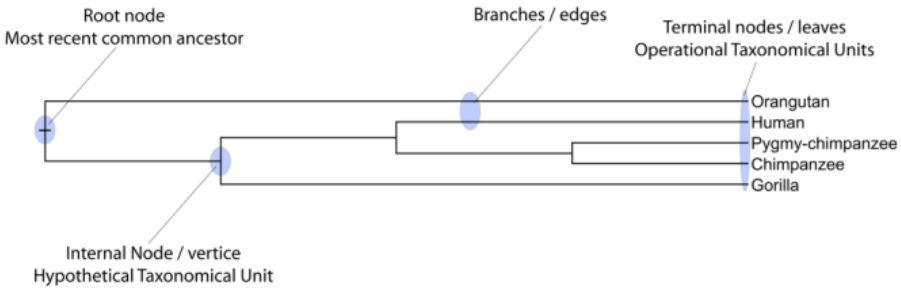


Figure 22.10: *A proposed phylogeny of the great apes (Hominidae). Different components of the tree are marked, see text for description.*

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. $C \to A$, $C \to C$, $C \to G$,...).

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 where 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) occuring 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:

$$\text{dist}(s_1, s_2) = \sum_i (p(s_1)_i/\sigma_i - p(s_2)_i/\sigma_i)^2.$$

  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 22.4.1).

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 reconstruct 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 lenghts. 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 $P(tree|data)$ 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 $P(tree|data)$ so instead ML based methods have to compute the probability of the data given a tree, i.e. $P(data|tree)$. 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 converges 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 22.3) 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.

22.5 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 22.6).

The preferred tree layout settings (user defined tree settings) can be saved and applied via the top right **Save Tree Settings** (figure 22.11). 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.

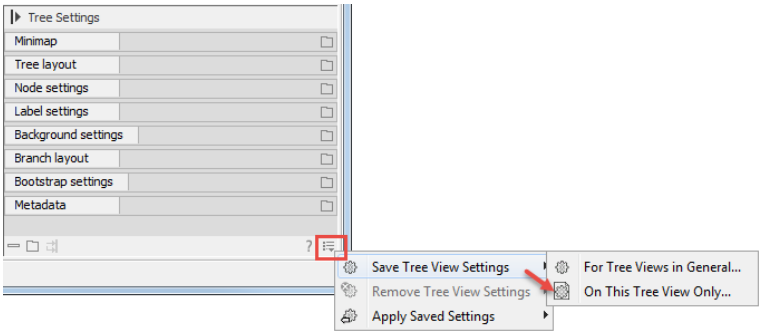


Figure 22.11: *Save, remove or apply preferred layout settings.*

22.5.1 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 22.12). To navigate the tree using the Minimap, click on the Minimap with the mouse and move the grey square around within the Minimap.

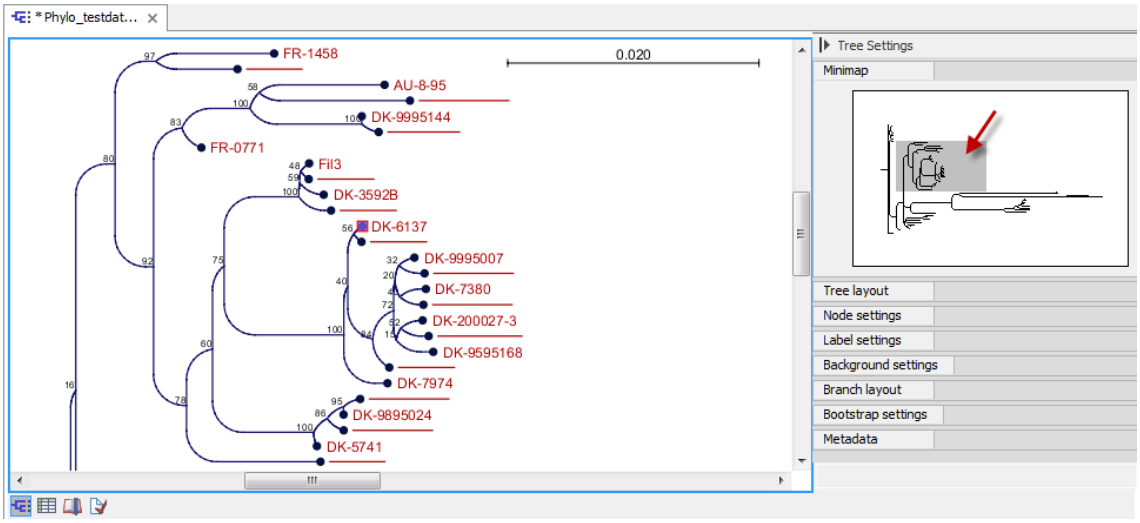


Figure 22.12: *Visualization of a phylogenetic tree. The grey square in the Minimap shows the part of the tree that is shown in the View Area.*

22.5.2 Tree layout

The **Tree Layout** can be adjusted in the Side Panel (figure 22.13).

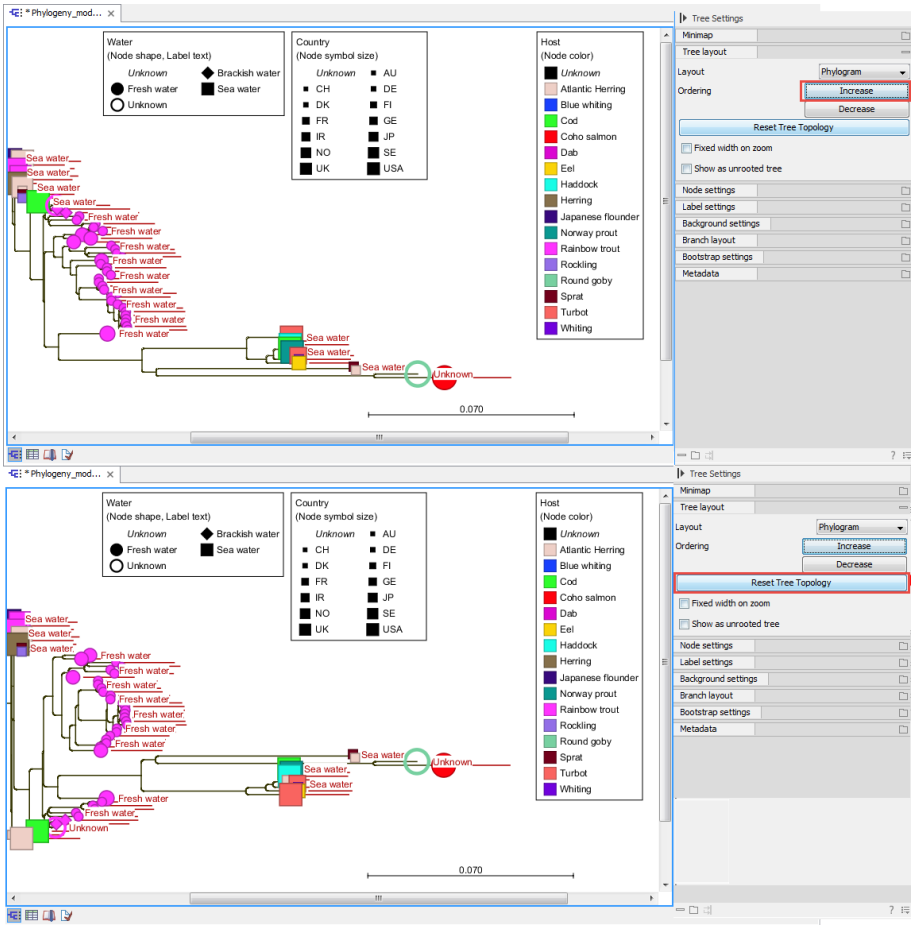


Figure 22.13: *The tree layout can be adjusted in the Side Panel. The top part of the figure shows a tree with increasing node order. In the bottom part of the figure the tree has been reverted to the original tree topology.*

- **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 22.13) or **Decreasing**.
- **Reset Tree Topology** Resets to the default tree topology and node order (see figure 22.13). 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.

22.5.3 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 22.14.

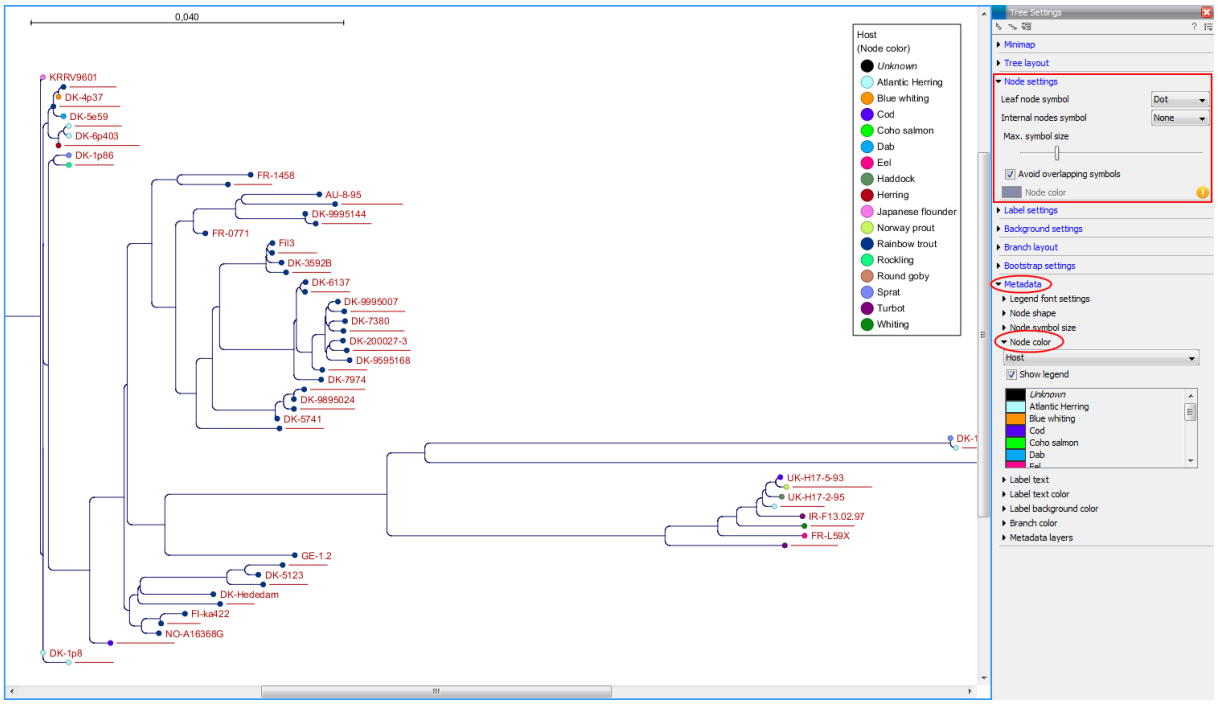


Figure 22.14: *The Node Layout settings. Node color is specified by metadata and is therefore inactive in this example.*

22.5.4 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 22.15).
- **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 22.5.9).
- **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.

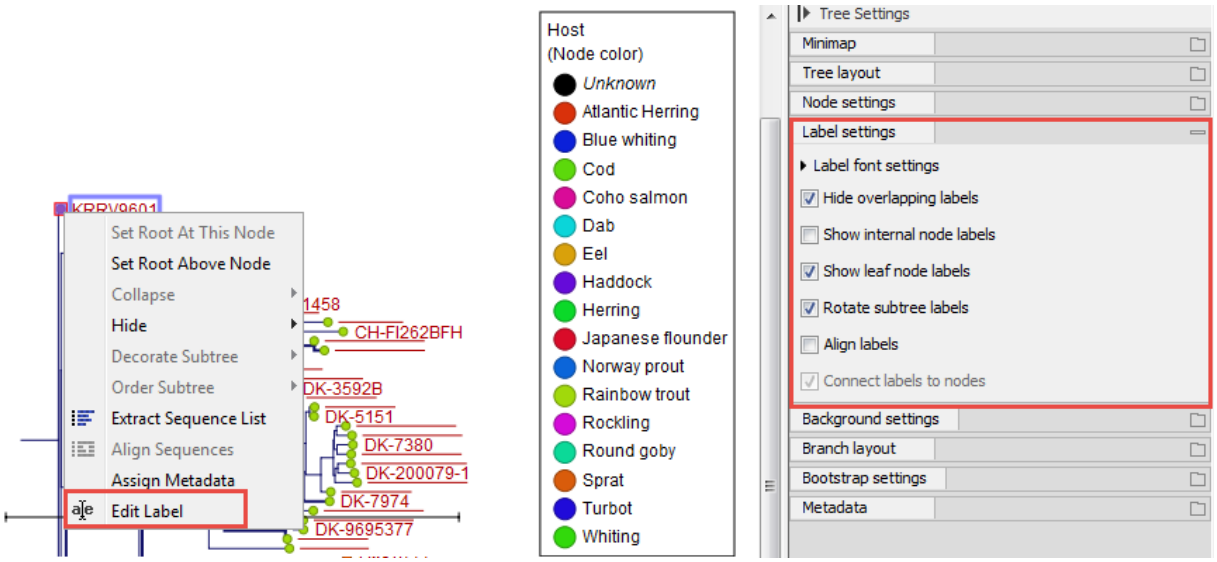


Figure 22.15: *"Edit label" in the right click menu can be used to customize the label text. The way node labels are displayed can be controlled through the labels settings in the right side panel.*

When working with big trees there is typically not enough space to show all labels. As illustrated in figure 22.15, only some of the labels are shown. The hidden labels are illustrated with thin horizontal lines (figure 22.16).

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 22.16 and figure 22.17). 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 22.5.5).

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.

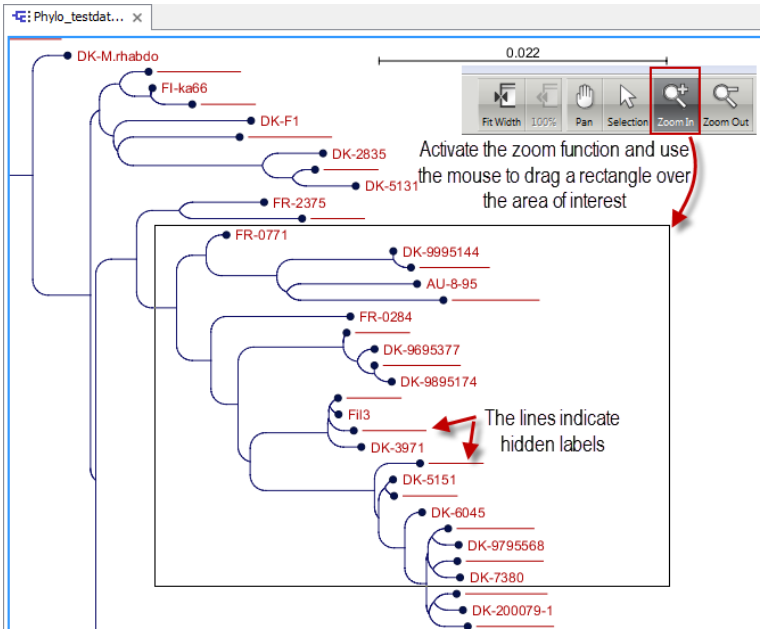


Figure 22.16: *The zoom function in the upper right corner of the Workbench can be used to zoom in on a particular region of the tree. When the zoom function has been activated, use the mouse to drag a rectangle over the area that you wish to zoom in at.*

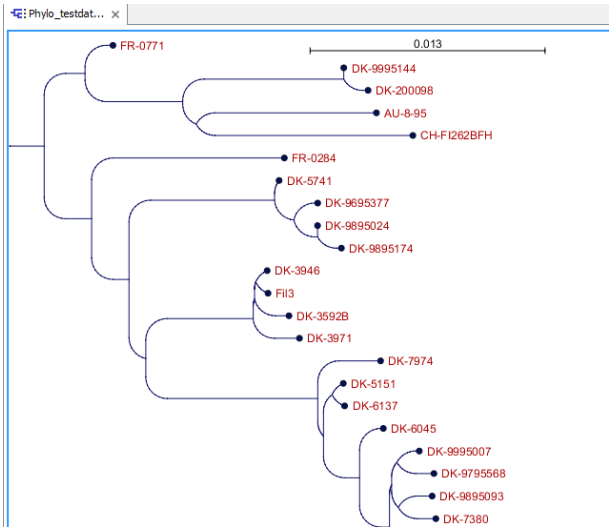


Figure 22.17: *After zooming in on a region of interest more labels become visible. In this example all labels are now visible.*

22.5.5 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.

22.5.6 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 22.18.

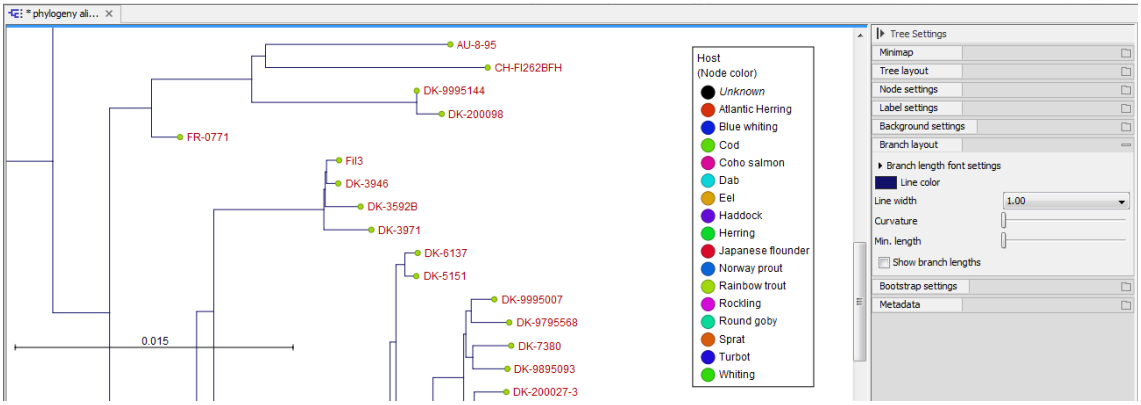


Figure 22.18: *Branch Layout settings.*

22.5.7 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.

22.5.8 Visualizing metadata

Metadata associated with a phylogenetic tree (described in detail in section 22.6) 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 22.19.
- **Label text color** The label text can be colored and used to visualize metadata (see figure 22.19).
- **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 22.14).

22.5.9 Node right click menu

Additional options for layout and extraction of subtree data are available when right clicking the nodes (figure 22.15):

- **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.
- **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.

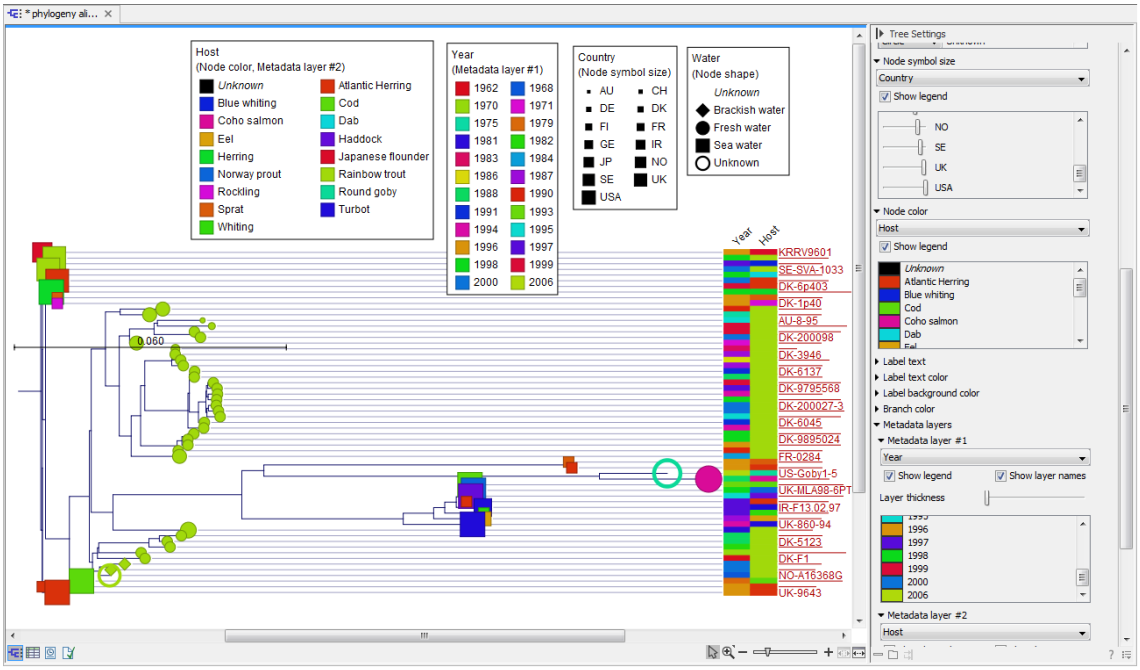


Figure 22.19: *Different types of metadata kan be visualized by adjusting node size, shape, and color. Two color-code metadata layers (Year and Host) are shown in the right side of the tree.*

- **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 22.20). When hiding nodes, a new button appears labeled "Show X hidden nodes" in the Side Panel under "Tree Layout" (figure 22.21). 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 22.11 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.

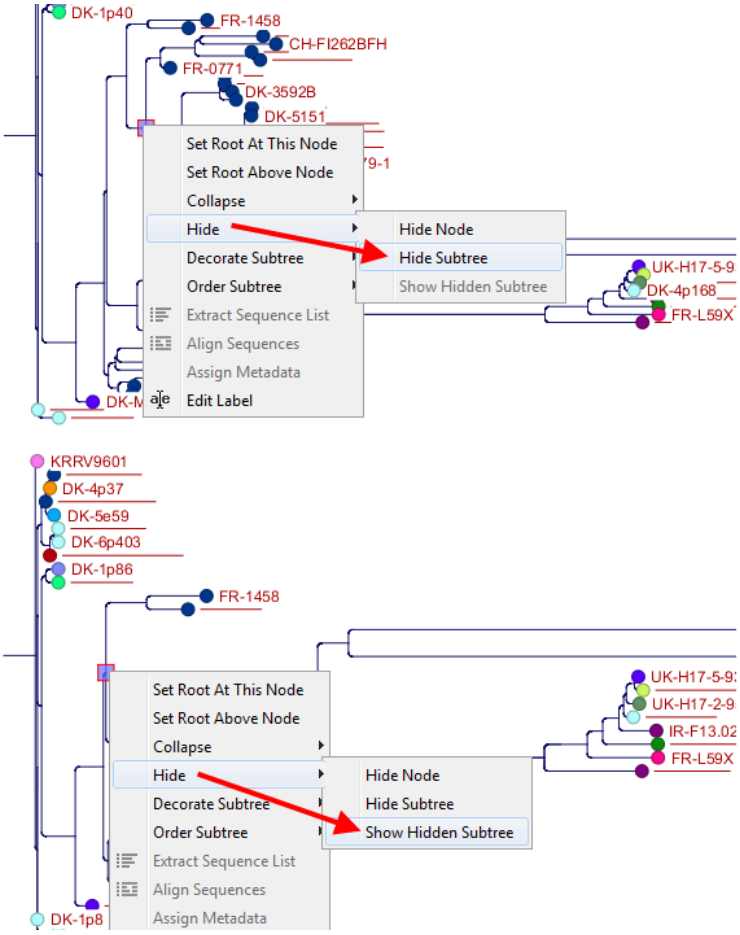


Figure 22.20: *A subtree can be hidden by selecting "Hide Subtree" and is shown again when selecting "Show Hidden Subtree" on a parent node.*

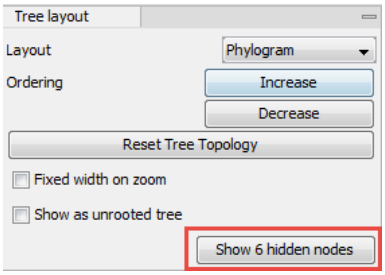


Figure 22.21: *When hiding nodes, a new button labeled "Show X hidden nodes" appears in the Side Panel under "Tree Layout". When pressing this button, all hidden nodes are brought back.*

- **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**

22.6 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 22.22).

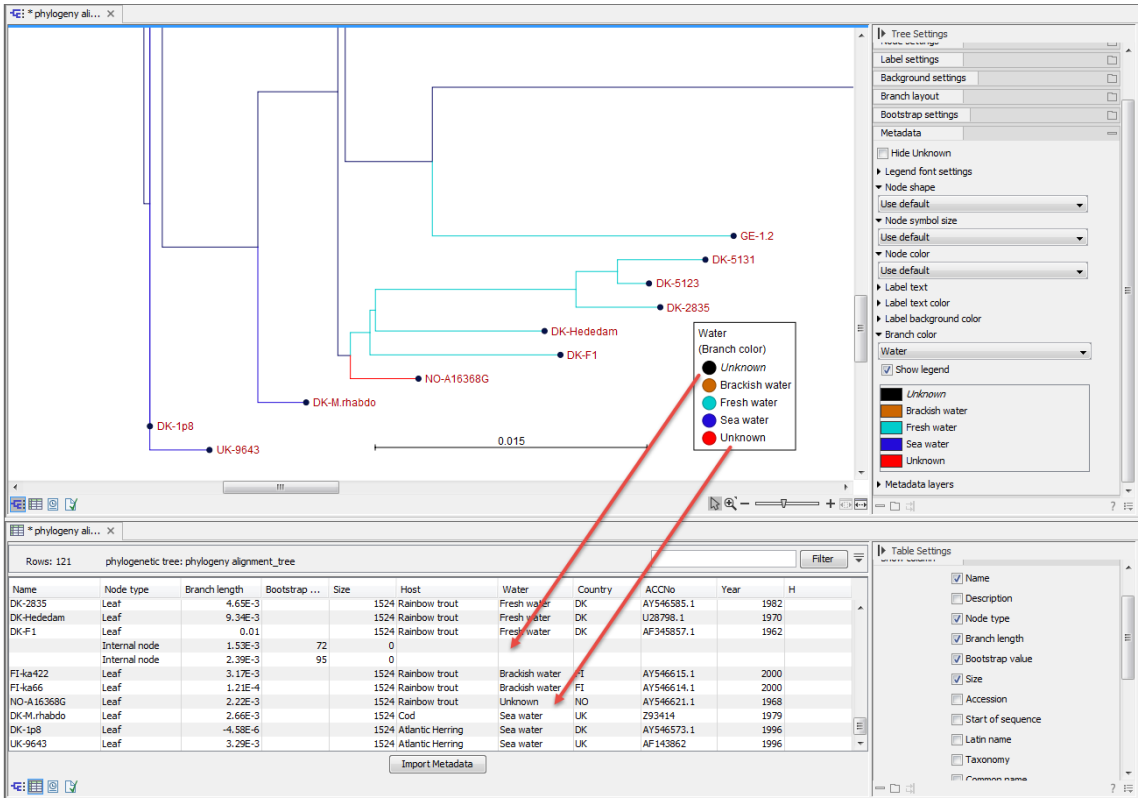


Figure 22.22: *Tabular metadata that is associated with an existing tree shown in a split view. Note that Unknown written in italics (black branches) refer to missing metadata, while Unknown in regular font refers to metadata labeled as "Unknown".*

Additional metadata can be associated with a tree by clicking the **Import Metadata** button. This will open up the dialog shown in figure 22.23.

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 22.23. 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.

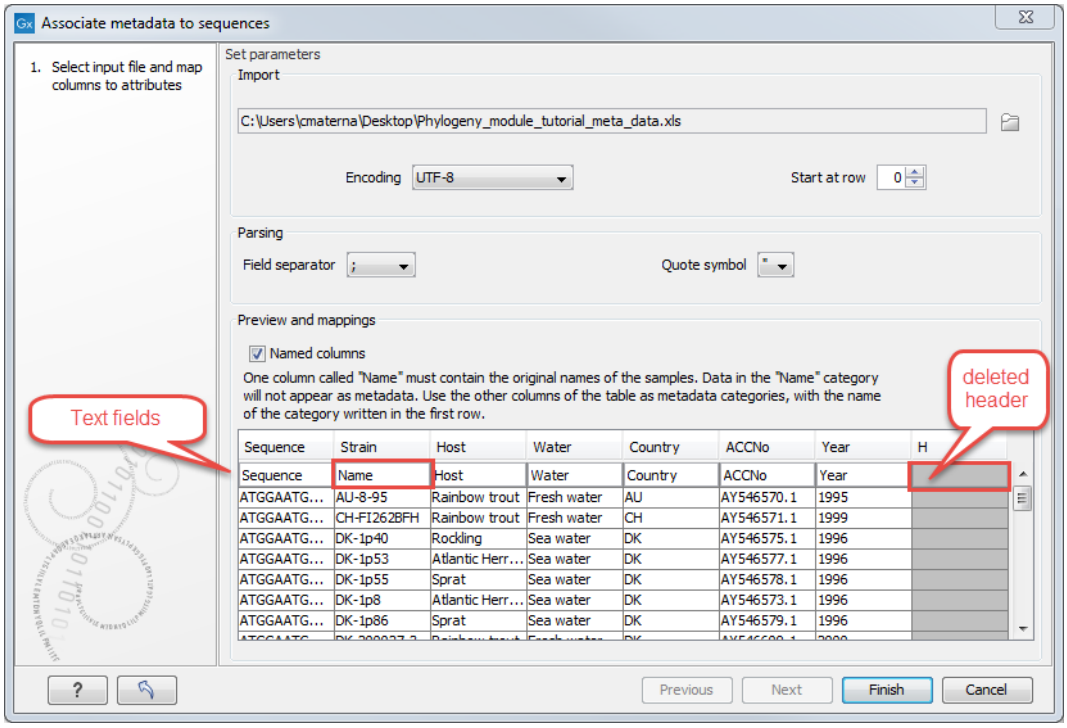


Figure 22.23: *Import of metadata for a tree. The second column named "Strain" is choosen as the common denominator by entering "Name" in the text field of the column. The column labeled "H" is ignored by not assigning a column heading to this column.*

22.6.1 Table Settings and Filtering

How to use the metadata table (see figure 22.24):

- **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 3.2.1).

22.6.2 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 22.25):

- **Assign Metadata** The right click option "Assign Metadata" can be used for four purposes.

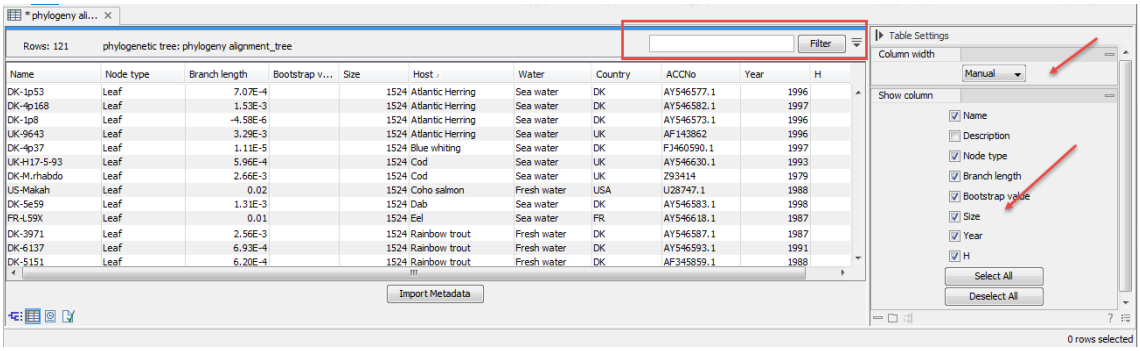


Figure 22.24: *Metadata table. The column width can be adjusted manually or automatically. Under "Show column" it is possible to select which columns should be shown in the table. Filtering using specific criteria can be performed.*

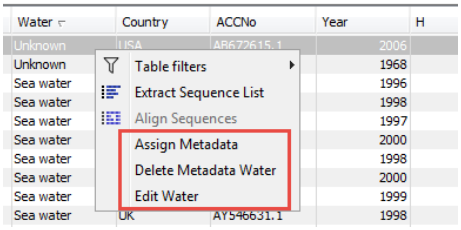


Figure 22.25: *Right click options in the metadata table.*

- 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 22.26). This will change values in all selected rows in the column that was clicked.

22.6.3 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

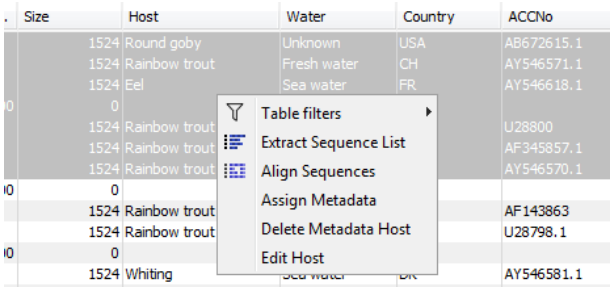


Figure 22.26: *To modify existing metadata, click on the specific field, select "Edit <column header>" and provide a new value.*

top of the legend (see the entry "*Unknown*" in figure 22.27). To remove this entry in the legend, all nodes must have a value assigned in the corresponding metadata category.

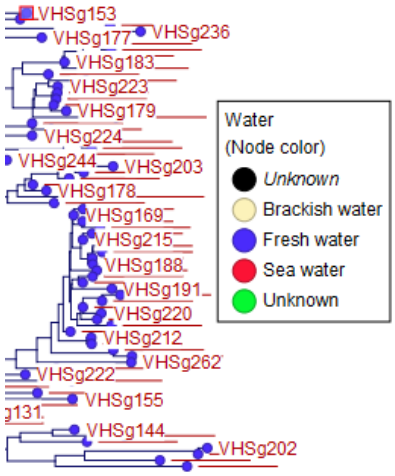


Figure 22.27: *A legend for a metadata category where one or more values are undefined. Fill your metadata table with a value of your choice to edit the mention of "*("Unknown" *in the legend. Note that the "Unknown" that is not in italics is used for data that had a value written as "Unknown" in the metadata table.*

22.6.4 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 22.28).
- *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 tool (see section 21.5.2). Next, change relevant option in the multiple alignment wizard that pops up and click **Finish**. The multiple alignment will now be generated.

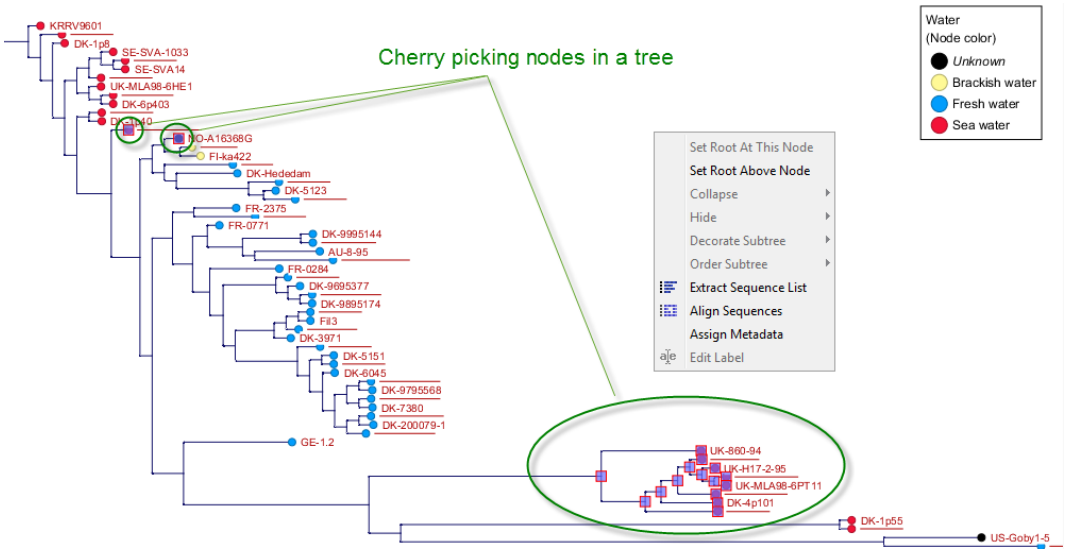


Figure 22.28: *Cherry picking nodes in a tree. The selected leaf sequences can be extracted by right clicking on one of the selected nodes and selecting "Extract Sequence List". It is also possible to Align Sequences directly by right clicking on the nodes or leaves.*

Chapter 23

RNA structure

Contents

23.1 RNA secondary structure prediction 544
23.1.1 Selecting sequences for prediction 544
23.1.2 Secondary structure prediction parameters 545
23.1.3 Structure as annotation 549
23.2 View and edit secondary structures 550
23.2.1 Graphical view and editing of secondary structure 550
23.2.2 Tabular view of structures and energy contributions 553
23.2.3 Symbolic representation in sequence view 556
23.2.4 Probability-based coloring 557
23.3 Evaluate structure hypothesis 558
23.3.1 Selecting sequences for evaluation 558
23.3.2 Probabilities 559
23.4 Structure scanning plot 560
23.4.1 Selecting sequences for scanning 560
23.4.2 The structure scanning result 561
23.5 Bioinformatics explained: RNA structure prediction by minimum free energy minimization 562
23.5.1 The algorithm 563
23.5.2 Structure elements and their energy contribution 564

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 know 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

23.1 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 23.5.

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, see `http://mfold.rna.albany.edu/?q=mfold/mfold-references`.

23.1.1 Selecting sequences for prediction

Secondary structure prediction can be accessed in the **Toolbox**:

Toolbox | **Classical Sequence Analysis ( )** | **RNA Structure ( )**| **Predict Secondary Structure ( )**

This opens the dialog shown in figure 23.1.

If you have selected sequences before choosing the Toolbox action, they are 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. You can use both DNA and RNA sequences - DNA will be folded as if it were RNA.

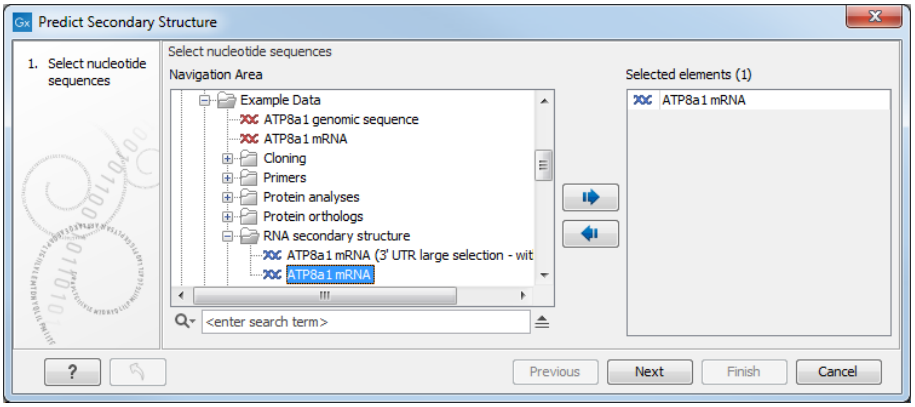


Figure 23.1: *Selecting RNA or DNA sequences for structure prediction (DNA is folded as if it was RNA).*

23.1.2 Secondary structure prediction parameters

Click **Next** to adjust secondary structure prediction parameters (figure 23.2).

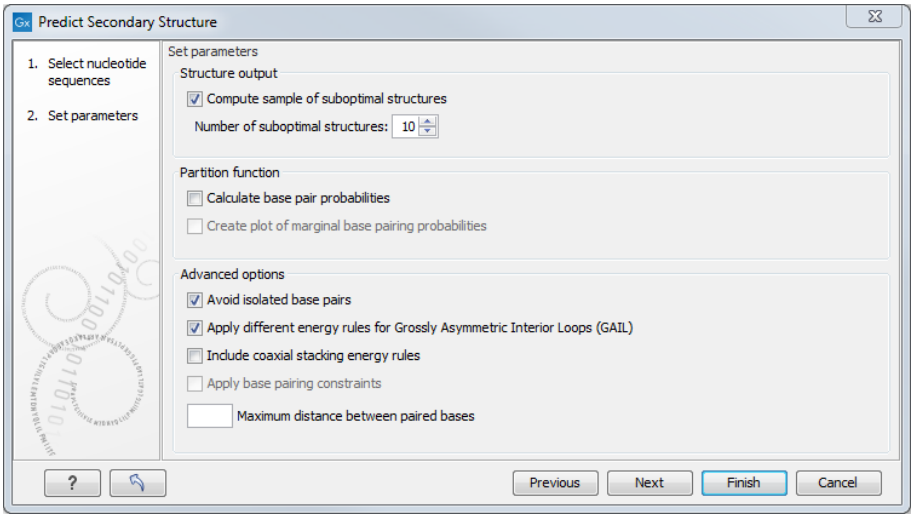


Figure 23.2: *Adjusting parameters for secondary structure prediction.*

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 23.3.

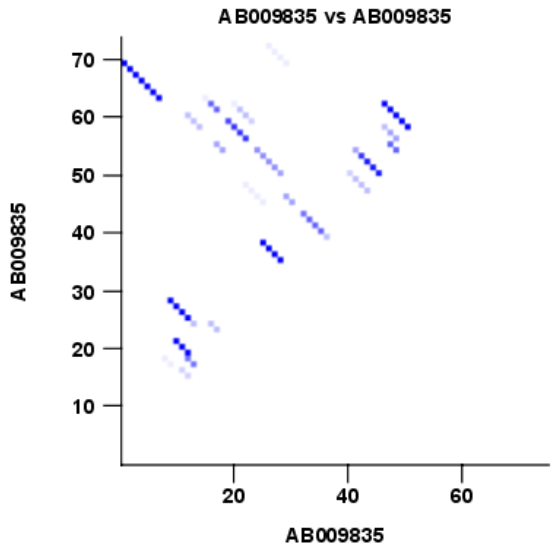


Figure 23.3: *The marginal base pair probability of all possible base pairs.*

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 23.2.3 and also in the secondary structure view. An example is shown in figure 23.4. Furthermore, the marginal probabilities are accessible from tooltips when hovering over the relevant parts of the structure.

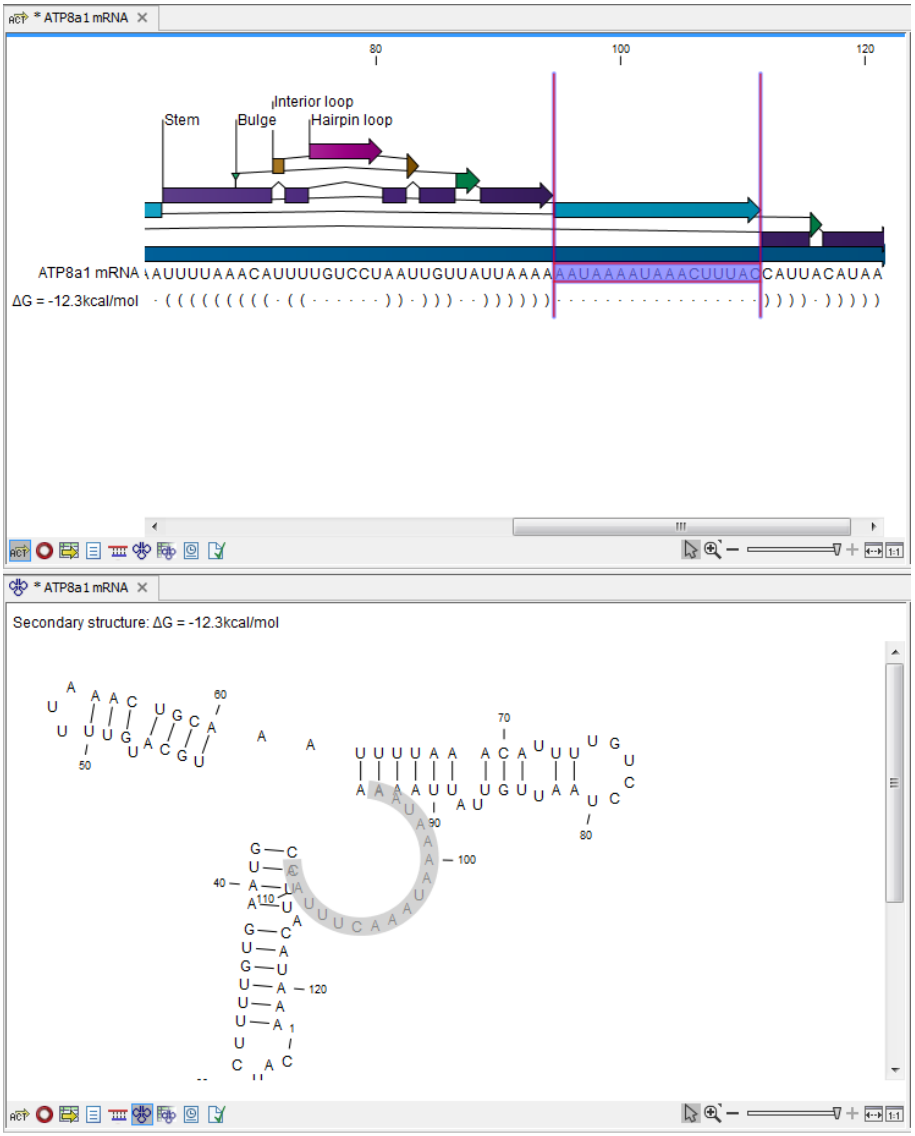


Figure 23.4: *Marginal probability of base pairs shown in linear view (top) and marginal probability of being unpaired shown in the secondary structure 2D view (bottom).*

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://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].
- **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 23.1.2).

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 23.5).

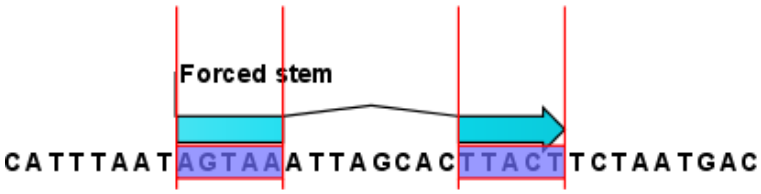


Figure 23.5: *Force a stem of the selected bases.*

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 23.6).

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:

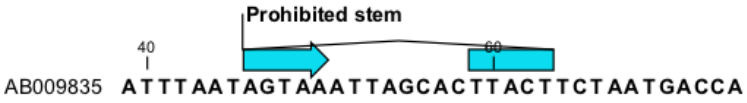


Figure 23.6: *Prohibit the selected bases from forming a stem.*

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 23.7.



Figure 23.7: *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.

23.1.3 Structure as annotation

You can choose to add the elements of the best structure as annotations (see figure 23.8).

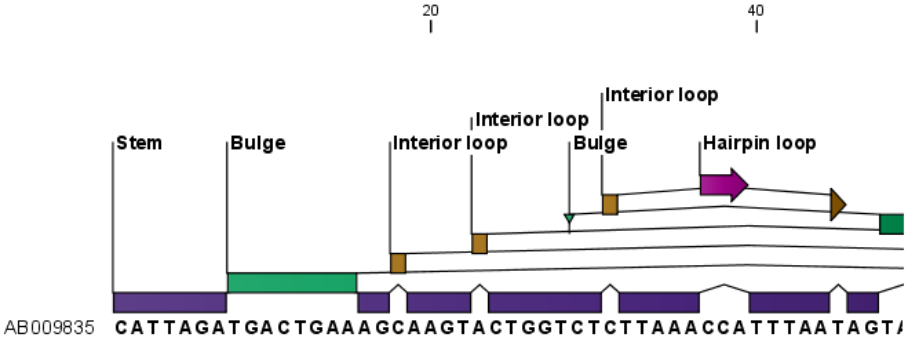


Figure 23.8: *Annotations added for each structure element.*

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 23.2.2.

23.2 View and edit secondary structures

When you predict RNA secondary structure (see section 23.1), 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 23.2. See an example in figure 23.8.
- Symbolic representation below the sequence (see section 23.2.3).
- A graphical view of the secondary structure (see section 23.2.1).
- 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 23.2.2.

23.2.1 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 23.9.

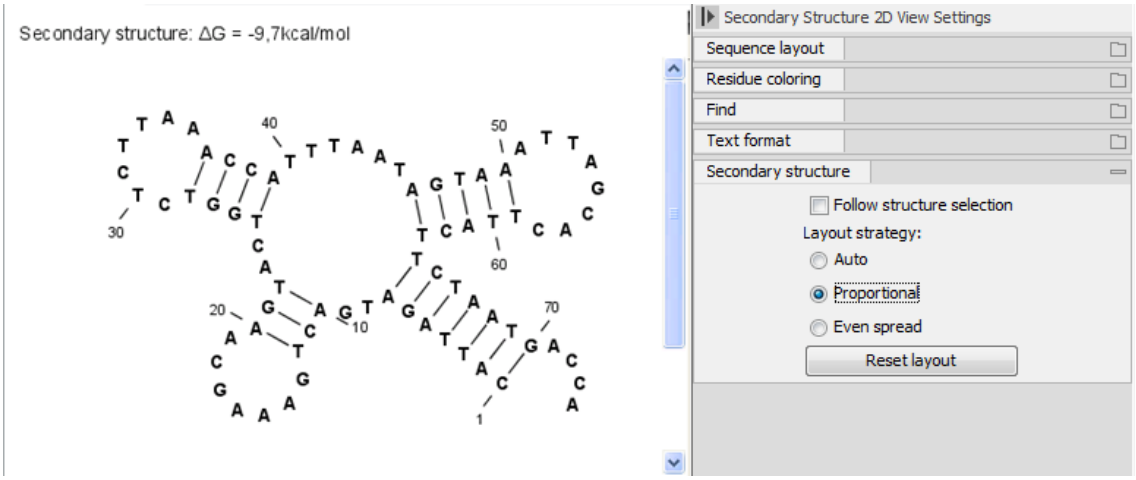


Figure 23.9: *The secondary structure view of an RNA sequence zoomed in.*

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 23.9. 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 12.1. 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 ( ). 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 23.2.2 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 23.10).
  - **Proportional.** Arc lengths are proportional to the number of residues (see figure 23.11). Nothing is done to prevent overlap.
  - **Even spread.** Stems are spread evenly around loops as shown in figure 23.12.
- **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.

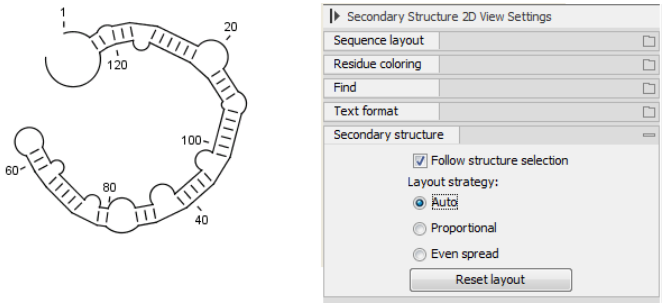


Figure 23.10: *Auto layout. Overlaps are minimized.*

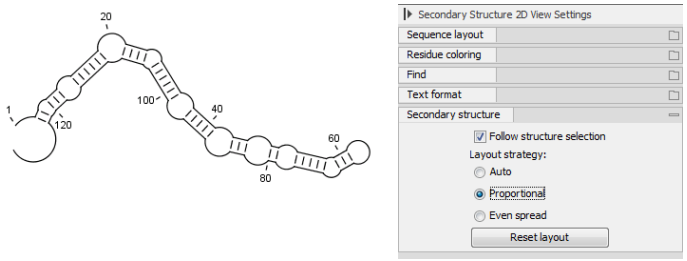


Figure 23.11: *Proportional layout. Length of the arc is proportional to the number of residues in the arc.*

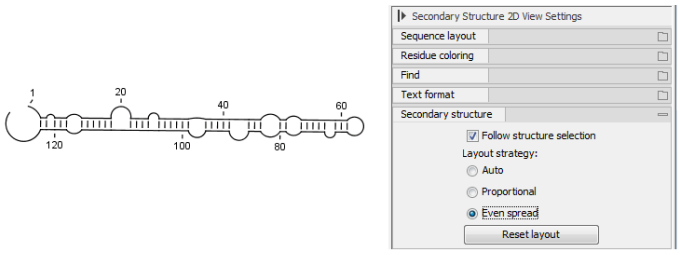


Figure 23.12: *Even spread. Stems are spread evenly around loops.*

Selecting and editing

When you are in **Selection mode** ( ), 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 23.13).

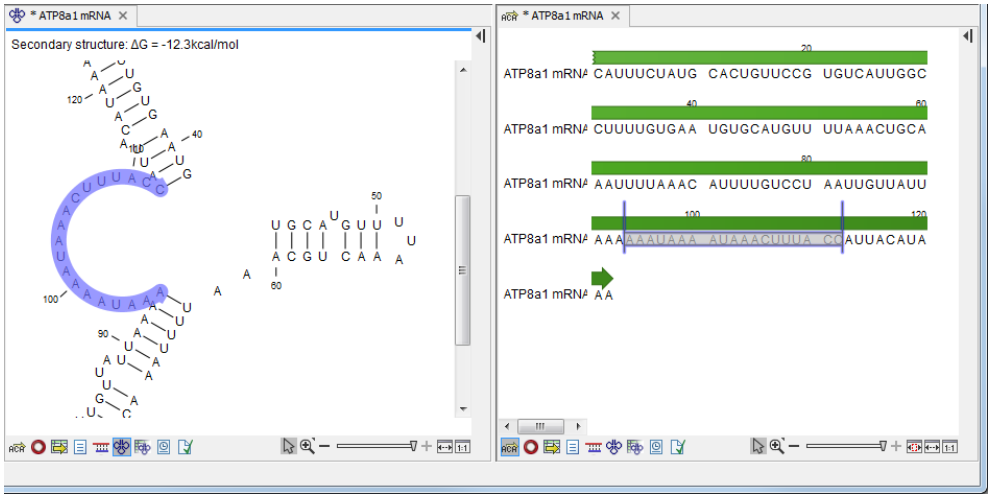


Figure 23.13: *A split view of the secondary structure view and a linear sequence view.*

If you make a selection in another sequence view, this will 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 23.14).

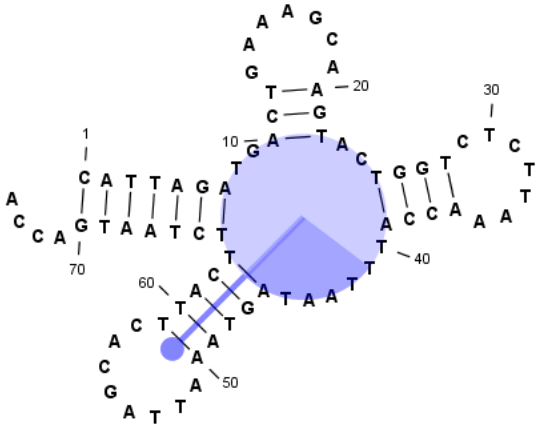


Figure 23.14: *The blue circle represents the anchor point for rotating the substructure.*

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 23.15, the structure shown in figure 23.14 has been modified by dragging with the mouse.

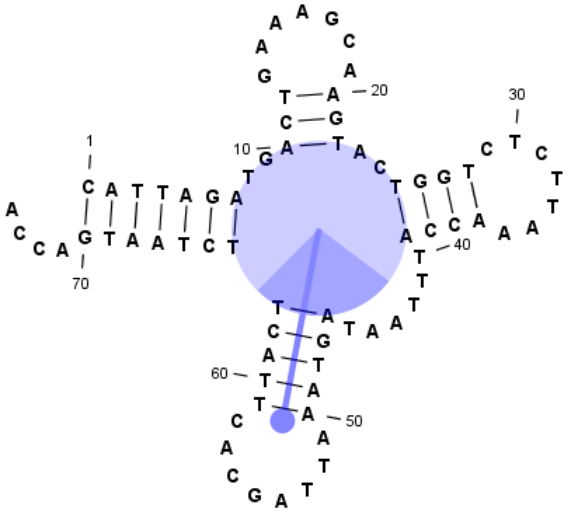


Figure 23.15: *The structure has now been rotated.*

Press **Reset layout** in the **Side Panel** to reset the layout to the way it looked when the structure was predicted.

23.2.2 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 23.1), 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 23.2.1).
- 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 23.16.

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 23.5.2). Each substructure

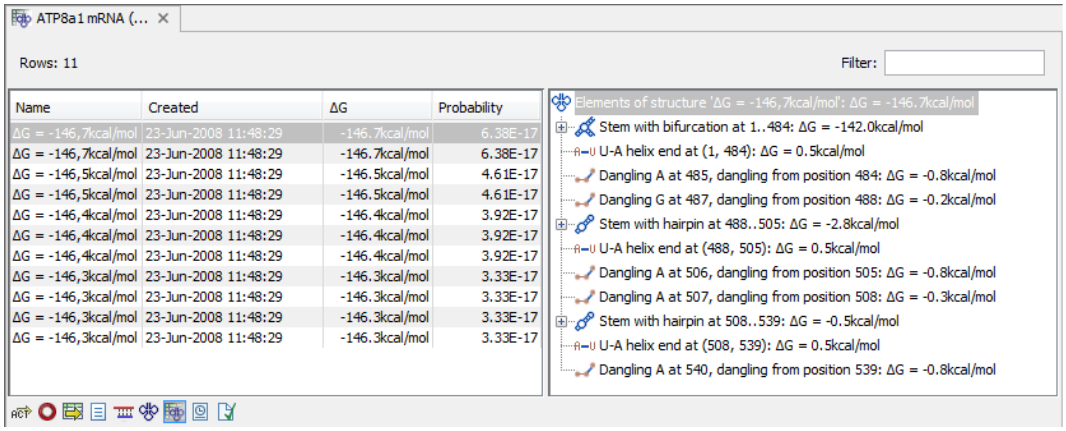


Figure 23.16: *The secondary structure table with the list of structures to the left, and to the right the substructures of the selected structure.*

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 23.17).

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 23.18).

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 23.18 the "Stem with bifurcation" is selected in the table, and this part of the structure is high-lighted in the Secondary Structure 2D view.

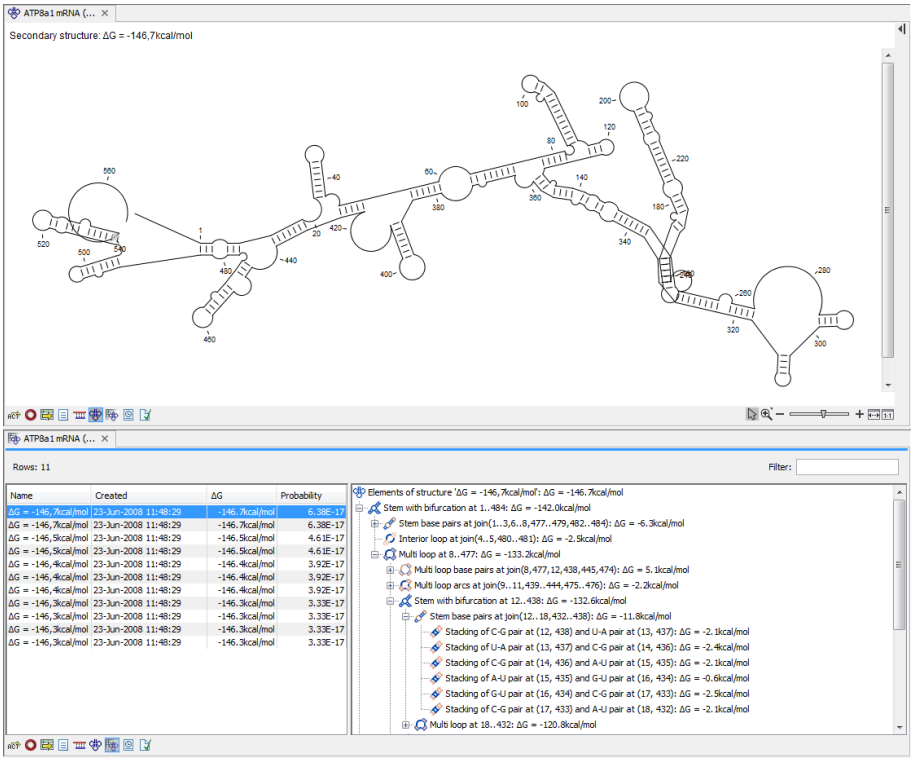


Figure 23.17: *A split view showing a structure table to the right and the secondary structure 2D view to the left.*

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.

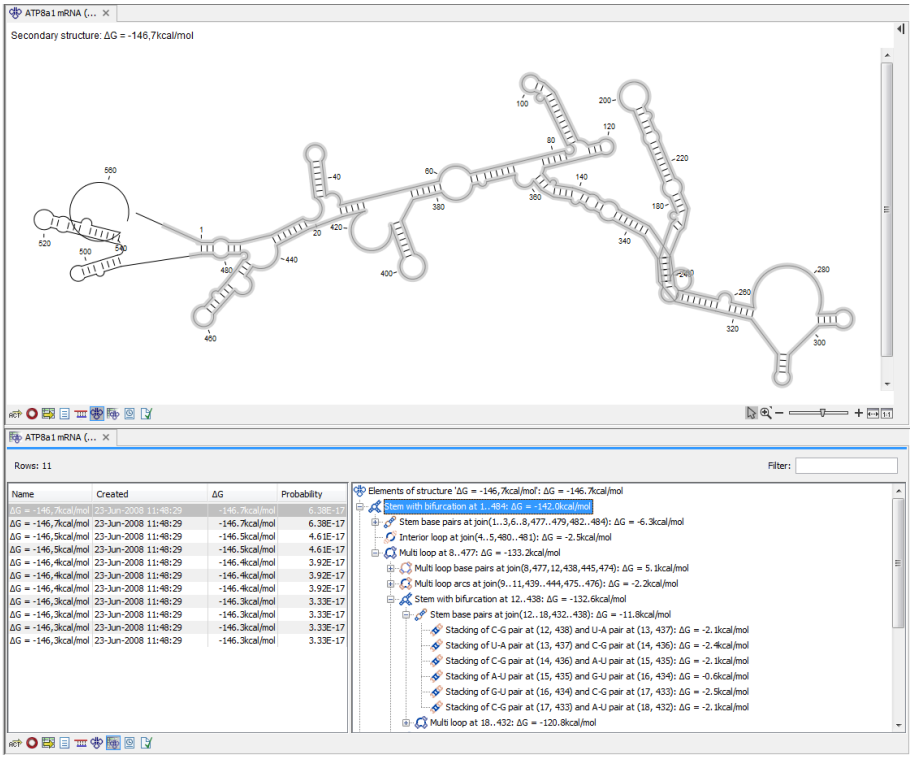


Figure 23.18: *Now the "Stem with bifurcation" node has been selected in the table and a corresponding selection has been made in the view of the secondary structure to the left.*

23.2.3 Symbolic representation in sequence view

In the **Side Panel** of normal sequence views (ACT), you will find an extra group under **Nucleotide info** called **Secondary Structure**. This is used to display a symbolic representation of the secondary structure along the sequence (see figure 23.19).

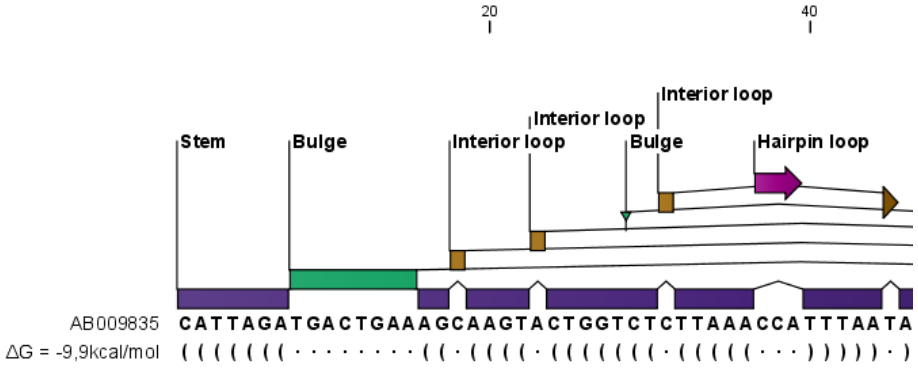


Figure 23.19: *The secondary structure visualized below the sequence and with annotations shown above.*

The following options can be set:

- **Show all structures.** If more than one structure is predicted, this option can be used if all the structures should be displayed.

- **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 23.19).
- **Unpaired symbol.** How bases which are not part of a base pair should be represented (see figure 23.19).
- **Height.** When you zoom out, this option determines the height of the symbols as shown in figure 23.20 (when zoomed in, there is no need for specifying the height).
- **Base pair probability.** See section 23.2.4 below).

When you zoom in and out, the appearance of the symbols change. In figure 23.19, the view is zoomed in. In figure 23.20 you see the same sequence zoomed out to fit the width of the sequence.

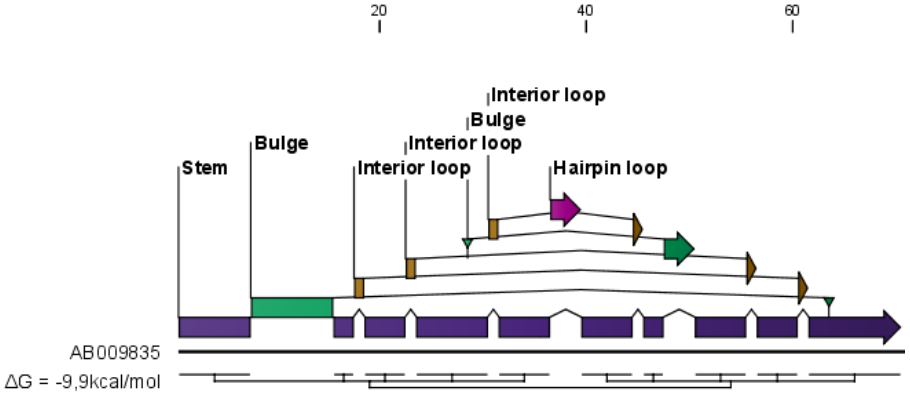


Figure 23.20: *The secondary structure visualized below the sequence and with annotations shown above. The view is zoomed out to fit the width of the sequence.*

23.2.4 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 23.4.

In the linear sequence view (), this is found in **Nucleotide info** under **Secondary structure**, and in the secondary structure 2D view (), 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%** (1:1) in order to see the coloring.

23.3 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 23.1.2. By adding several annotations complex structural hypotheses can be formulated (see 23.21).

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. 23.21.

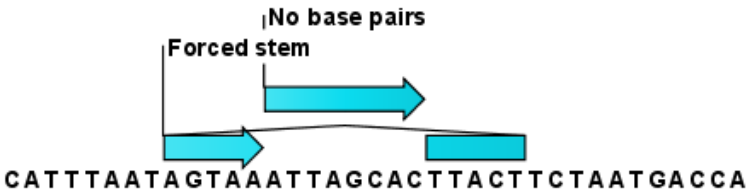


Figure 23.21: *Two constraints defining a structural hypothesis.*

23.3.1 Selecting sequences for evaluation

The evaluation is started from the **Toolbox**:

> **Toolbox** | **Classical Sequence Analysis** ( ) | **RNA Structure** ( )| **Evaluate Structure Hypothesis** ( )

This opens the dialog shown in figure 23.22.

If you have selected sequences before choosing the Toolbox action, they are 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. 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 23.23).

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).

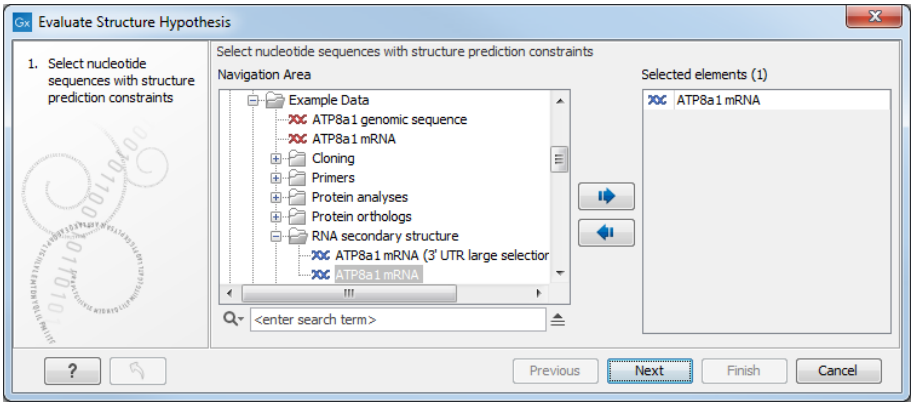


Figure 23.22: *Selecting RNA or DNA sequences for evaluating structure hypothesis.*

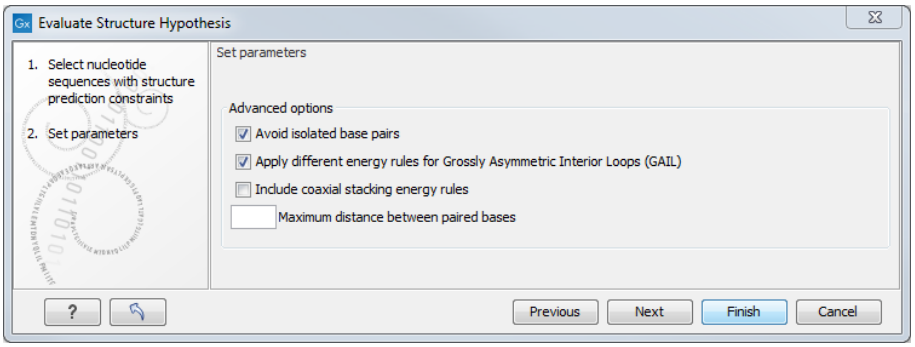


Figure 23.23: *Adjusting parameters for hypothesis evaluation.*

- **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].

23.3.2 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 23.24).

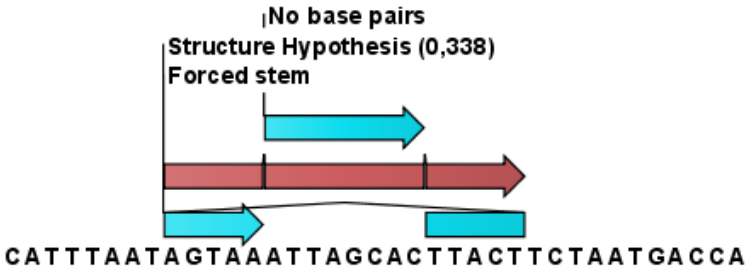


Figure 23.24: *This hypothesis has a probability of 0.338 as shown in the annotation.*

23.4 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].

23.4.1 Selecting sequences for scanning

The scanning is started from the **Toolbox**:

> **Toolbox** | **Classical Sequence Analysis ()** | **RNA Structure ()** | **Evaluate Structure Hypothesis ()**

This opens the dialog shown in figure 23.25.

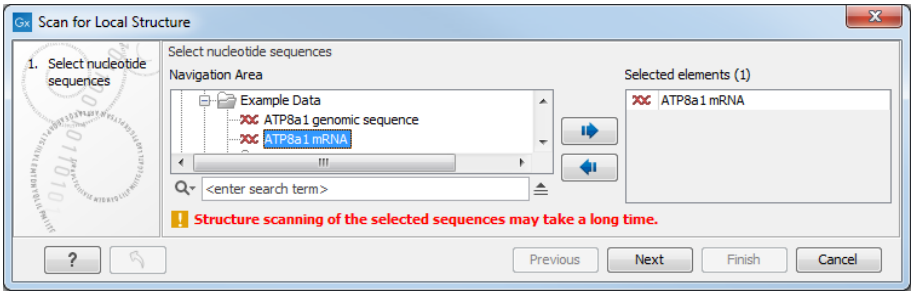


Figure 23.25: *Selecting RNA or DNA sequences for structure scanning.*

If you have selected sequences before choosing the Toolbox action, they are 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** to adjust scanning parameters (see figure 23.26).

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

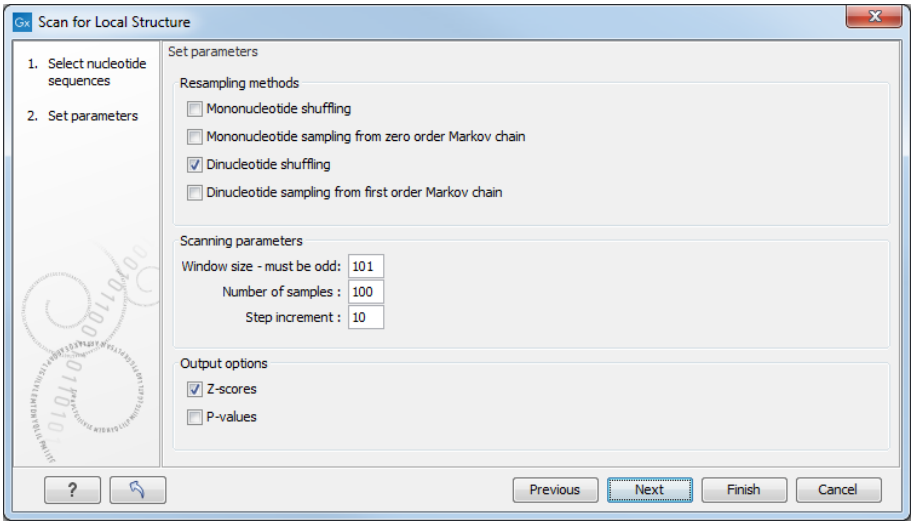


Figure 23.26: *Adjusting parameters for structure scanning.*

- **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.

23.4.2 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 23.27).

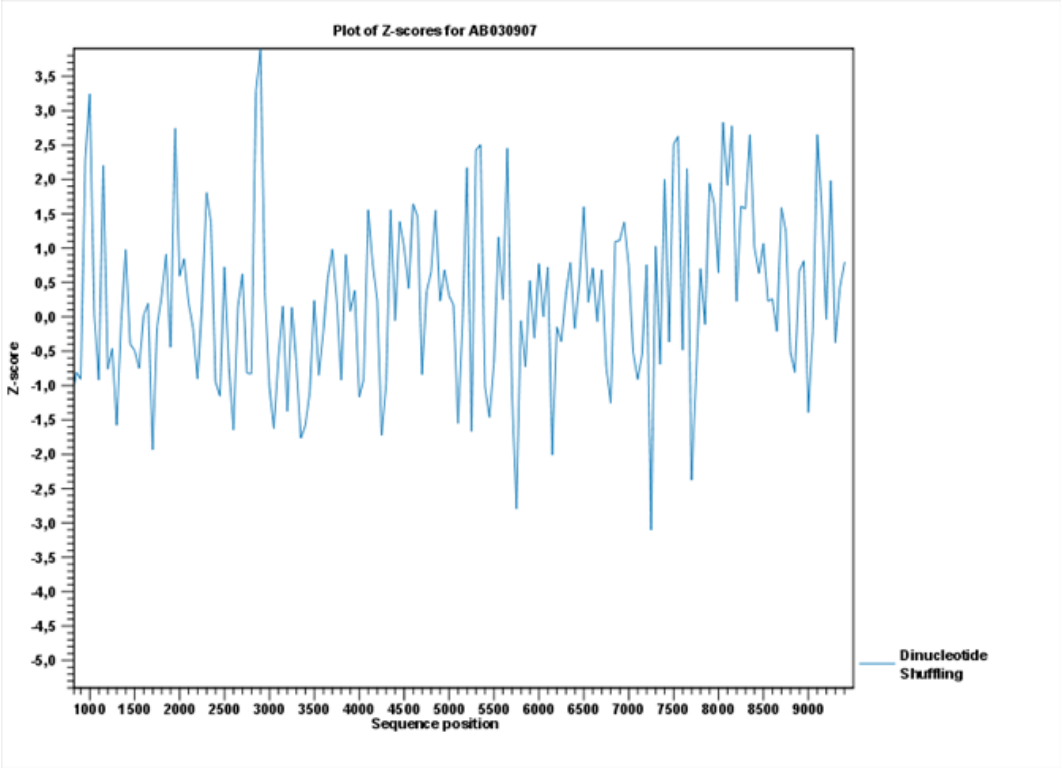


Figure 23.27: *A plot of the Z-scores produced by sliding a window along a sequence.*

23.5 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.

23.5.1 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 23.28.

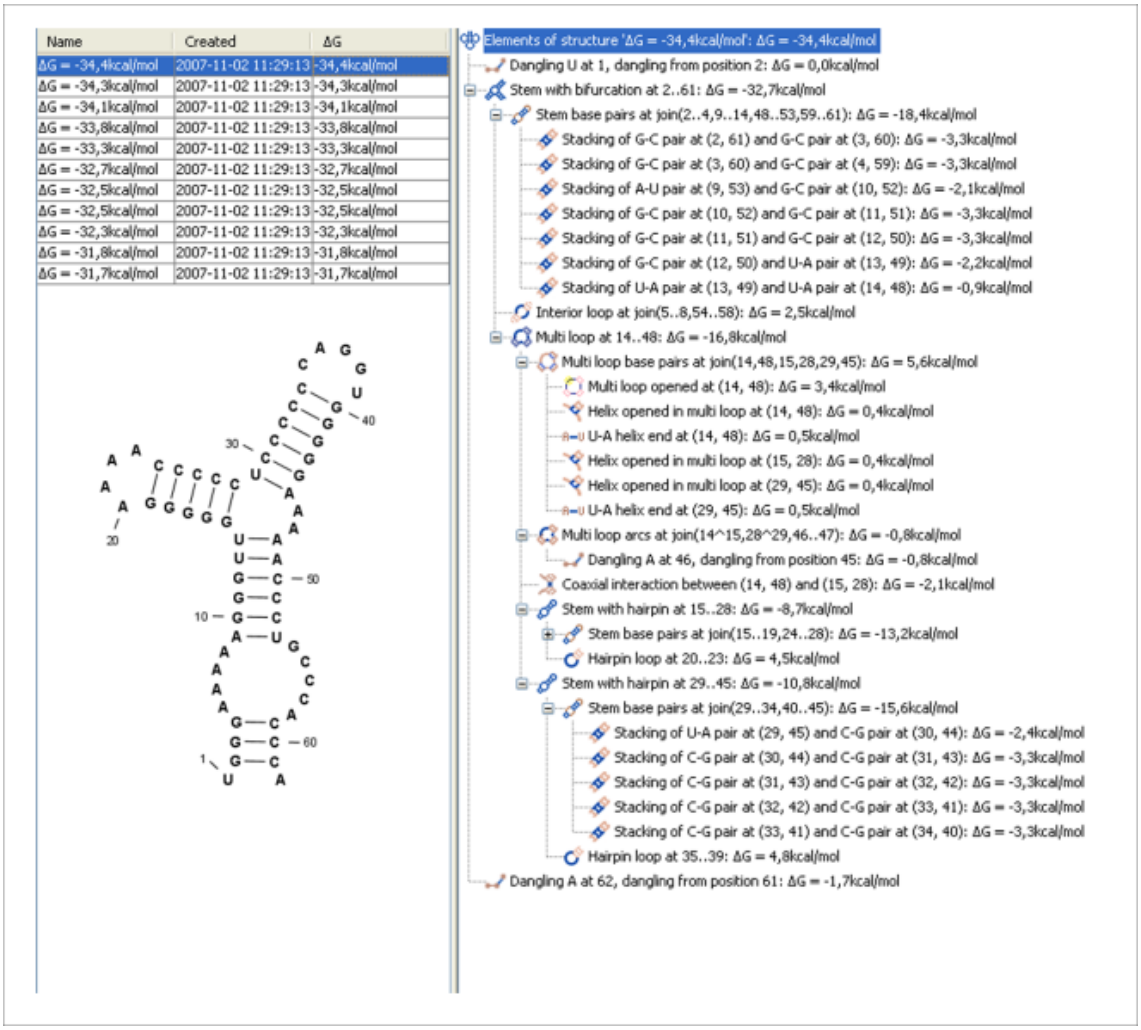


Figure 23.28: *A number of suboptimal structures have been predicted using* **CLC Genomics Workbench** *and are listed at the top left. At the right hand side, the structural components of the selected structure are listed in a hierarchical structure and on the left hand side the structure is displayed.*

23.5.2 Structure elements and their energy contribution

In this section, we classify the structure elements defining a secondary structure and describe their energy contribution.

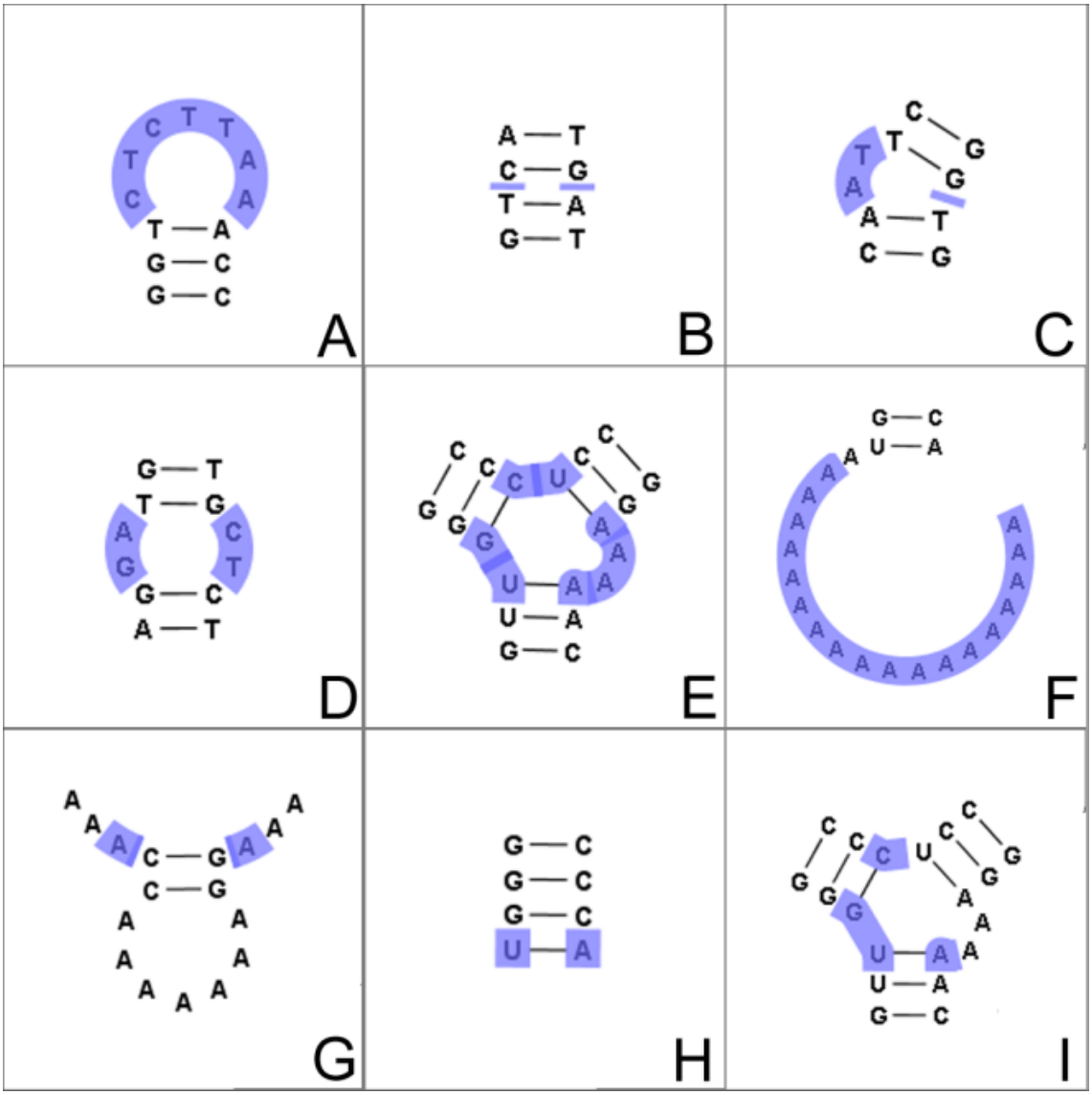


Figure 23.29: *The different structure elements of RNA secondary structures predicted with the free energy minimization algorithm in* **CLC Genomics Workbench**. *See text for a detailed description.*

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 pase 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 23.29A).

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 23.29B). 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 23.29C). 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 $i - j' > 1$ This means that the two base pairs enclose an unpaired region of length ≥ 1 on both sides (see figure 23.29D). 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 23.29E). 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 23.29F). 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 23.29G). 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 23.29H).
- **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 23.29I). 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 23.30) and it has been shown that this can dramatically increase the fidelity of the secondary structure prediction [Mathews and Turner, 2006].



Figure 23.30: *Known structural features can be added as constraints to the secondary structure prediction algorithm in* **CLC Genomics Workbench**.

Part IV

High-throughput sequencing

Chapter 24

Tracks

Contents

24.1 Track types . . . 570
24.2 Working with tracks . . . 572
24.2.1 Visualizing, zooming and navigating tracks . . . 572
24.2.2 Showing a track in a table . . . 574
24.2.3 The Chromosome Table view . . . 575
24.2.4 Finding annotations on the genome . . . 576
24.2.5 Extract sequences from tracks . . . 577
24.3 Track lists . . . 580
24.3.1 Adding, removing and reordering tracks . . . 581
24.3.2 Open track from a track list in table view . . . 582
24.3.3 Creating track lists in workflows . . . 582
24.4 Retrieving reference data tracks . . . 583
24.5 Merge Annotation Tracks . . . 583
24.6 Merge Variant Tracks . . . 584
24.7 Track Conversion . . . 585
24.7.1 Convert to Tracks . . . 585
24.7.2 Convert from Tracks . . . 586
24.8 Annotate and Filter . . . 588
24.8.1 Filter on Custom Criteria . . . 588
24.8.2 Annotate with Overlap Information . . . 589
24.8.3 Filter Annotations on Name . . . 589
24.8.4 Filter Based on Overlap . . . 590
24.9 Graphs . . . 591
24.9.1 Create GC Content Graph . . . 591
24.9.2 Create Mapping Graph . . . 593
24.9.3 Identify Graph Threshold Areas . . . 595

A track is the fundamental building block for NGS analysis in the *CLC Genomics Workbench*. The idea behind tracks is to provide a unified framework for the visualization, comparison and analysis of genome-scale studies such as whole-genome sequencing or exome resequencing projects and a variety of different -Seq data (i.e. RNA-seq, ChIP-Seq, DNAse-Seq).

In tracks, all information is tied to genomic positions, with the central coordinate system provided by a reference genome. Different types of data, and results for different samples, can be visualized and analyzed together as long as the tracks are compatible, i.e. they share the same coordinate system. Compatible tracks contain the same number of chromosomes, with the chromosome lengths being the same in each track. If two chromosomes have the same length, then the chromosome names must be identical for the tracks to be considered compatible.

Different types of data are represented in different types of tracks, and each type of track has its own particular editors. An example of a paired-end mapping read-track displaying reads and coverage is shown in figure 24.1.

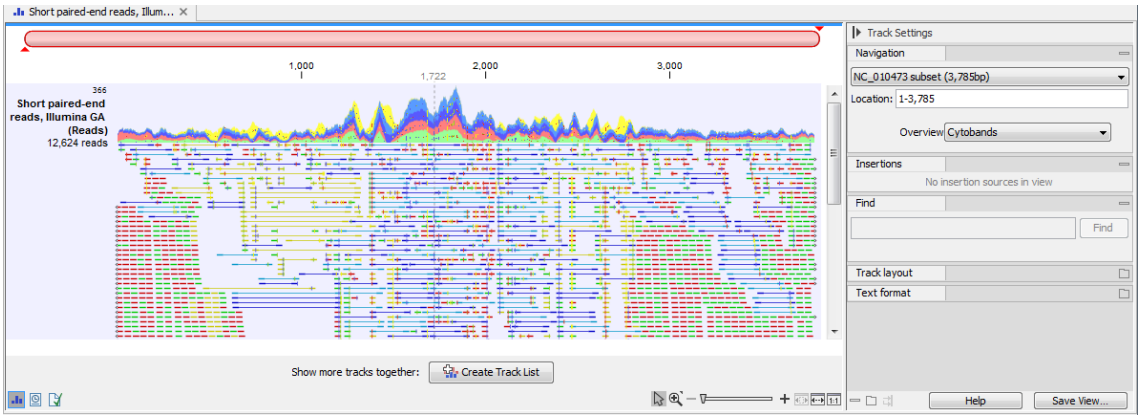


Figure 24.1: *A paired-end mapping read-track opened, displaying reads and coverage. Below the track, the button for creating a Track List is visible. On the right is the Side Panel.*

24.1 Track types

The different track types in the *CLC Genomics Workbench* are:

Sequence Track ( ) A sequence track contains one or more sequences and is usually used to store the reference sequences of a genome (e.g. the chromosomes or the consensus sequences of de novo assembled contigs).

Reads Track ( ) Reads Tracks hold read mappings such as the ones produced by the Map Reads to Reference (see section 27.1), Local Realignment (see section 27.3) or RNA-Seq Analysis (see section 30) tools. The reads track contains all the reads that have been mapped at their mapped positions, and you can zoom in all the way to base resolution. In case there are more reads than the height of the track allows, an overflow graph will be displayed above the reads in the same colors than the reads that it represents. When hovering over the reads, a vertical scroll bar will appear to the right of the reads to navigate through high coverage regions, and in non-aggregated view, a tooltip shows the read counts

for each observed nucleotide in that position together with the directions of the reads with that nucleotide. Read more about reads tracks here: section 27.2.2.

Variant Track ( ) A variant track (see section 28.6.1) stores features that fulfill the requirements for being a variant. A particular requirement for being a variant is that it refers to a particular region of the reference, and it is possible to describe exactly how the sample "Allele" sequence looks in this region, as compared to what the "Reference allele" sequence looks like in this region. Variants may be of type SNV, MNV, replacement, insertion or deletion. A variant track may be produced either by running a Variant detection tool in the Workbench, or by importing a variant format file (such as a VCF or a GVF file), or by downloading it from a database (e.g. dbSNP). Note on the InDels and Structural Variants (see section 28.10): this tool detects structural variants, including insertions, deletions, inversions, translocations and tandem duplications. It will produce a variant track, which will contain some insertions and deletions (the "InDel" track). However, the tool will also detect some insertions for which the "Allele" sequence is not fully, but only partially, known. These insertions do not fulfill the requirements of being a variant and therefore cannot be put in the variant track. Instead they are put in the "SV track", along with the inversions and translocations. The "SV" track is an "annotation" (or "feature") track, which is less strict and more flexible in the requirements to the types of annotations (or features) that it can contain (see below).

Annotation Track ( ) Each annotation track contains a certain type of annotations. Examples are gene or mRNA tracks, UTR tracks, conservation score tracks and target region tracks. They may be obtained either by importing a BED, GTF or GFF file, or by downloading a database, such as ENSEMBL, in the workbench using the Import | Tracks tool (see section 6.2). Annotation tracks can also be downloaded with the Reference Data Manager. Also, many of the tools in the Workbench will output annotation tracks: for example, the Indels and Structural Variants tool will put the detected structural variants (that do not fulfill the requirements for being of type "variant") in an annotation track called SV track, or the ChIP-Seq detection tool which will compile the detected "peaks" into a peak annotation track. Finally, a gene (annotation) track can be created (see 24.7.1Convert to Tracks). A description of how to annotate and filter tracks is found in section 24.8.

Coverage Graph Track ( ) The coverage graph track is calculated from a reds track and contains a graphical display of the coverage at each position in the reference.

Expression Track ( ) The RNA-seq algorithm produces expression tracks: one for genes and one for transcripts. These have an annotation for each gene or transcript, and an expression value associated to that annotation. The type of expression value associated with each annotation is determined by the expression value parameter selected in the RNA-Seq tool. These values are visualized as a color gradient from blue to red; the lowest expression value within each chromosome of the track is represented as 0% and the highest expression value within each chromosome of the track is represented by 100%.

An example of the different types of tracks is given in figure 24.2.

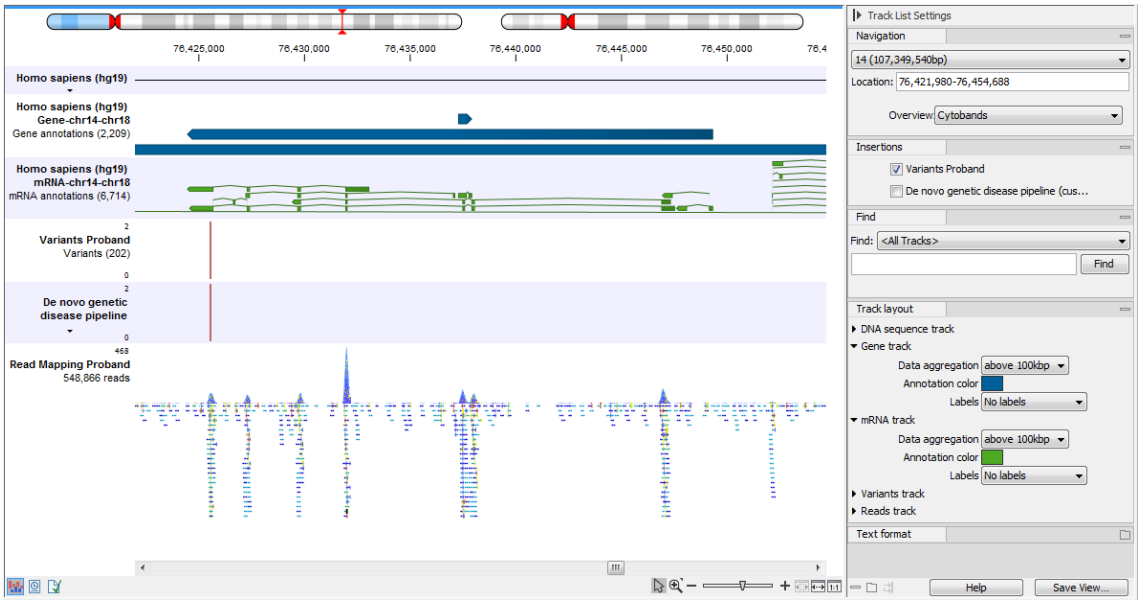


Figure 24.2: *A track list containing different types of tracks. From the top: a sequence track, three annotation tracks with gene, mRNA and CDS annotations respectively, two variant tracks, a gene-level (GE) and a transcript level (TE) expression track, a coverage track and a reads track.*

24.2 Working with tracks

24.2.1 Visualizing, zooming and navigating tracks

Side Panel

The Side Panel is shown to the right of a track opened in the View Area (figure 24.3). 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.

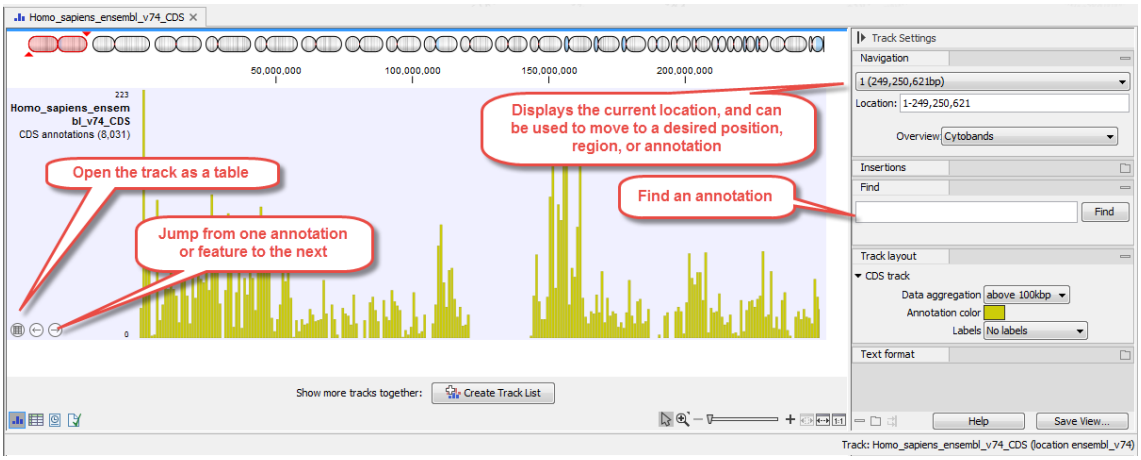


Figure 24.3: *A CDS track with its Side Panel visible to the right.*

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 here: section 27.2.2. Similarly, when working with annotation tracks, the settings **Track Layout | Labels** controls where labels will be displayed in relation to the annotations (figure 24.4).

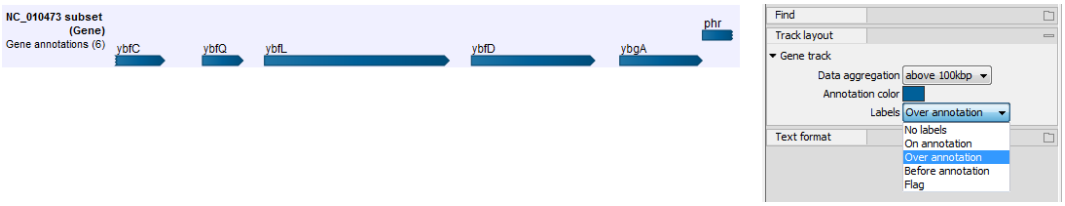


Figure 24.4: *The Side Panel for annotation tracks showing the Labels drop down menu.*

.

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.

Specified tracks have buttons that will appear under the track name on the left side of the View Area (highlighted in figure 24.5): these buttons can be used to open the track as a table, or jump to the previous and next element.

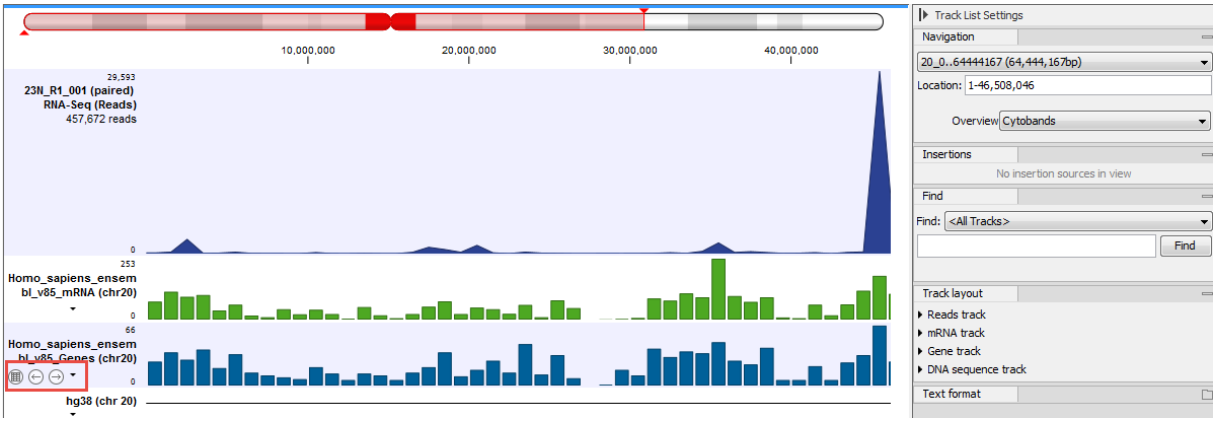


Figure 24.5: *Hovering on a track will show additional buttons under the track name.*

Zooming

It is possible to zoom in and out in a track using the buttons in the lower right-hand corner of the View Area.

- 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

24.2.2 Showing a track in a table

All tracks containing annotations (including variants) can be opened in a table. it is usually helpful to open the table in split view so that both the track and the table can be seen at the same time. In particular, because track and table are connected, it becomes possible to navigate and zoom the track by selecting successively the different rows in the table. Similarly, making a selection in the track view will select the corresponding row in the table. Buttons in the variant table (figure 24.6) view can filter the table to show only the region selected in the track. One can also create a new track from the rows selected in the table.

To open a track as a table in split view, press Ctrl (⌘ on Mac) while you click the table button at the bottom of the track view. You can also right-click on the track tab, and select "Show View | Table".

The table will have one row for each annotation, and the columns will reflect its information content. Figure 24.7 shows an example of a variant database track that is presented in a table.

You can use the table to sort, filter and select annotations (see Appendix 3.2). Please note that there are two additional options for *filtering on overlaps* in the "Region" column

When selecting a row in the table the graphical view will jump to this position on the genome. Please note that table filtering only affects the table. The track itself remains unaffected and keeps all annotations. If you also wish to filter tracks in the graphical view, the **Annotate and Filter** tools (see section 24.8) can be used instead.

At the bottom of the table a button labeled **Create Track from Selection** is available. This function can be used to create tracks showing only a subset of the data and annotations. Select the relevant rows in the table and click the button to create a new track that only includes the selected subset of the annotations. This function is particularly useful when used in combination with the filter.

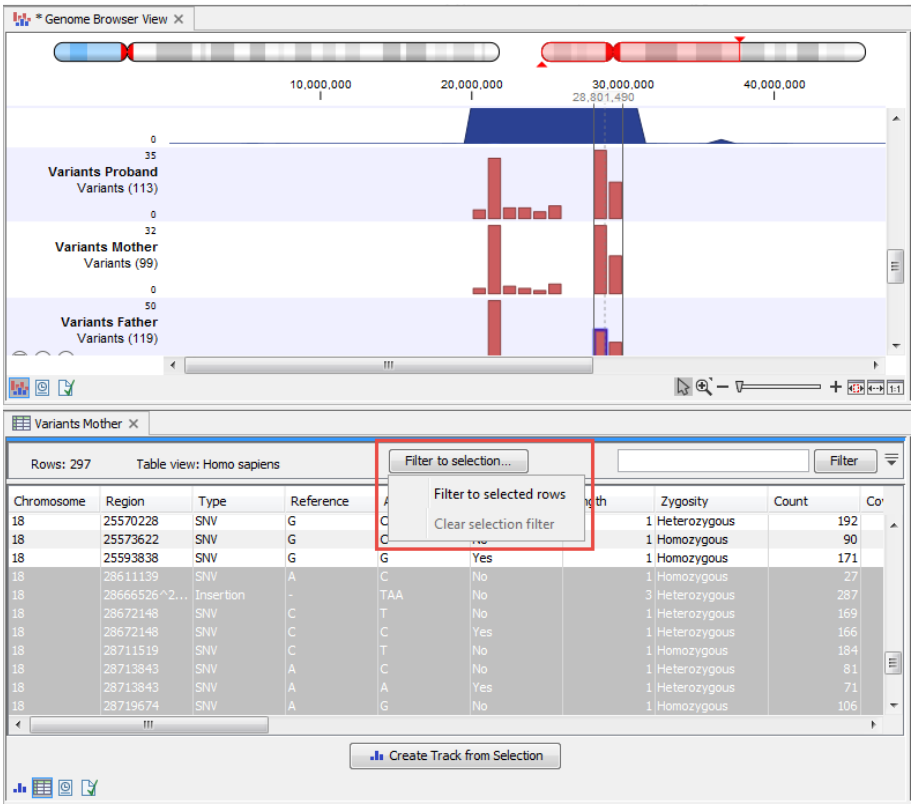


Figure 24.6: *Filter the table to display only the rows selected in the track suing the menu highlighted in red. It is also possible to create a new track from the selected rows.*

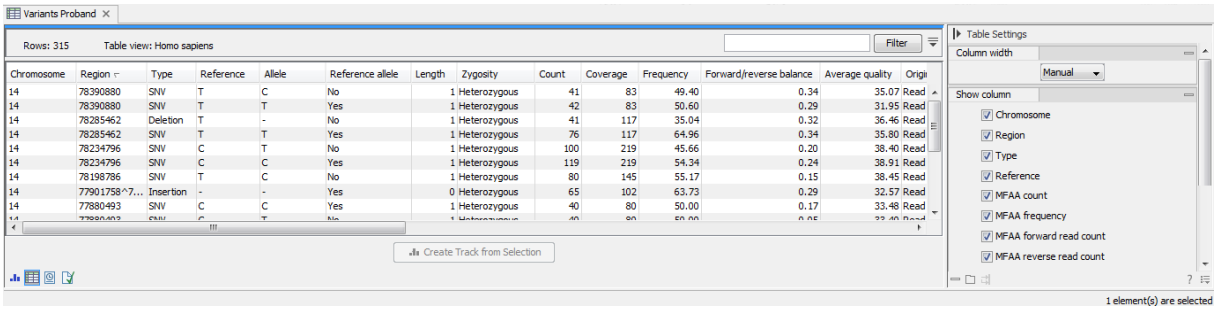
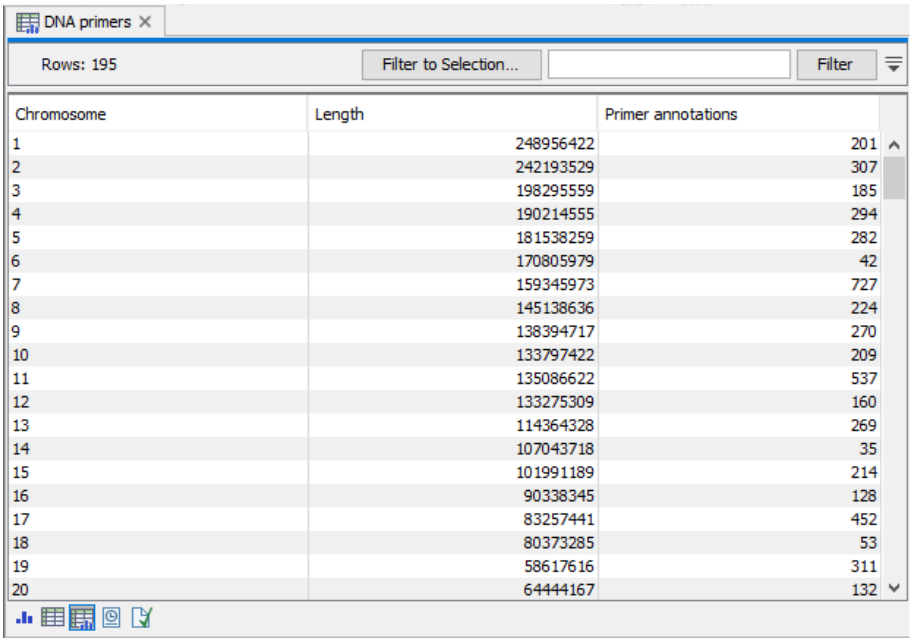


Figure 24.7: *Showing a variant track in a table view.*

24.2.3 The Chromosome Table view

The Chromosome Table view provides an overview of the data contained in tracks or track lists, summarized by chromosome (figure 24.8). 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 24.2.2 for details on opening 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 24.8: *The Chromosome Table view gives an overview of data contained in a track or a track list.*

24.2.4 Finding annotations on the genome

In the **Side Panel** under **Find**, a search field allows you to quickly find the annotation that you are looking for. The list of tracks further allows you to restrict the search to a particular track (e.g. a gene track).

In the search field you can enter any kind of text that exists in the annotation track. As an example, consider the gene and tool tip shown in figure 24.9.

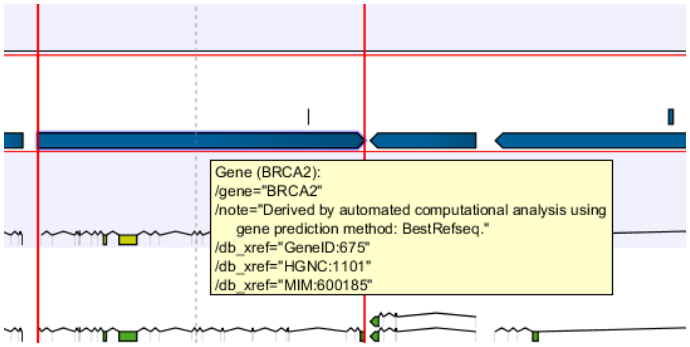


Figure 24.9: *The BRCA2 gene.*

This gene could be found by searching for the name specifically or, by using a wildcards (asterisks), less specific search terms could be use, giving more flexibility. To find BRCA2, for example, any of the following entries could be typed in the search field:

- **BRCA2** This would match the annotation name exactly.
- **BRCA*** This would match the annotation name as well as other genes with a text starting with BRCA (e.g. the BRCA1 gene).

- ***RCA2** This would match the annotation name as well as other genes with a text ending with RCA2 (e.g. the SMARCA2 gene).
- **600185** This would match the `db_xref` qualifier for the OMIM database. All the text shown for the annotation in figure 24.9 can be searched this way, both as exact matches and with the * before or after the search term.

Just below the search field in the **Side Panel**, a status label informs about the progress of the search and the hit that has been found. Placing the mouse on top of the label will display a tooltip with more info (see 24.10).

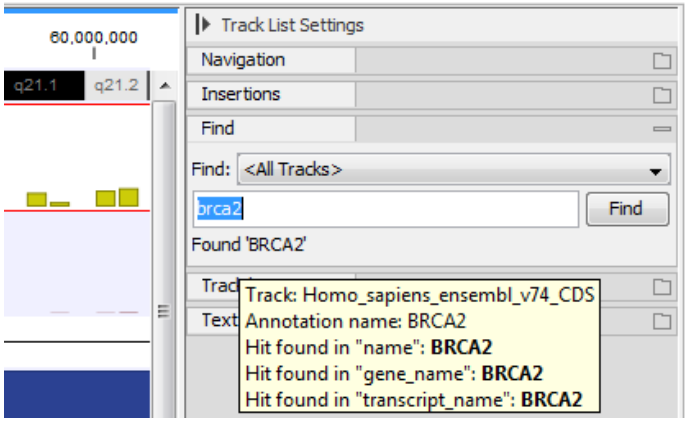


Figure 24.10: *The BRCA2 gene found.*

The search will be performed throughout the entire genome beginning with the chromosome currently shown and stopping when it finds the first hit. Press **Find** again to find the next hit. Once the whole genome has been traversed, the status will inform you that you have searched the whole genome. Click the **Find** button to start the search again.

Please note that you can also use the table view of an annotation track to perform more advanced queries of the data. When clicking on the annotation of interest in the table, the track list will zoom to that annotation.

Hovering the mouse cursor over an annotation in the track will display (figure 24.11):

- in the View Area, a tooltip with information about the annotation.
- in the ruler at the top of the Track list view: the position of the mouse cursor relative to the reference, the name of the annotation, the strand of the annotation, which exon is currently being hovered over, and the location of the mouse cursor relative to the annotation start (ignoring introns).
- in the lower right corner of the workbench, the annotation type, the relative position of the mouse cursor in the annotation, the strand, and the location of the mouse cursor in the reference.

24.2.5 Extract sequences from tracks

Note that the functionalities described in this page are valid for sequence or reads tracks. For similar functionalities on read mappings, see section 19.7.6.

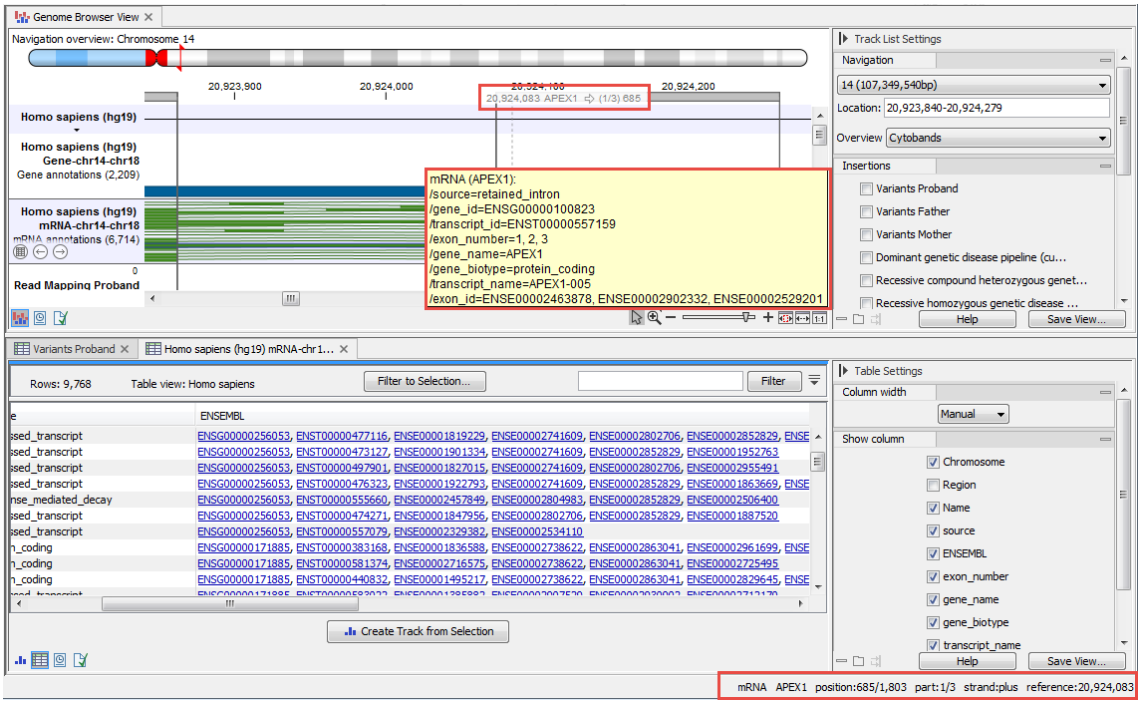


Figure 24.11: *When hovering on an annotation in a track view, relevant information about this annotation can be found in three different locations of the workbench.*

Extracting the sequence from a sequence track It is possible to extract a DNA sequence from a sequence track included in a Track List with the **Extract Sequence...** option of the right-click menu on the sequence (menu to the right of figure 24.12). In this case, a pop up window proposes to **Extract annotations**, which will extract all the annotations present on the other tracks of the Track list, and add them to the extracted sequence.

Note that it is possible to extract the sequence of a single sequence track, but in this case (the track is not included in a track list), or when the sequence track is in a track list that does not contain annotation tracks, the option to Extract annotations is unavailable.

Extracting a single read/sequence from a reads track Right-click on the sequence of interest and choose the **Selected read**... option to Copy, Open in a new view or Blast the selected sequence.

Extracting all reads/sequences from a track Use the tool Extract Sequences from the Toolbox to extract sequences from the subset reads track as described in section 15.1.

Extracting only selected reads/sequences from a track The sequences of interest can be selected by dragging the mouse over the region of interest, followed by a right click on the reads (or on the sequences in the case of a sequence track) and a click on **Create Reads Track from Selection** (as can be seen on the menu to the left in figure 24.12).

An Extract from Selection pop up dialog lets you specify what kind of reads you want to include in the subset of the original reads track (figure 24.13).

Per default all reads are included. The options are:

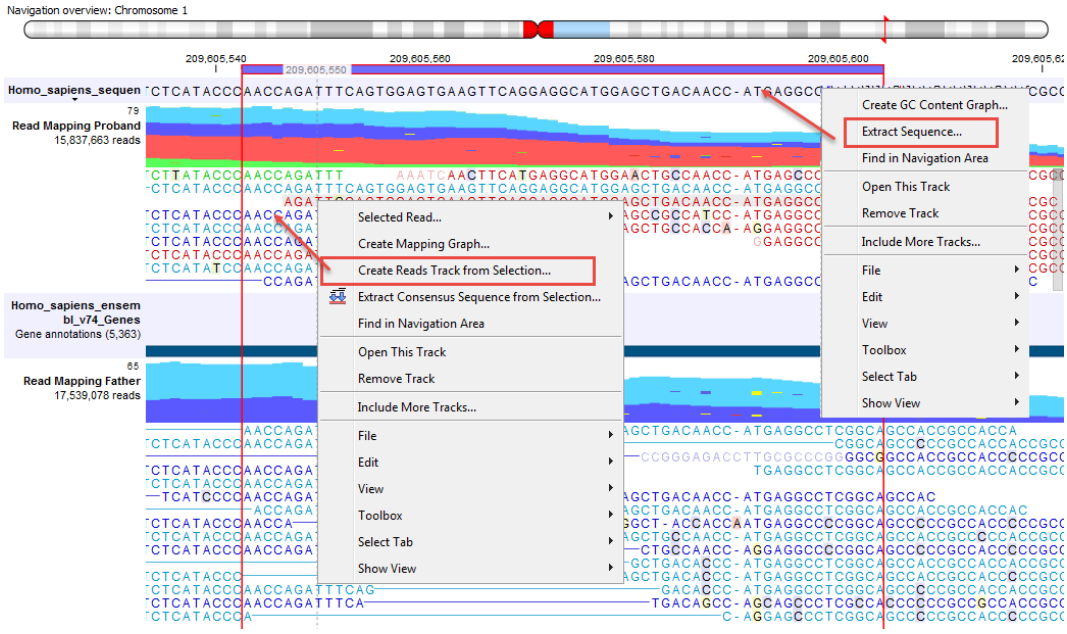


Figure 24.12: *Extract sequences from a read mapping track. This screenshot shows the menus available when right-clicking on the reference sequence and the sequences/reads in the tracks below.*

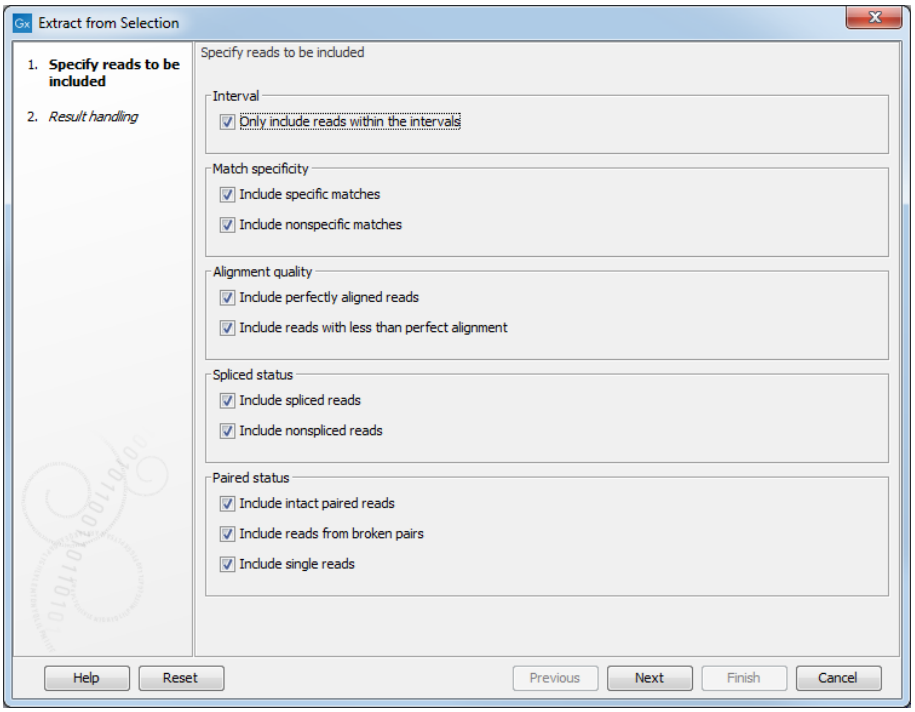


Figure 24.13: *Selecting the reads to include.*

- **Paired status**
  - **Include intact paired reads**. When paired reads are placed within the paired distance specified, they will fall into this category. Per default, these reads are colored in blue.
  - **Include paired reads from broken pairs**. When a pair is broken, either because only one read in the pair matches, or because the distance or relative orientation is wrong, the reads are placed and colored as single reads, but you can still extract them by

checking this box.

  - **Include single reads**. This will include reads that are marked as single reads (as opposed to paired reads). Note that paired reads that have been broken during assembly are not included in this category. Single reads that come from trimming paired sequence lists are included in this category.

- **Match specificity**

  - **Include specific matches**. Reads that only are mapped to one position.
  - **Include non-specific matches**. Reads that have multiple equally good alignments to the reference. These reads are colored yellow per default.

- **Alignment quality**

  - **Include perfectly aligned reads**. Reads where *the full read* is perfectly aligned to the reference sequence (or consensus sequence for de novo assemblies). Note that at the end of the contig, reads may extend beyond the contig (this is not visible unless you make a selection on the read and observe the position numbering in the status bar). Such reads are not considered perfectly aligned reads because they don't align in their entire length.
  - **Include reads with less than perfect alignment**. Reads with mismatches, insertions or deletions, or with unaligned nucleotides at the ends (the faded part of a read).

- **Spliced status**

  - **Include spliced reads**. Reads that are across an intron.
  - **Include non spliced reads**. Reads that are not across an intron.

24.3 Track lists

For details on how to find and import different tracks see section 6.2. Tracks are saved as files in the **Navigation Area** with specific icons representing each track type, e.g. an annotation track ().

To visualize several tracks together, they can be combined into a **Track List** (). Track lists can be created in different ways. One way is via the menu bar:

File | **New** | **Track List ()**

Another way is to use the Track Tool **Create Track List** (). Finally, tracks can be created directly using the button labeled **Create Track List** that is found in the top right corner of the open track in the view area. Figure 24.14 shows an example of a track list including a track with mapped reads at the top, followed by a variant track, and in the lower part of the figure, the reference sequence with CDS annotations.

The track list is designed to be used as a container for multiple tracks for easy visualization and comparative analysis. Therefore all the involved tracks and the track list are required to be present and located in one single location (Workbench or CLC Server). Otherwise, they will be marked as "Unresolved track" in the track list.

The **Chromosome Table view** (see section 24.2.3) summarizes the data contained in the tracks included in the track list.

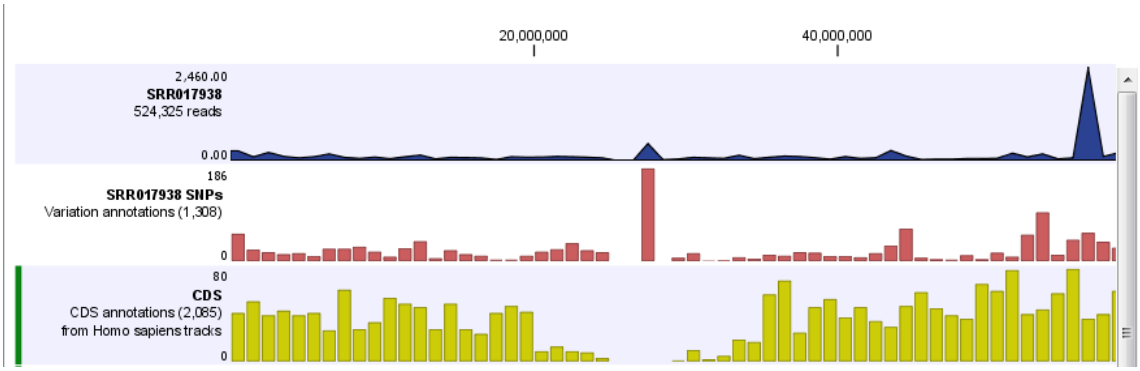


Figure 24.14: *Three tracks shown in the track list view*

24.3.1 Adding, removing and reordering tracks

You can organize your tracks by dragging them up and down. Right-clicking on any of the tracks opens up a context menu with several options (Figure 24.15). The options shown in the context menu will vary depending on which tracks you have open in the viewing area. Hence, you may not be presented with all the options described here.

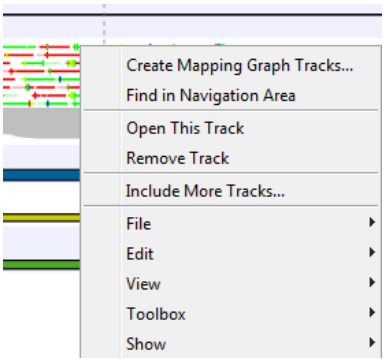


Figure 24.15: *Options to handle and organize tracks.*

- **Create Mapping Graph Tracks**. This will allow you to create a new track from a mapping track (learn more in section 24.9).
- **Find in Navigation Area**. This will select the track in the **Navigation Area**.
- **Open This Track**. This opens a new view of the track. For annotations and variant tracks, a table view is opened as described in section 24.2.2. This can also be accomplished by double-clicking the track.
- **Remove Track**. This will remove the track from the current view. You can add it again by dragging it from the **Navigation Area** into the track list view or by pressing **Undo** ( ).
- **Include More Tracks**. This will allow you to add other track sets to your current track set. Please note that the information in the track will still be stored in its original track set. This means that you by including a track in this way at the same time is adding a reference to this track in another track set. An example of this could be the inclusion of a SNP track from another sample to your current analysis.

24.3.2 Open track from a track list in table view

To open a table view of a track that is part of a track list, open the track list by double-clicking on the track name in the **Navigation Area**. The track will open in a graphical view. To open a single track from the track list in table view, either right-click on the track and choose "Open This Track" (see figure 24.16) or double-click on the name of the track you would like to open in table view (in the left side of the track when it is open in the **View Area**. This will automatically open op the specific track in table view.

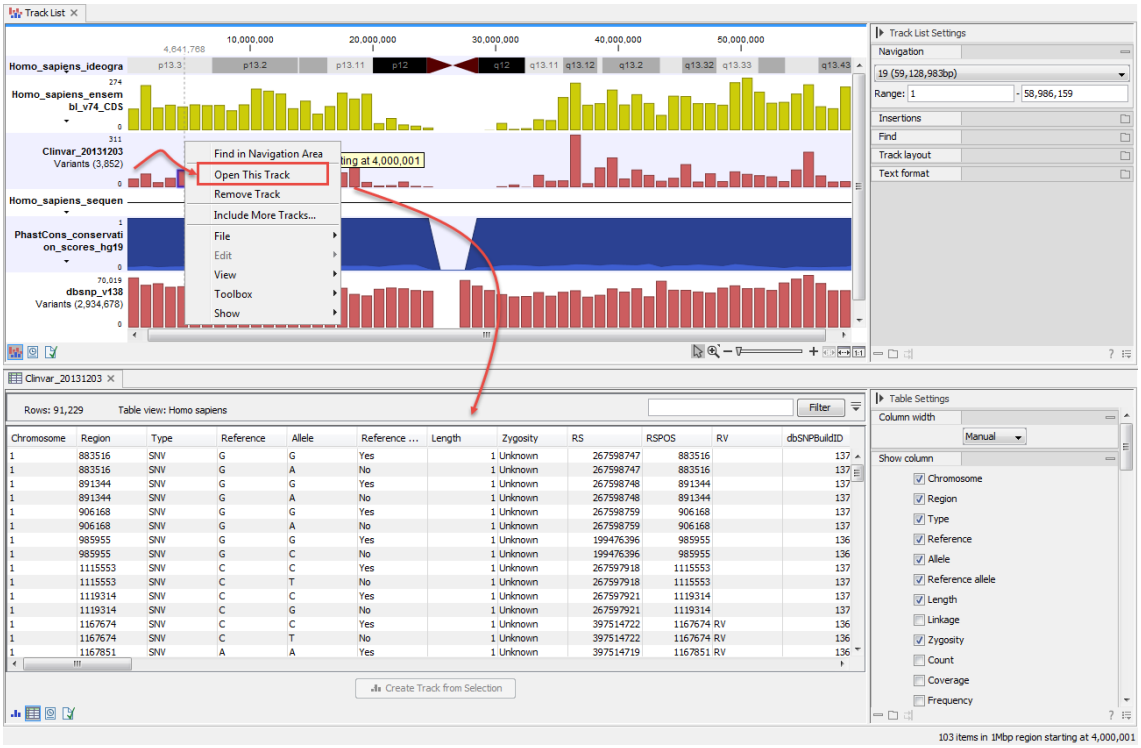


Figure 24.16: *One way to open a table view of a track that is part of a track list is to right click on the track of interest and select "Open This Table".*

24.3.3 Creating track lists in workflows

Track lists can be created as part of workflows. Track lists are different from all other workflow outputs in the sense that the tracks inside the track lists have to be saved separately, even if they are included in a track list.

Figure 24.17 shows an example of a workflow where two tracks are fed into the **Create Track List** element.

In the left hand side example, there is a warning at the bottom of the editor pointing at the fact that these two tracks need to be selected as output in order for the workflow to be validated. To the right, the workflow has been corrected by selecting the tracks as output, and the workflow can now be executed.

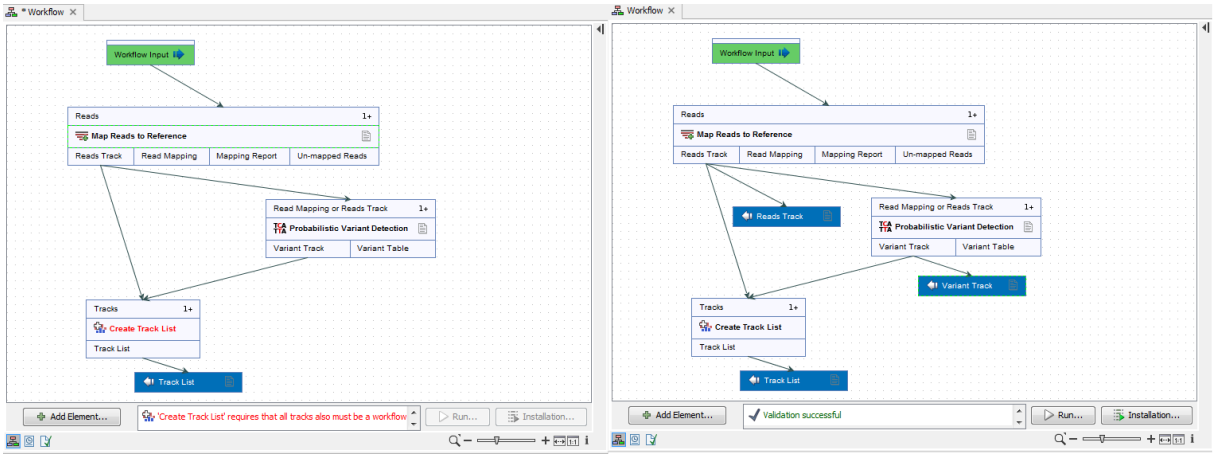


Figure 24.17: *This workflow does not work because the two tracks need to be marked as output.*

24.4 Retrieving reference data tracks

For most applications (except de novo sequencing), you will need reference data in the form of a reference genome sequence, annotations, known variants etc. There are three basic ways of obtaining reference data tracks:

1. Use the integrated tool for downloading reference genomes as tracks (see section 8.1).
2. Import tracks from files (see section 6.2).
3. Convert sequences with annotations to tracks (learn more in section 24.7). Sequences can come from a variety of sources:

   Standard Import ( ) The standard import accepts common data formats like fasta, genbank etc. (learn more in section 6)

   Downloading from NCBI The integrated tool for searching and downloading data from NCBI (learn more in section 7.1).

   Contigs created from de novo assembly. Contig sequences from de novo assembly (see section 32.2) can be considered a reference genome for e.g. subsequent resequencing analysis applications.

Please note that tracks are not yet supported with all transcriptomics tools of *CLC Genomics Workbench*. In this case, you have to provide standard sequences (downloading from NCBI or importing files).

24.5 Merge Annotation Tracks

Two tracks can be merged using the **Merge Annotation Tracks** tool:

Toolbox | **Track Tools** ( ) | **Merge Annotation Tracks** ( )

Select two or more tracks to be merged. The tracks have to be of the same type, e.g. gene tracks, and be based on the same genome.

Click **Next** and **Finish** to merge the tracks.

If the same annotation is found in both tracks, it is merged into one. If the two annotations share the same region (i.e. same coordinates), they are merged. Information from both annotations are retained in the output. Annotations are merged even when their names differ. Which name is being kept is entirely based on the order of the input tracks as the name being kept is derived from the track that was selected first in the **Merge Read Mappings** wizard. An extra column labeled **Origin tracks** is added to the resulting track indicating which track the annotation originates from. The Merge Annotation Tracks tool is useful when merging gene tracks from different sources using different naming conventions.

Please note that this tool is not well-suited for comparing tracks (see section 29.3 instead).

24.6 Merge Variant Tracks

Merge Variant Tracks merges variants from two or more variant tracks based on the same reference genome to a single track.

To run the tool, go to:

Toolbox | **Track Tools** ( ) | **Merge Variant Tracks** ( )

Select two or more variant tracks, based on the same reference genome, as input.

In the next wizard step, the following options can be configured (figure 24.18:

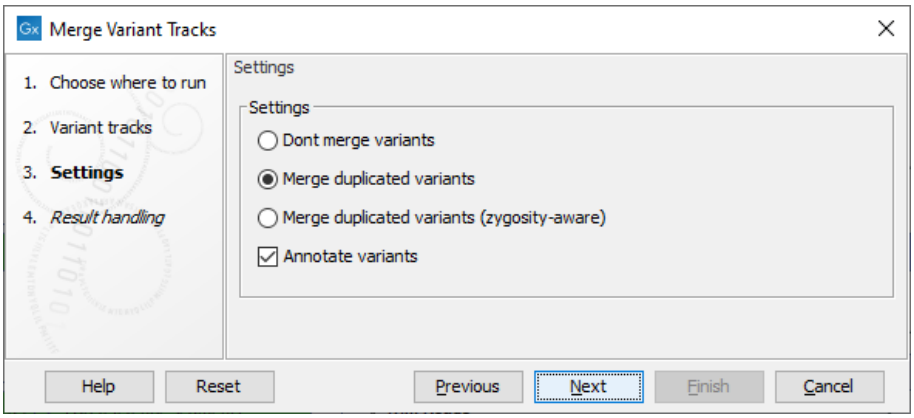


Figure 24.18: *Merge variant Tracks settings*

- **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 29.3 for information about tools designed to support variant comparison.

24.7 Track Conversion

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.

24.7.1 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:

Toolbox | **Track Tools** ( ) | **Track Conversion** ( ) | **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 24.19).

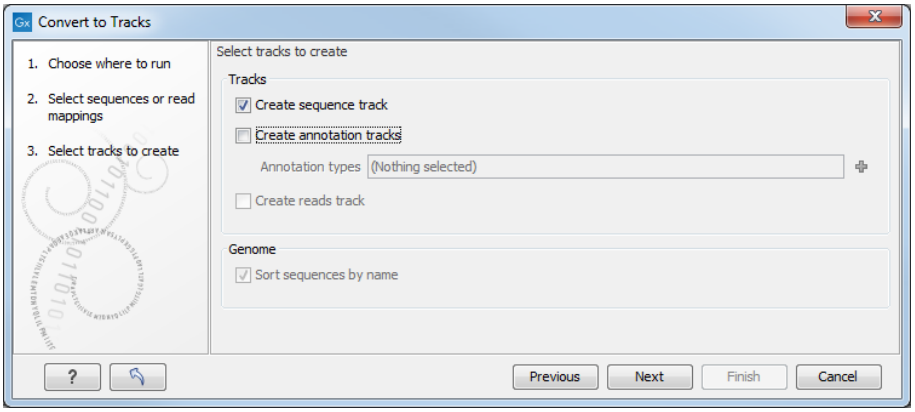


Figure 24.19: *Converting data to tracks.*

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.

24.7.2 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:

Toolbox | **Track Tools** ( ) | **Track Conversion** ( ) | **Convert from Tracks** ( )

One or more tracks can be used as input. In the example given in figure 24.20 a reads track and two annotation tracks are converted simultaneously to an annotated read mapping (figure 24.21).

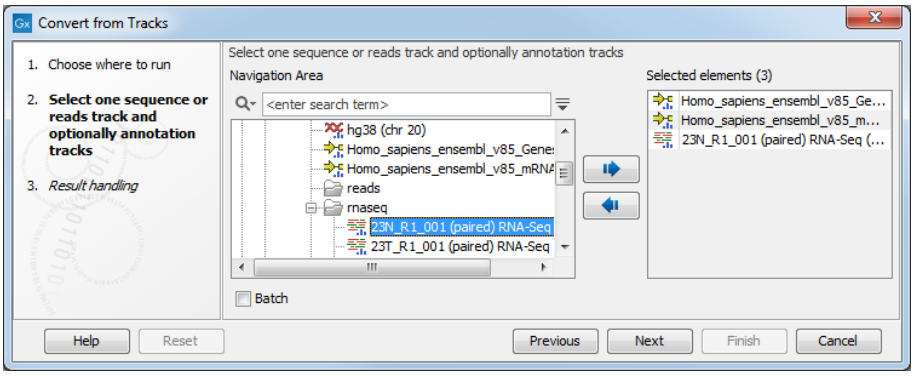


Figure 24.20: *A reads track and two annotation tracks are converted from track format to stand-alone format.*

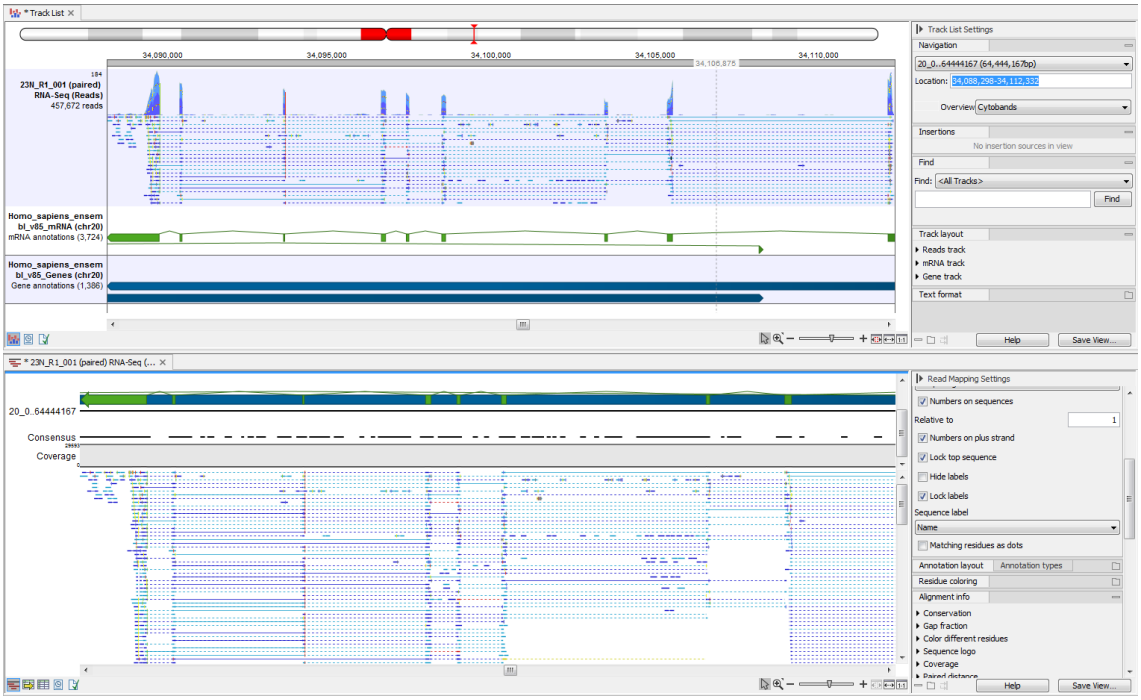


Figure 24.21: *The upper part of the figure shows the three individual input tracks, arranged for simplicity in a track list. The lower part of the figure shows the resulting stand-alone annotated read mapping.*

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 24.22 where one reference and two annotation tracks are used as input.

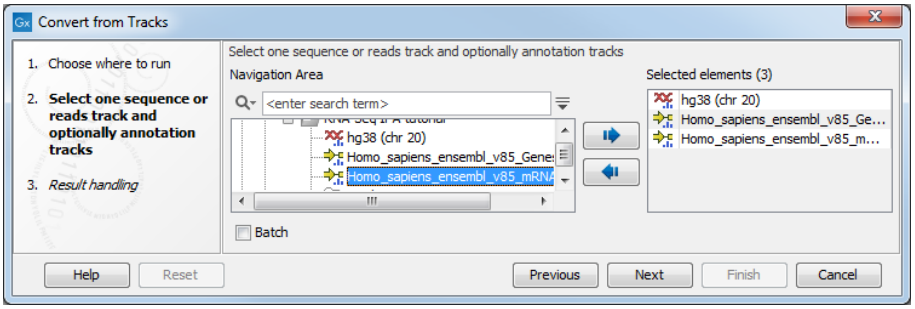


Figure 24.22: *A reference track and two annotation tracks are converted from track format to stand-alone format.*

The output is shown in figure 24.23. The reference sequence has been transformed to stand-alone format with the two annotations "CDS" and "Gene".



Figure 24.23: *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.*

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.

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 24.21.

24.8 Annotate and Filter

One of the big advantages of using tracks is that tracks support comparative analysis between different kinds of data. This section describes generic tools for annotating and filtering tracks (for filtering and annotating variants, please refer to chapter 29).

24.8.1 Filter on Custom Criteria

The **Filter on Custom Criteria** tool can be used to identify and extract variants or annotations that fulfill certain criteria. It will create a filtered copy of the track used as input containing only the annotations that were not filtered away.

To run the Filter on Custom Criteria, go to:

Toolbox | **Track Tools** ( ) | **Annotate and Filter** ( ) | **Filter on Custom Criteria** ( )

In the first step, select a variant track or an annotation track as input and click **Next**.

In the Filter criteria dialog, use the browse icon ( ) to specify the variant or annotation track that contains at least all the criteria you want to filter the first track on. Once specified, click on the **Load Annotations** button to create the drop down menu available in the filter field below. Note that in this context, we call this track a guidance track because it will define what are the filter criteria available; in the dialog, Load Annotations will transform every column headers of the guidance track in a filtering criteria.

An example of filter criteria is shown in figure 24.24. In this example, we filter away all variants that are not found on chromosome 1 and that are not homozygous. As a result you will only keep the variants that are homozygous and found on chromosome 1. The bottom filter field on the figure shows the various other filter criteria available from the guidance variant track.

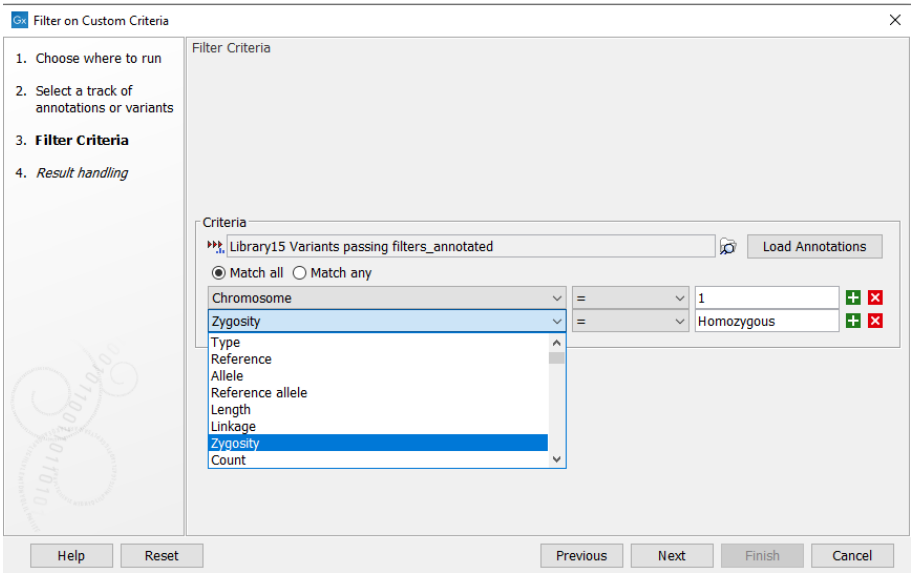


Figure 24.24: *Create filter criteria to extract the variants of interest. In this example we will extract the homozygous variants found on chromosome 1.*

The output from the **Filter on Custom Criteria** tool is a variant or annotation track (and associated table) that contains only the variants or annotations that fulfill the specified filter criteria.

24.8.2 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:

> **Toolbox** | **Track Tools** ( ) | **Annotate and Filter** ( ) | **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 24.25).

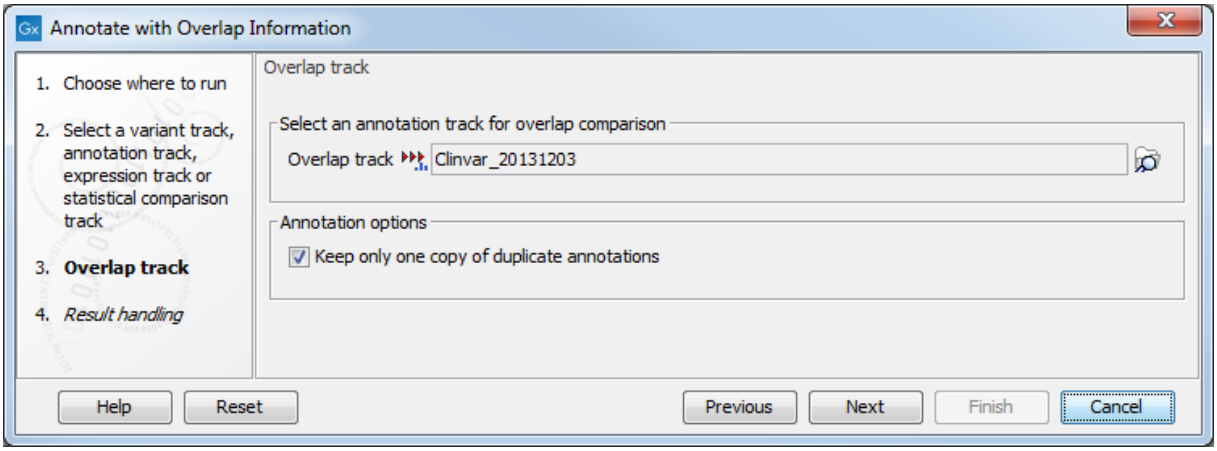


Figure 24.25: *Choose an overlap track with which you wish to annotate your input file.*

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.

24.8.3 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.

The proposed workflow would be to first create a new gene track only containing the genes of interest. This is done using this tool. Next, use the filter from the overlapping annotations tool (see section 24.8.4) to filter the variants based on the track with genes of interest.

> **Toolbox** | **Track Tools** ( ) | **Annotate and Filter** ( ) | **Filter Annotations on Name** ( )

Select the track you wish to filter and click **Next**.

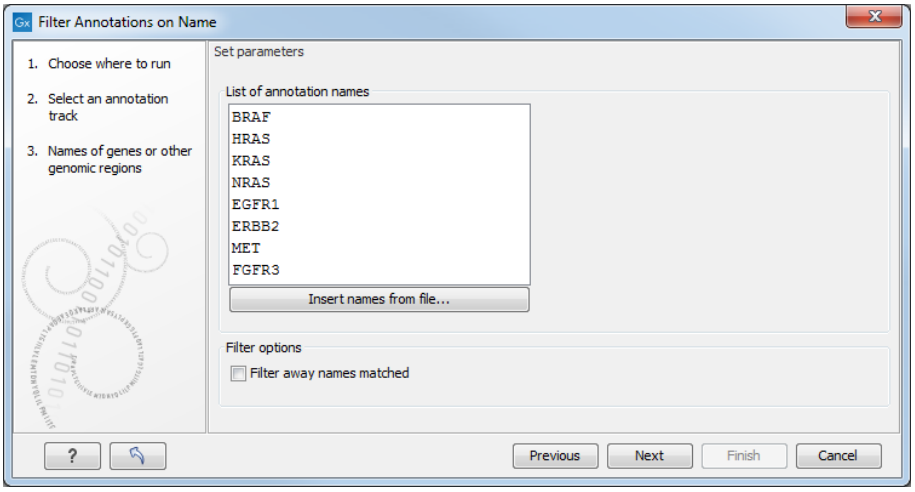


Figure 24.26: *Specify names for filtering.*

As shown in figure 24.26, 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).

24.8.4 Filter Based on Overlap

The overlap filter will be used for filtering an annotation track based on an overlap with another annotation track. This can be used to e.g. only show variants that fall within genes or regulatory regions or for restricting variants results to only cover a subset of genes as explained in section 24.8.3. Please note that for comparing variant tracks, more specific filters should be used (see section 29.1.1).

If you are just interested in finding out whether one particular position overlaps any of the annotations, you can use the advanced table filter and filter on the region column (track tables are described in section 24.2.2).

Toolbox | **Track Tools** ( ) | **Annotate and Filter** ( ) | **Filter Based on Overlap** ( )

Select the track you wish to filter and click **Next** to specify the track of overlapping annotations (see figure 24.27).

Next, select the track that should be used for comparison and tick whether you wish to keep annotations that overlap, or whether to keep annotations that do not overlap with the track selected. An overlap has a simple definition – if the annotation used as input has at least one shared position with the other track, there is an overlap. The boundaries of the annotations do not need to match.

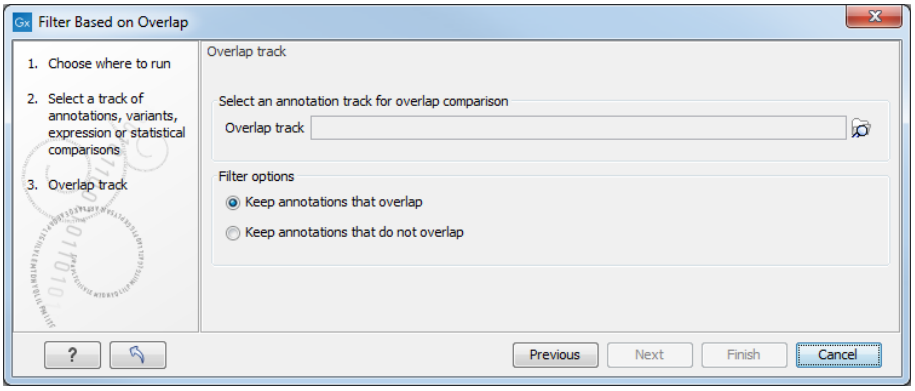


Figure 24.27: *Select overlapping annotations track.*

24.9 Graphs

Graphs can be a good way to quickly get an overview of certain types of information. This is the case for the GC content in a sequence or the read coverage for example. The *CLC Genomics Workbench* offers two different tools that can create graph tracks from either a sequence or a read mapping, respectively **Create GC Content Graph** and **Create Mapping Graph**.

Graph tracks can also be created directly from the track view or track list view by right-clicking the track you wish to use as input, which will give access to the toolbox.

To understand what graph tracks are, we will look at an example. We will use the **Create GC Content Graph** tool to go into detail with one type of graph tracks.

24.9.1 Create GC Content Graph

The **Create GC Content Graph** tool needs a sequence track as input and will create a graph track with the GC contents of that sequence.

To run the tool go to the toolbox:

Toolbox | **Track Tools** ( ) | **Graphs** ( ) | **Create GC Content Graph** ( )

Select the sequence track that should be used as input (see figure 24.28).

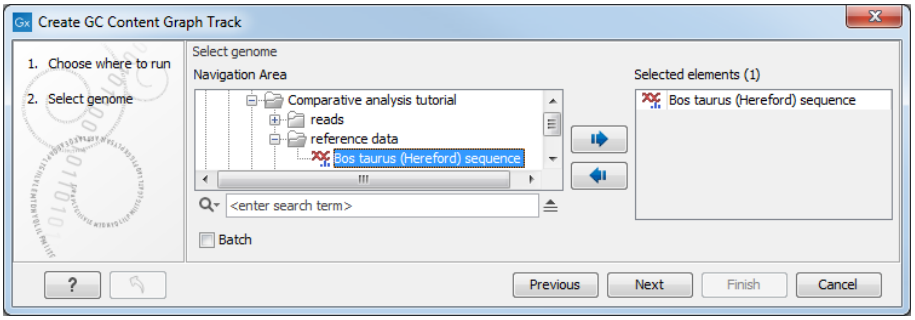


Figure 24.28: *Select the sequence track that should be used as input.*

In the next wizard step (see figure 24.29), you can specify the window size, i.e., the size of the window around the central base in the region that is used to calculate the GC content. This number must be odd as you need a central base and an equal number of bases to each side of the central base. For example, with a window size of 25, the GC content for the central base will be calculated based on the nucleotide composition in the central base and the 12 bases

upstream and 12 bases downstream of the central base.

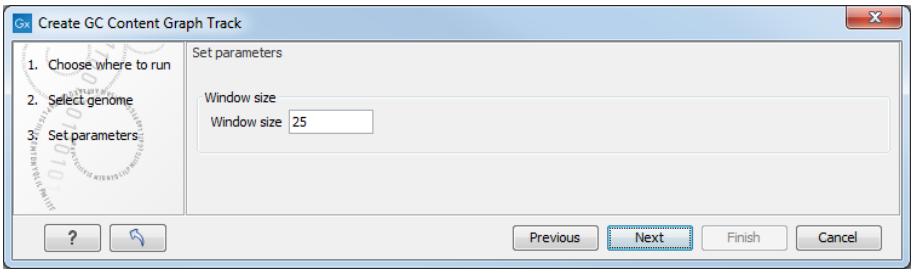


Figure 24.29: *Specify the window size. The window size is the region around each individual base that should be used to calculate the GC content in a given region.*

Click **Next**, choose to save your results, and click on the button labeled **Finish**. The output can be seen in figure 24.30. The output from "Create GC Content Graph" is a graph track. The graph track shows one value for each base with one graph being available for each chromosome. When zoomed out as shown in this figure, three different graphs with three different colors can be seen. The top graph with the darkest blue color represents the maximum observed GC content values in the specific region, the graph in the middle with the intermediate blue color shows the mean observed GC content values in the specific region, and the graph at the bottom with the light blue color shows the minimum observed GC content values in the specific region.

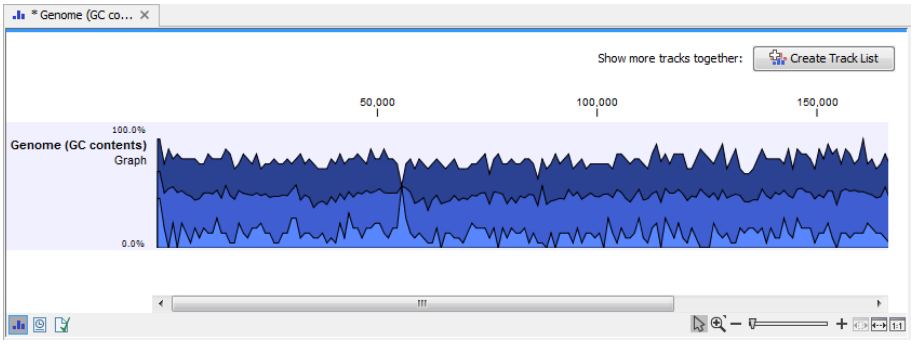


Figure 24.30: *The output from Create GC Content Graph is a graph track. The graph track shows one value for each base with one graph being available for each chromosome.*

When zooming all the way in to single nucleotide level only one graph can now be seen as you ar now no longer looking at large genomic regions. Instead, you can now use the tooltip by mousing over each individual base to look at the GC content for that particular base and the number of bases that you specified as the window size to be used. This is shown in figure 24.31 where the top part of the figure shows the graph track when zoomed all the way out and the bottom part of the figure shows a track list with the sequence track that was used as input together with the output graph track. The input and the output tracks were combined in one view as a track list (see section 24.3) by clicking on the button labeled **Create Track List** found in the upper right corner of the graph track in the top part of the figure (see the red arrow).

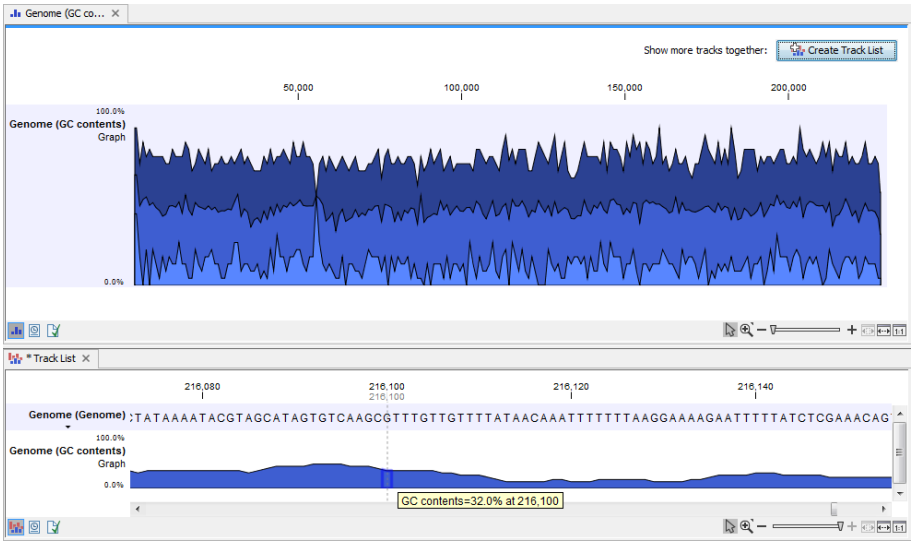


Figure 24.31: *The top part of the figure shows the graph track when zoomed all the way out. The bottom part of the window shows a graph track together with the input genomic sequence at single nucleotide resolution. By mousing over one nucleotide, you can see the GC content for this position. In our example we chose a window size of 25 nucleotides and the GC content that is shown for one nucleotide is the GC content for the central nucleotide and the 12 bases upstream and downstream of this nucleotide.*

24.9.2 Create Mapping Graph

The **Create Mapping Graph** tool can create a range of different graphs from a read mapping track. To run the tool go to the toolbox:

Toolbox | **Track Tools** ( ) | **Graphs** ( ) | **Create Mapping Graph** ( )

Select the read mapping as shown in figure 24.32 and click on the button labeled **Next**.

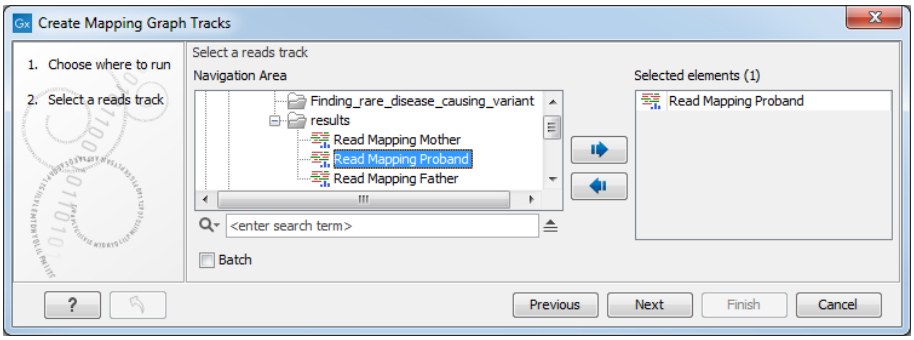


Figure 24.32: *Mapping graph tracks containing different types of information can be created.*

Select the graph track types to create, as shown in figure 24.33. One graph track is created for each type selected.

Each position in a graph track contains the values of the track type selected. These are:

- Read coverage. The number of reads contributing to the alignment at the position. A more detailed definition is in section 26.2.3.
- Non-specific read coverage The number of reads mapped at the position that would map

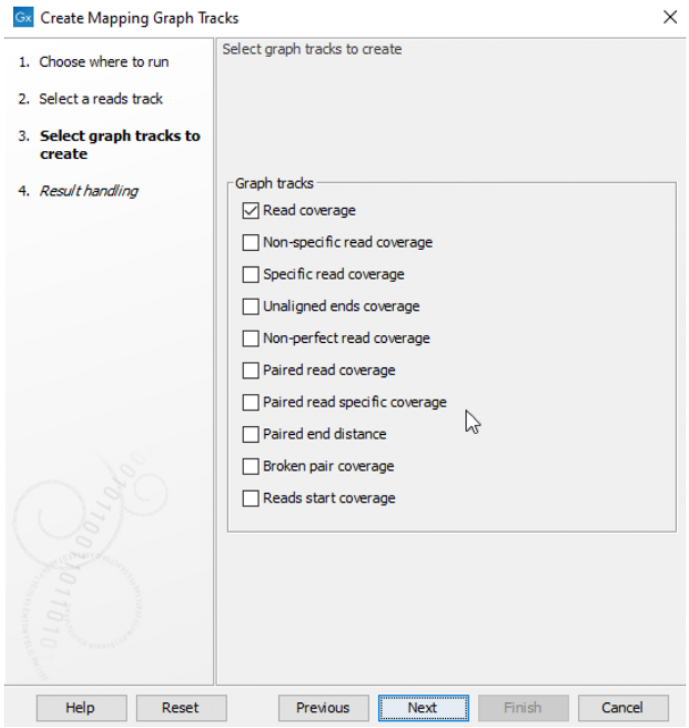


Figure 24.33: *Choose the types of graph tracks to be created.*

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 the 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.

Click **Next**, choose where to save the generated output(s) and click on the button labeled **Finish**.

An example of three different outputs is shown in figure 24.34.

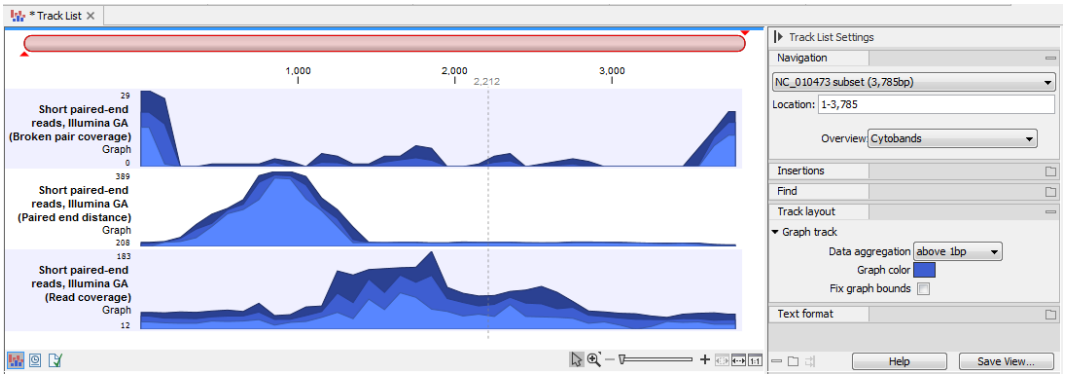


Figure 24.34: *Three types of graph tracks are shown.*

Note that the option "Fix graph bounds" found under **Track layout** in the **Side Panel** is useful to manually adjust the numbers on the y-axis.

When zoomed out, the graph tracks are composed of three curves showing the maximum, mean, and minimum value observed in a given region (see figure 24.37). When zoomed in all the way down to base resolution only one curve will be shown reflecting the exact observation at each individual position (figure 24.35).

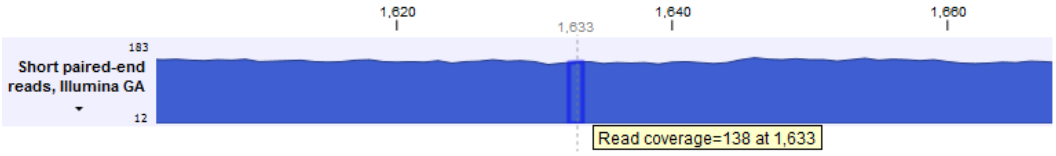


Figure 24.35: *A graph track zoomed in.*

24.9.3 Identify Graph Threshold Areas

The **Identify Graph Threshold Areas** tool uses graph tracks as input to identify graph regions that fall within certain limits or thresholds.

To run the tool go to the toolbox:

Toolbox | **Track Tools** ( ) | **Graphs** ( ) | **Identify Graph Threshold Areas** ( )

Both a lower and an upper threshold can be specified to create an annotation track for those regions of a graph track where the values are in the given range (see figure 24.36). The range chosen for the lower and upper thresholds will depend on the data (coverage, quality etc).

The **window-size** parameter specifies the width of the window around every position that is used to calculate an average value for that position and hence "smoothes" the graph track beforehand. A window size of 1 will simply use the value present at every individual position and determine if it is within the upper and lower threshold. In contrast, a window size of 100 checks if the average value derived from the surrounding 100 positions falls between the minimum and maximum threshold. Such larger windows help to prevent "jumps" in the graph track from fragmenting the output intervals or help to detect over-represented regions in the track that are only visible when looked at in the context of larger intervals and lower resolution.

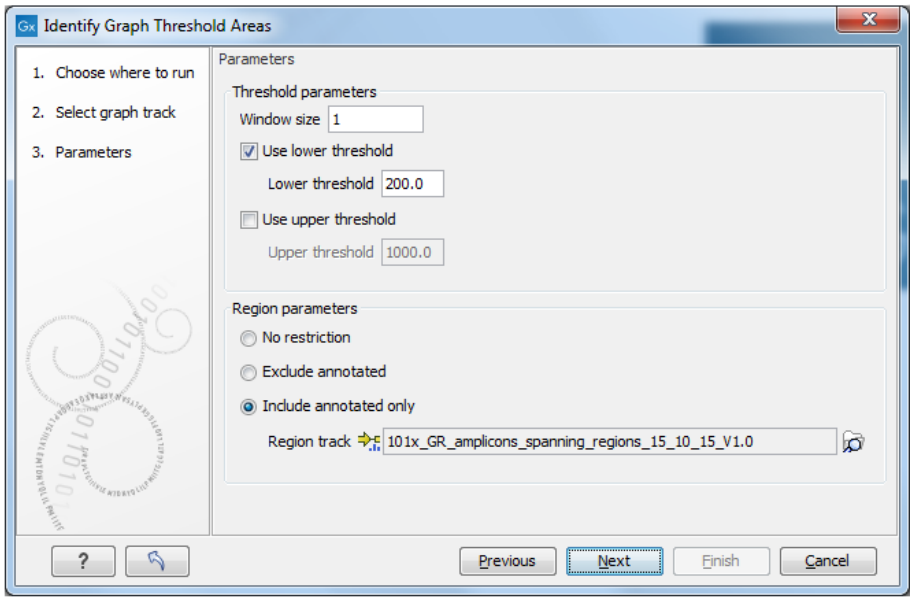


Figure 24.36: *Specification of lower and upper thresholds.*

It is also possible to restrict the tool to certain regions with specifying a region track.

An example output is shown in figure 24.37 where the coverage graph has some local minima. However, by using the averaging window, the tool is able to produce a single unbroken annotation covering the entire region. Of course larger window sizes result in regions that are broader and hence their boundaries are less likely to exactly coincide with the borders of visually recognizable borders of regions in the track.

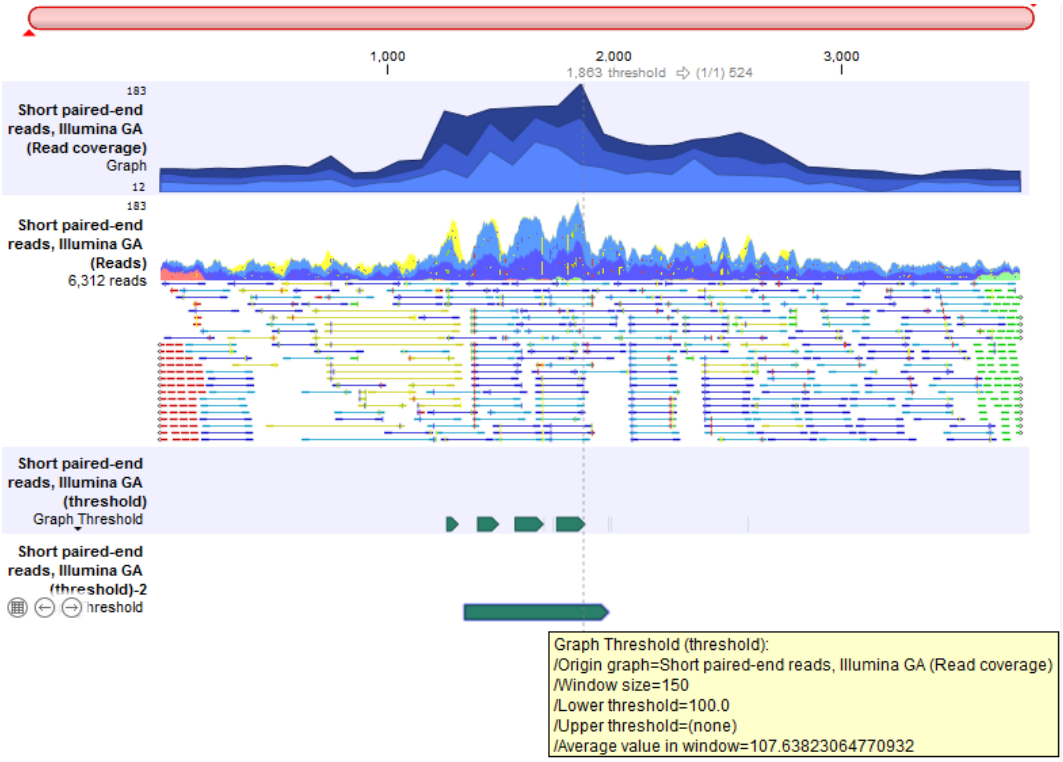


Figure 24.37: *Track list including a read coverage graph, a reads track, and two read graph threshold graph generated to annotate regions where the coverage was above 100. The top track graph threshold was generated with a window size of 1, while the one from below was generated with a window size of 150.*

Chapter 25

Prepare sequencing data

Contents

25.1 QC for Sequencing Reads 598
25.1.1 Per-sequence analysis 600
25.1.2 Per-base analysis 601
25.1.3 Over-representation analyses 601
25.2 Trim Reads 602
25.2.1 Quality trimming 603
25.2.2 Adapter trimming 604
25.2.3 Trim adapter list 606
25.2.4 Homopolymer trimming 613
25.2.5 Sequence filtering 614
25.2.6 Trim output 614
25.3 Demultiplex Reads 617
25.3.1 Demultiplexing single reads 618
25.3.2 Demultiplexing paired reads 620
25.3.3 Entering barcodes 621
25.3.4 Demultiplexing output options 622
25.3.5 An example using Illumina barcoded sequences 625

25.1 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 (http://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.

QC for Sequencing Reads is in the Toolbox at:

Toolbox | **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 9.3.

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 25.1. 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.

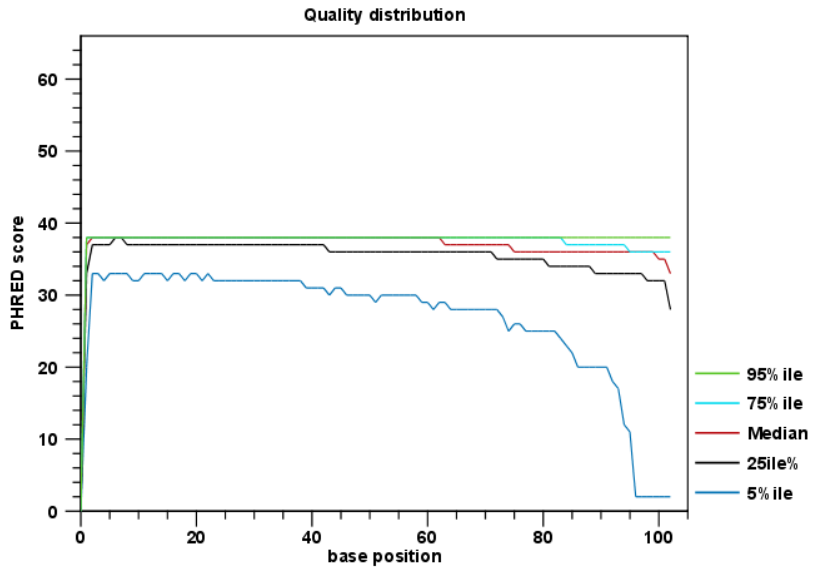


Figure 25.1: *An example of a plot from the graphical report, showing the quality values per base position.*

25.1.1 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.

25.1.2 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 25.1).

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.

25.1.3 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 25.1).

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 / 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.

25.2 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. Ambiguity trimming to trim off stretches of Ns for example
3. Adapter trimming (automatic, or also with a Trim Adapter List, see section 25.2.2)
4. Homopolymer trimming
5. Base trim to remove a specified number of bases at either 3' or 5' end of the reads
6. Length trimming 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:

Toolbox | **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 a list of trimmed sequences for each input sequence list.

When the sequences are selected, click **Next**.

25.2.1 Quality trimming

This opens the dialog displayed in figure 25.2 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: $Q = -10log10(P)$, 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: $p_{error} = 10^{\frac{Q}{-10}}$. (This now means that low values are high quality bases.)

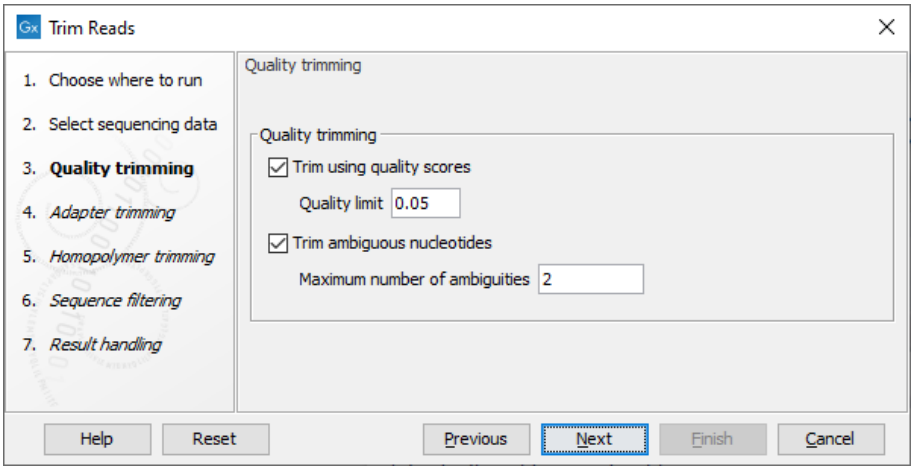


Figure 25.2: *Specifying quality trimming.*

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 `http://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).

25.2.2 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 25.3).

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 read. Leaving this option enabled is always recommended: the trimming performed automatically

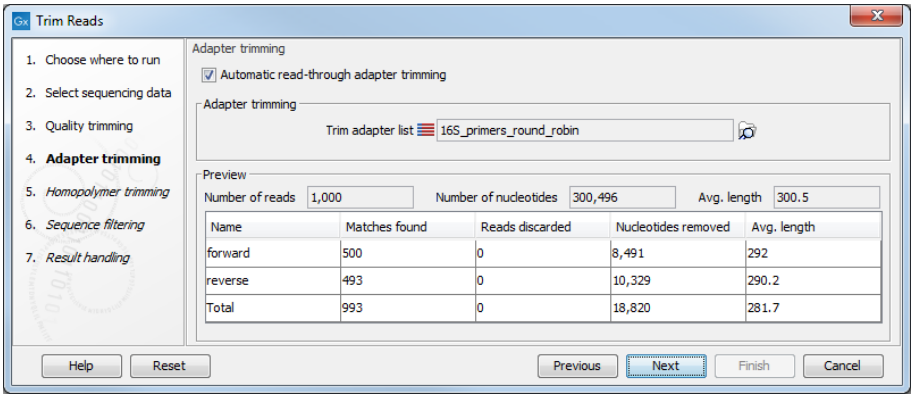


Figure 25.3: *Trimming your sequencing data for adapter sequences.*

can detect read-through of even a single nucleotide, which is not the case when trimming using a trim adapter list. The detected adapters for the first and second read can be found in the Trim Reads report.

There are however 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 25.2.3 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). 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.

25.2.3 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 25.4), 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 25.5), 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.

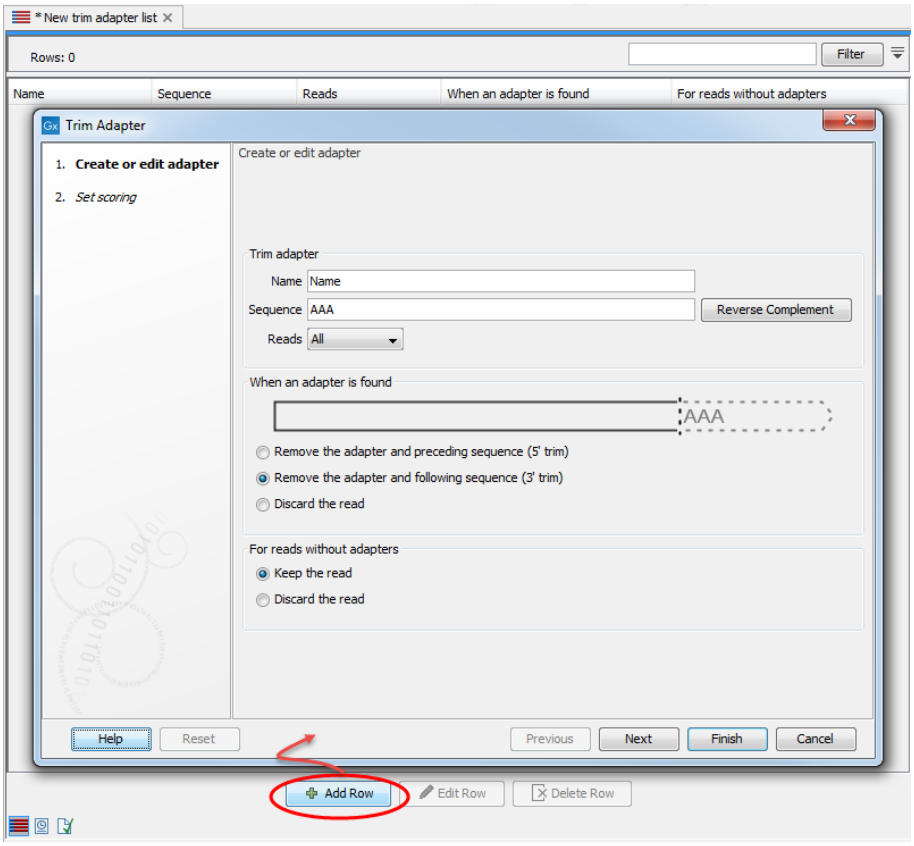


Figure 25.4: *Add an adapter to the Trim Adapter List by clicking on the button labeled "Add Row" found at the bottom of the New Trim Adapter view.*

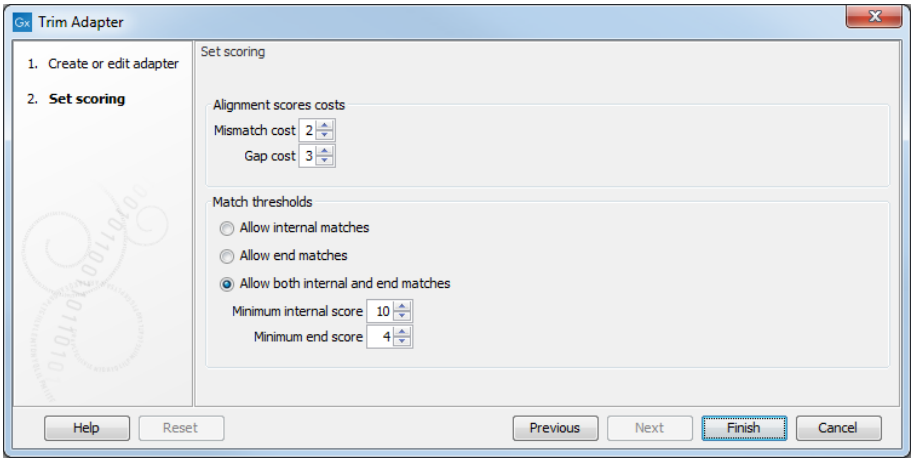


Figure 25.5: *Set the scoring used to define what will be considered as adapter.*

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 25.6). These examples are all internal matches where the alignment of the adapter falls within the read.

```
    CGTATCAATCGATTACGCTATGAATG
a)      ||||||| ||||            11 matches - 2 mismatches = 7
       TTCAATCGGTTAC

    CGTATCAATCGATTCGCTATGAATG
       |||||||||| ||||          14 matches - 1 gap = 11
b)     ATCAATCGAT-CGCT

    CGTATCAATCGATTACGCTATGAATG
c)      |||||||                 7 matches - 3 mismatches = 1
       TTCAATCGGG
```

Figure 25.6: *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**. An end match happens when the alignment of the adapter starts at the end of the sequence that is being trimmed. This can be 5' or 3' depending on the option chosen in the first dialog. Note that for 3' trim, we internally reverse-complement the read and look for a match at the 5' end of the reverse complemented sequence. So 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.

If a match can be treated as either an end match or an internal match, the workbench will treat it as an end match.

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 25.7 shows a few examples with an adapter match at the end.

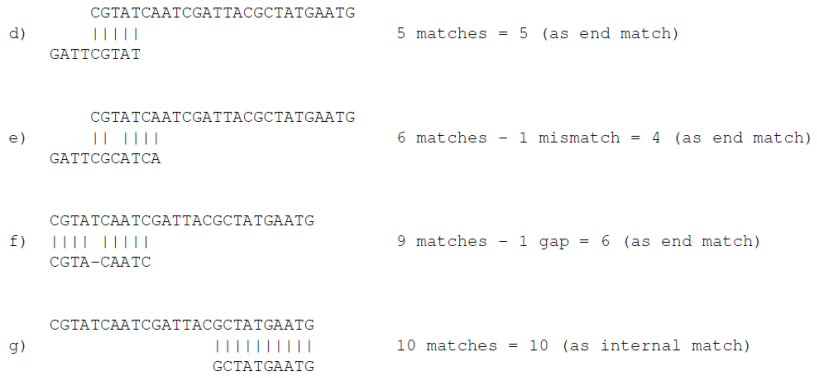


Figure 25.7: *Four examples showing a sequencing read (top) and an adapter (bottom). The examples are artificial.*

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 a 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 25.8), 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

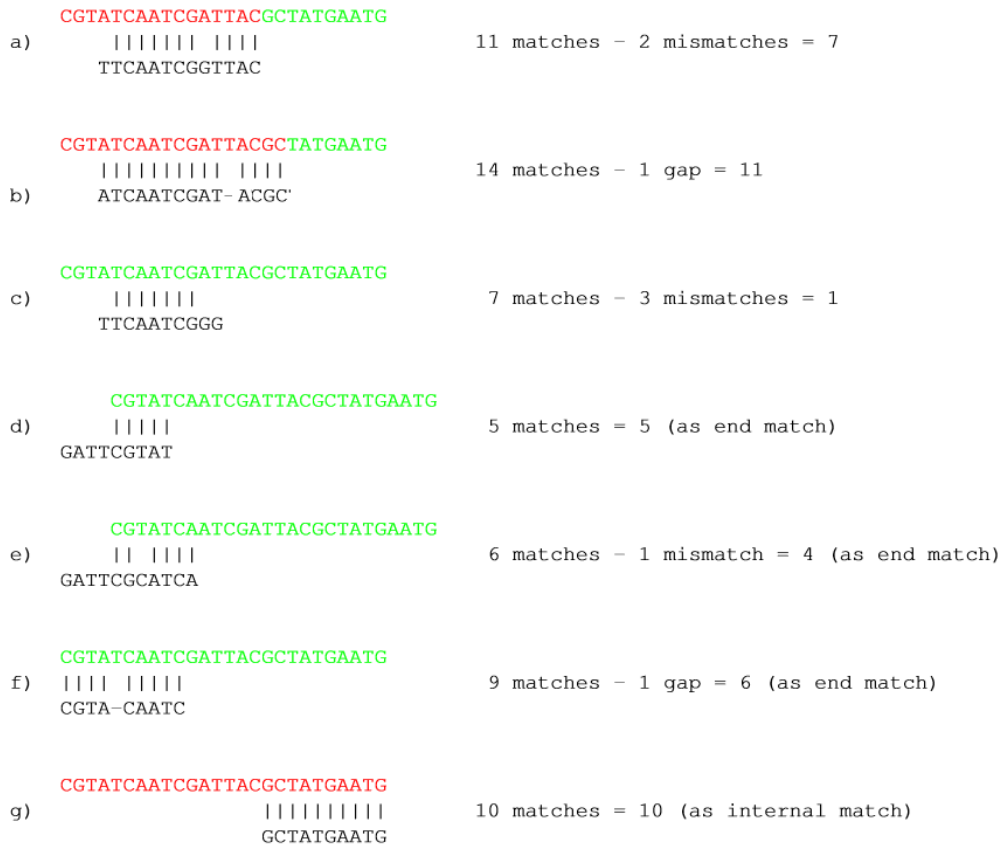


Figure 25.8: *The results of trimming with internal matches only. Red is the part that is removed and green is the retained part. Note that the read at the bottom is completely discarded.*

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 25.9.

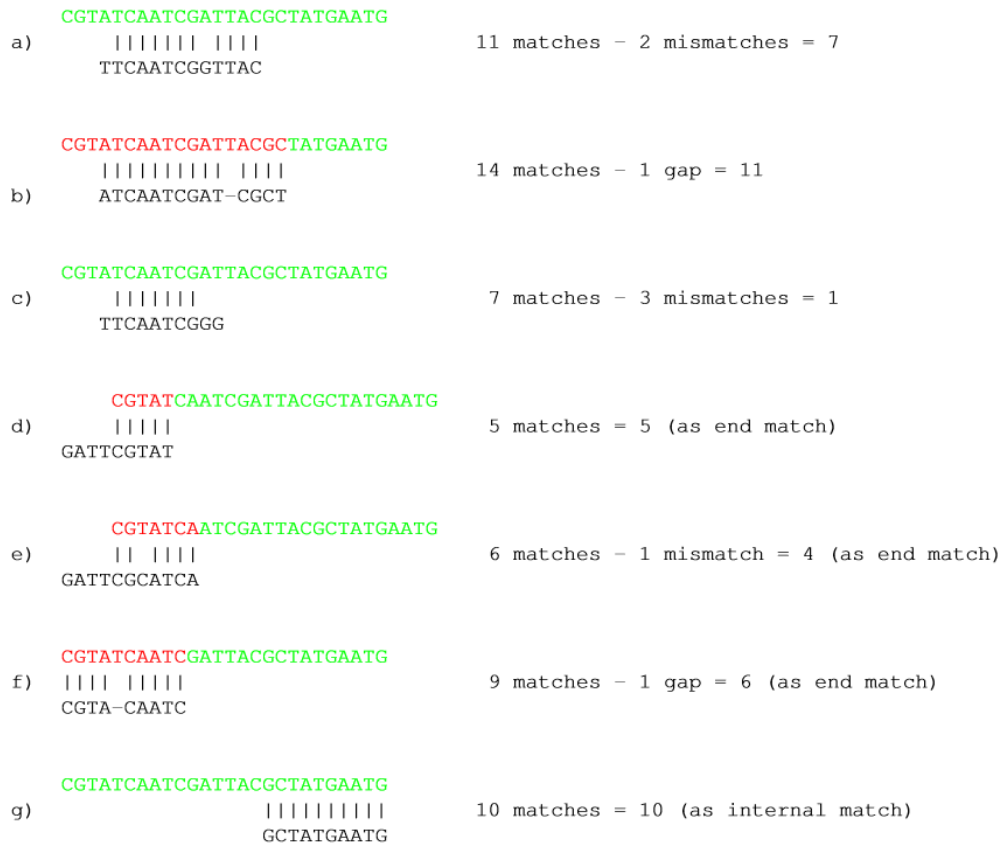


Figure 25.9: *The results of trimming with both internal and end matches. Red is the part that is removed and green is the retained part.*

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 the 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 25.10).

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 CCGTCTTCTGCTTTAAAAAAAAAAAACAAACAAGACAACACCAAAATA
Read-through sequence (High confidence)	CT

Statistics for read 2

Read 2 trimmed	65
Read 2 trimmed (percent)	0.05%
Read-through sequence	CTGTCTCTTATACACATCTGACGCTGCCGACGCCTAGTCGAGTGTAGATC CCGGTGGTCCCCGGATCATTCAAAACAAAAAACCCCATGCAACCGCGACC
Read-through sequence (High confidence)	C

Figure 25.10: *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-though 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 25.11:

```
"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","Micmatch: 2, Gapcost: 3, Cutoff: 10, Cutoff at end: 4"
```

Figure 25.11: *The expected import format for Adapter Lists.*

25.2.4 Homopolymer trimming

Configuration for the homopolymer trimming step is shown in figure 25.12.

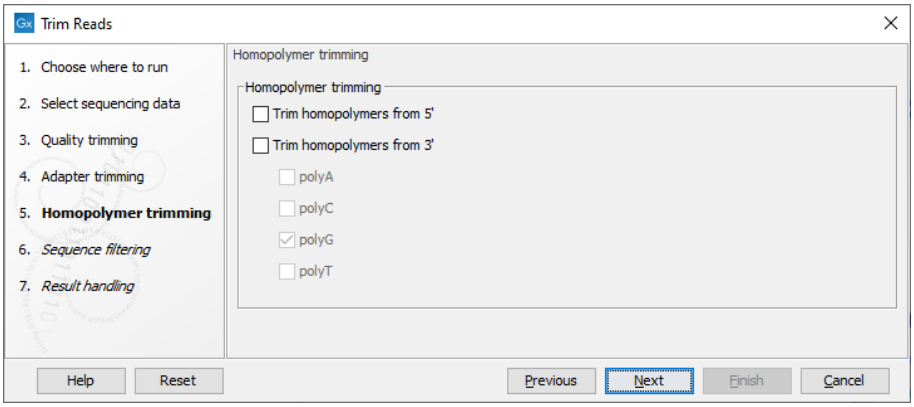


Figure 25.12: *Trimming homopolymer.*

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 at least 9 bases are Gs, then this stretch of 10 bases will later be trimmed away. The window then slides by one position, to cover 9 of the original bases and 1 additional base. If at least 9 of these 10 bases are Gs, then this stretch will be marked for trimming. 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 `CCCCCCTCCCCCCATATATATATATATCCCCCCTCCCCC` for polyC at the start and end of the read would result in `ATATATATATATAT`.

- Trimming the sequence `CCCCCCTCCCCCCATATATATATATATTTTTTTTTTGTTTTTT` 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 `TATTTTTTT` contains nine Ts, and thus all 10 bases in this stretch are removed.
- Trimming the sequence `AAAAAAAAAATATTTTTTTTTTGTTTTTT` for polyA and polyT at the start and end of the read would result in the whole sequence being trimmed away.

25.2.5 Sequence filtering

Clicking **Next** will allow you to specify length trimming as shown in figure 25.13.

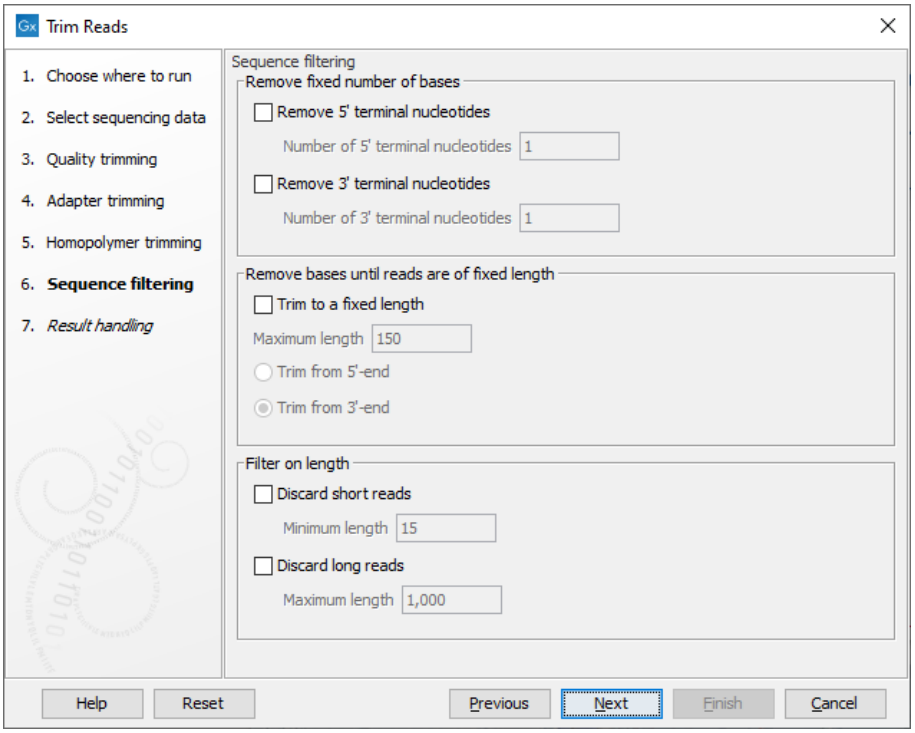


Figure 25.13: *Trimming on length.*

At the top you can choose to **Trim bases** by specifying a number of bases to be removed from either the 3' or the 5' end of the reads.

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.

Finally 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).

25.2.6 Trim output

Clicking **Next** will allow you to specify the output of the trimming as shown in figure 25.14.

In most case, independently of what option are selected in this dialog, a list of trimmed reads will be generated:

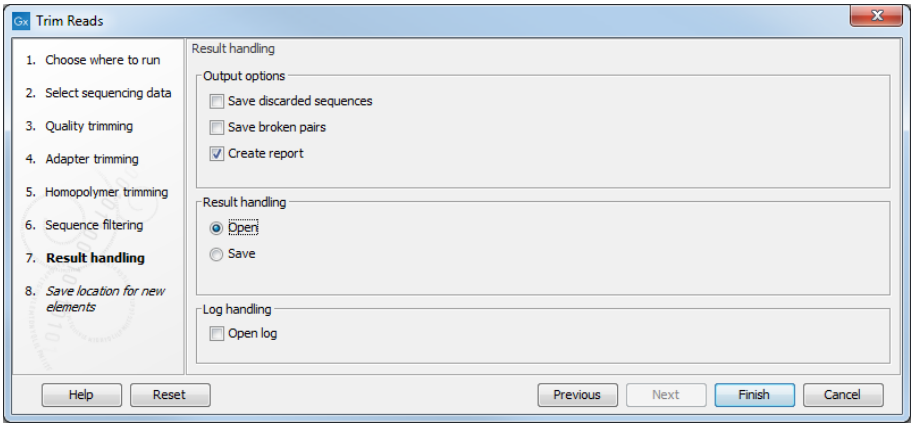


Figure 25.14: *Specifying the trim output.*

- 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 25.15. 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.
    - **Number of reads after trim.** The 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.

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

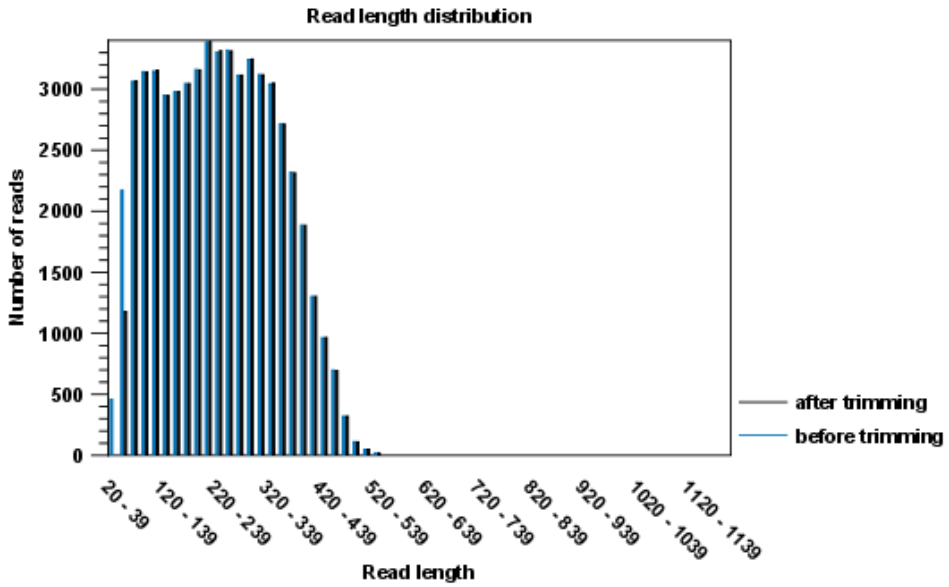


Figure 25.15: *A report with statistics on the trim results. Note that the Average length after trimming (232,8bp) is bigger than before trimming (228bp) because 2.000 very short reads were discarded in the trimming process.*

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%.

25.3 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 and after the sequence of interest. This principle is shown in figure 25.16.

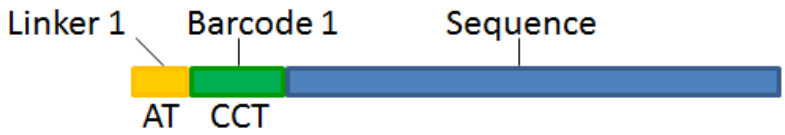


Figure 25.16: *Tagging the target sequence, which in this case is single reads from one sample.*

The sample-specific tag, also called the barcode or the index, can then be used to distinguish between the different samples when analyzing the sequencing data.

Post-processing of the sequencing data is required to separate the reads into their corresponding samples. The **Demultiplex Reads** tool does this, based on the sequence barcodes. Using this tool, sequences are associated with a particular sample when they contain an exact match to a particular barcode. (An option to allow one mismatch is available.) Sequences that do not match any barcode sequence are classified as not grouped and are put into a sequence list with the name "Not grouped".

When Demultiplex Reads is used within a workflow, the sets of reads to be analyzed together as a unit can be organized based on the barcode table. See section 11.5.3 for further details.

An example of demultiplexing reads using Illumina-barcoded sequences is provided in section 25.3.5.

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 it is available to you.

25.3.1 Demultiplexing single reads

To demultiplex your data, please go to:

Toolbox | **Prepare Sequencing Data** ( ) | **Demultiplex Reads ( )**

This opens a dialog where you can specify the sequences to process (figure 25.17).

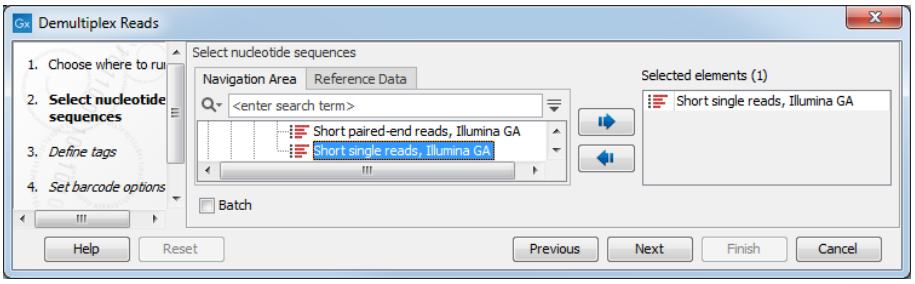


Figure 25.17: *Specify the sequences to demultiplex.*

When you click on the button labeled **Next**, you can then specify the details of how the demultiplexing should be performed. At the bottom of the dialog, there are three buttons, which are used to **Add**, **Edit**, and **Delete** the elements that describe how the barcode is embedded in the sequences.

First, click **Add** to define the first element. This will bring up the dialog shown in 25.18.

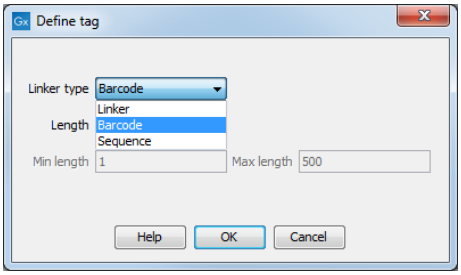


Figure 25.18: *Defining an element of the barcode system.*

At the top of the dialog, you can choose the type of element 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 25.16, the linker is two nucleotides long. For this element, you simply define its length - nothing else.
- **Barcode**. The barcode (also known as index) is the stretch of nucleotides used to group the sequences. In this dialog, you simply need to specify the length of the barcode. The valid sequences for your barcodes must be provided at a later wizard step.
- **Sequence**. This element 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 elements 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 of

course edit and delete elements by selecting them and clicking the buttons below. In the example shown in figure 25.16, the dialog should include a linker, a barcode, and a sequence as shown in figure 25.19.

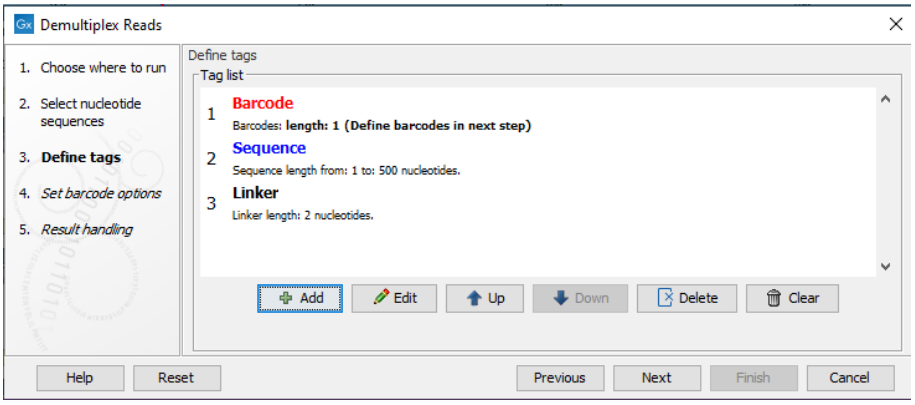


Figure 25.19: *Processing the tags as shown in the example of figure 25.16.*

Click **Next** to set the barcode options (figure 25.20). At the top, you can choose to search on both strands for the barcodes; this is needed for some 454 protocols where the MID is located at either end of the read. 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: If a sequence is one mismatch away from two barcodes, it will not be assigned to any of them.

In the table below, the **Preview** column will show a preview of the results by running through the first 10,000 reads.

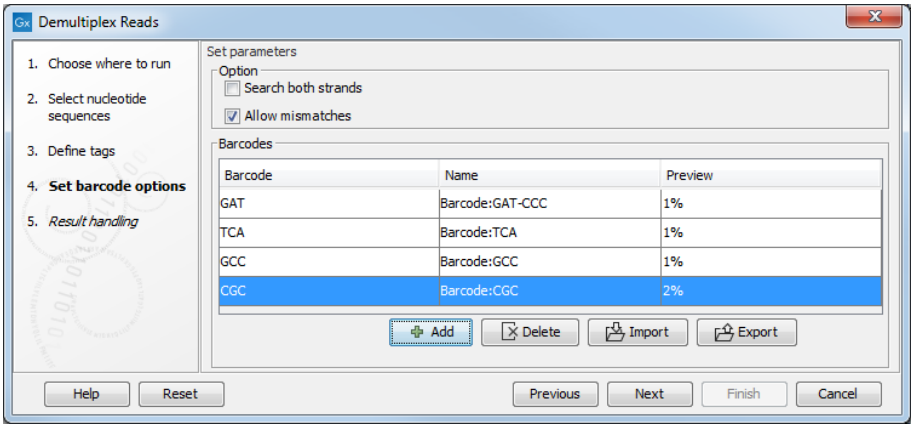


Figure 25.20: *A preview of the results.*

If you would like to change the name of the sequence(s), this can be done at this step by double-clicking on the specific name that you would like to change. This is shown in figure 25.21.

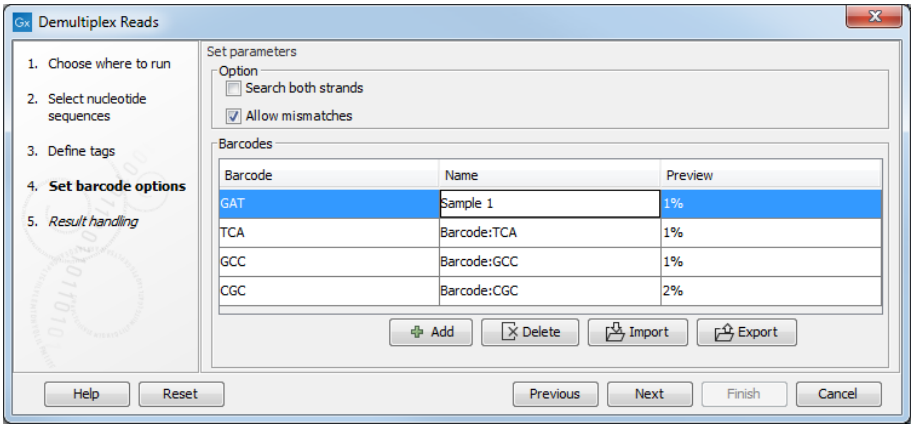


Figure 25.21: *The name of the sequence can be renamed by double-clicking on the existing name.*

25.3.2 Demultiplexing paired reads

If you have paired data the procedure is exactly the same, except for one thing: the dialog shown in figure 25.19 will be displayed twice - one for each part of the pair.

Figure 25.22 shows an example that could illustrate paired end reads. In this example we have paired end reads from two different samples mixed together.

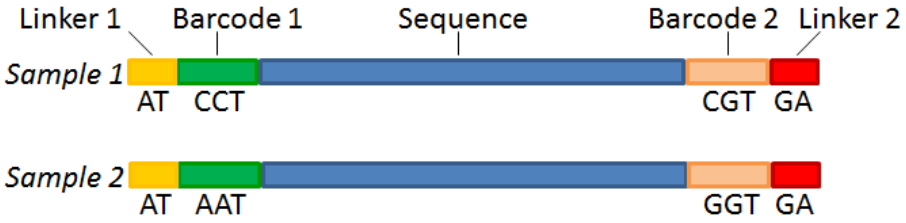


Figure 25.22: *Paired end reads with linkers and barcodes. Two different samples are mixed together in this example.*

In a situation where the paired reads are expected to be barcoded in the same way (see example below), you would set the parameters for read1 (wizard step 3) and read2 (wizard step 4) to be the same.

Read1 : –Linker–Barcode1–Sequence

Read2 : –Linker–Barcode1–Sequence

However, if read2 of the pair is not expected to be the same as read1 in the pair, it is necessary to adjust these settings accordingly. For example, it is possible that read2 does not contain any barcode sequence at all. In this case, you would simply set the sequence parameter for the mate and exclude the barcode and linker parameters. If the two reads in the read pair have different barcodes, the situation would look like this:

Read1 : –Linker–Barcode1–Sequence

Read2 : –Linker–Barcode2–Sequence

To demultiplex paired end reads the first two steps are similar to the demultiplexing of single

reads. Select the paired end read sequences that should be demultiplexed (figure 25.23).

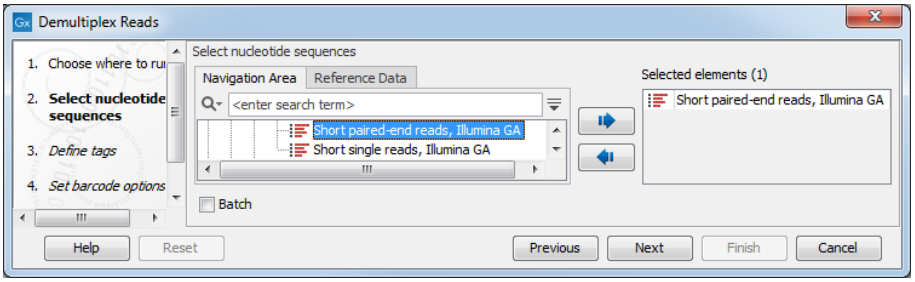


Figure 25.23: *Specifying the paired end reads to demultiplex.*

Click **Next** to define the tags and sequence for the forward reads (figure 25.24).

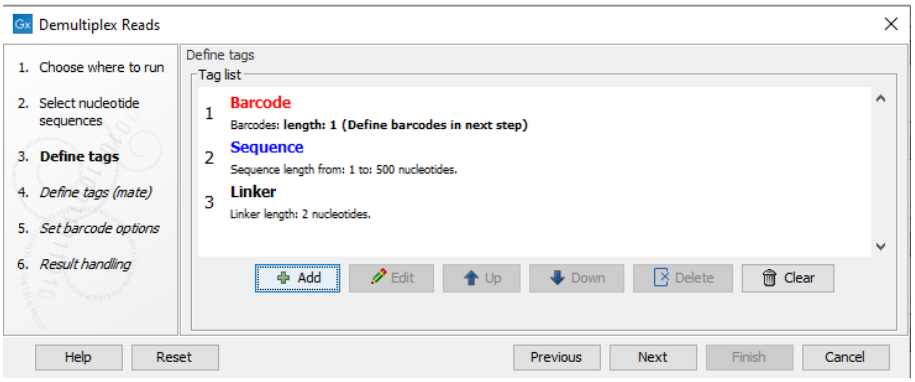


Figure 25.24: *Specifying the setup of the forward read (tags and sequence).*

Click **Next** to define the tags and sequence for the reverse reads and set barcode options as explained for single reads.

25.3.3 Entering barcodes

Barcodes can be entered manually or imported from a properly formatted CSV or Excel file.

Manually enter barcodes

Barcodes can be entered manually when launching the Demultiplex Reads tool by clicking the **Add** ( ) button in the "Set barcode options" wizard step. In the example shown in figure 25.22 the barcodes should be defined as shown in figure 25.25.

You can edit the barcodes and the names by clicking the cells in the table. The barcode name is used when naming the results.

Import barcodes from CSV or Excel format files

If running the Demultiplex Reads tool, click on the **Import** ( ) button in the "Set barcode options" wizard step.

When importing barcodes this way, the input format consists of two columns: the first contains the barcode sequence, the second contains the name of the barcode. An acceptable CSV format file would look like:

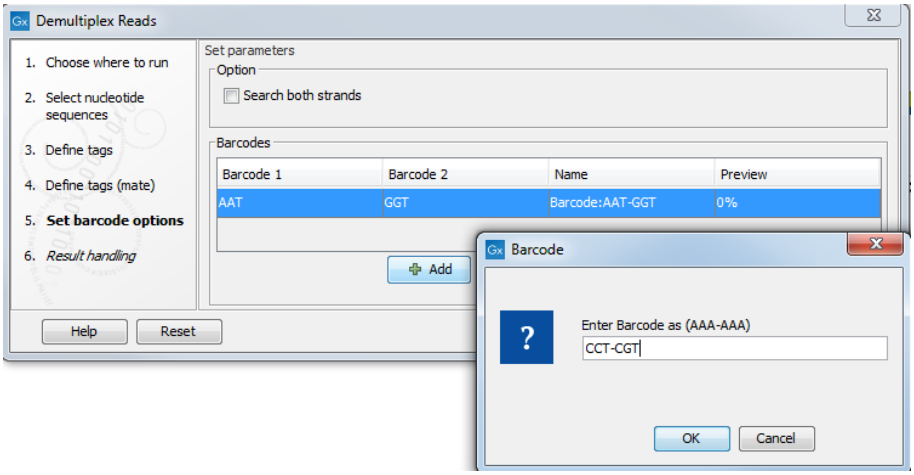


Figure 25.25: *The barcodes for the set of paired end reads for sample 1 have already been defined and the barcodes for sample 2 is being entered in the format AAA-AAA, which corresponds to Barcode1-Barcode2 for sample 2 in the example shown in figure 25.22.*

```
 "AAAAAA","Sample1"
"GGGGGG","Sample2"
 "CCCCCC","Sample3"
```

For Demultiplex Reads within a workflow, barcodes can only be entered using a CSV or Excel format file, which can be selected when launching the workflow. The format of the file can be as simple as two columns, the first with the barcode sequences, the second with the names of the barcodes, as described above. However, in a workflow context, the file can optionally contain a header, and more columns. Columns not defining the barcode and sample name are interpreted as metadata and can be used for defining batch units. See section 11.5.3 for further details.

25.3.4 Demultiplexing output options

In the last wizard step, where you can specify the output options. If you choose to keep the default settings, three different types of output will be generated; 1) The demultiplexed reads, one output for each specified barcode (the file name starts with the sample name, followed by the specified barcode name), 2) The discarded reads that did not have a barcode (specified by the suffix "Not grouped"), and 3) a "Demultiplex Reads report", which shows the fraction of reads with and without a barcode (see figure 25.26).

There is also an option to create subfolders for each sequence list. This can be handy when the results need to be processed in batch mode (see section 9.3).

A new sequence list will be generated for each barcode 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.

An example of the demultiplexing summary report is shown in figure 25.27.

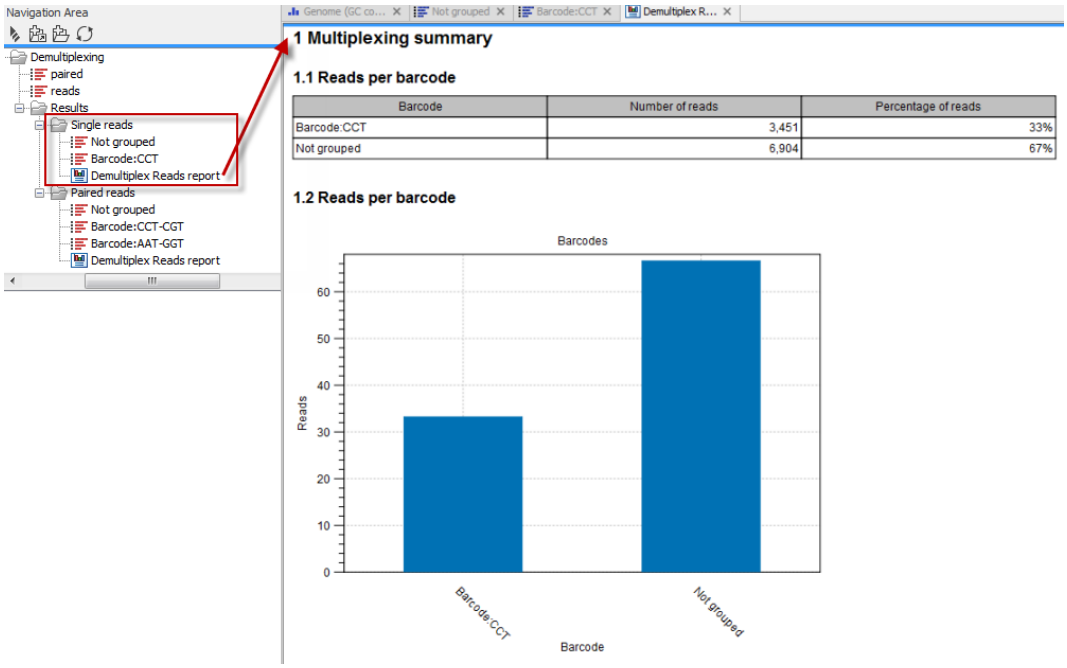


Figure 25.26: *Three different outputs are generated when analyzing single reads with only one sample using the default output settings. If several samples had been mixed together there would be a sequence list for each sample (each specified barcode). The Demultiplex Reads report is shown in the right-hand side of the figure.*

1 Demultiplexing summary

1.1 Reads per barcode

Barcode	Number of reads	Percentage of reads
Barcode:GGT	1,745,043	26%
Barcode:CGT	1,305,703	20%
Barcode:AAT	1,850,050	28%
Barcode:CCT	1,251,849	19%
Not grouped	445,560	7%

1.2 Reads per barcode

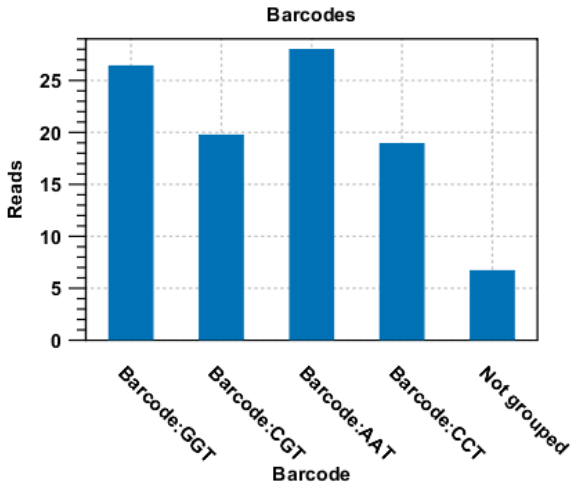


Figure 25.27: *An example of a report showing the number of reads in each group. In this example four different barcodes were used to separate four different samples.*

25.3.5 An example using Illumina barcoded sequences

The data set in this example can be found at the Short Read Archive at NCBI. Use the Search for Reads in SRA... tool to search for SRX014012. Select the SRR03730 item and click **Download Reads and Metadata**. Save the sequence list in the Navigation Area, and use it with the Demultiplex Reads tool.

The barcoding was done using the following tags at the beginning of each read: CCT, AAT, GGT, CGT (see supplementary material of Cronn et al., 2008). The settings in the dialog should thus be as shown in figure 25.28.

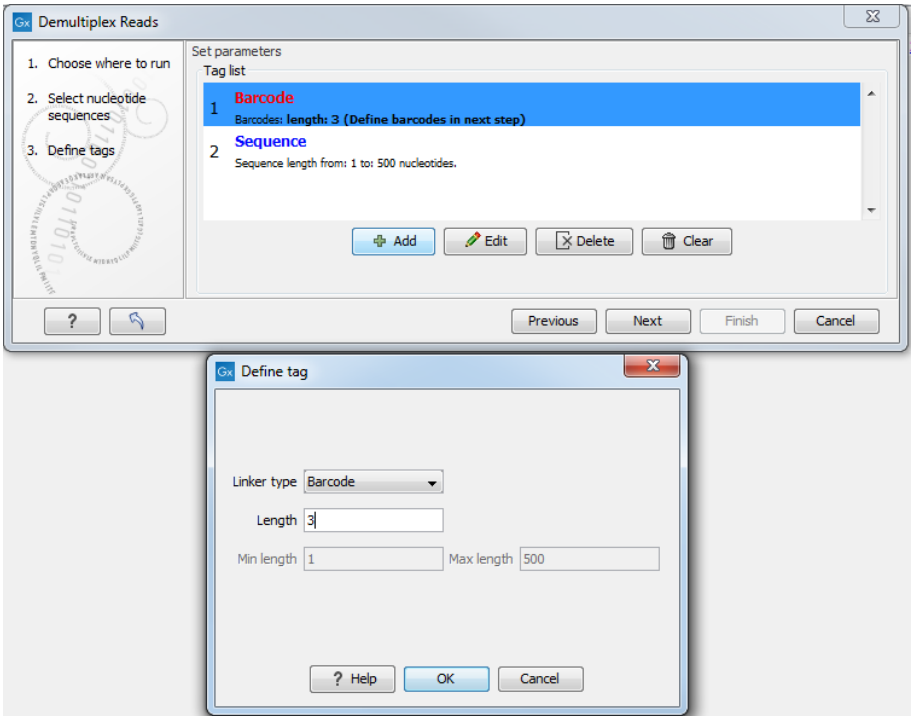


Figure 25.28: *Setting the barcode length at three.*

Click next to the "Set barcode options" dialog and use the **Add** button to specify the bar codes as shown in figure 25.29.

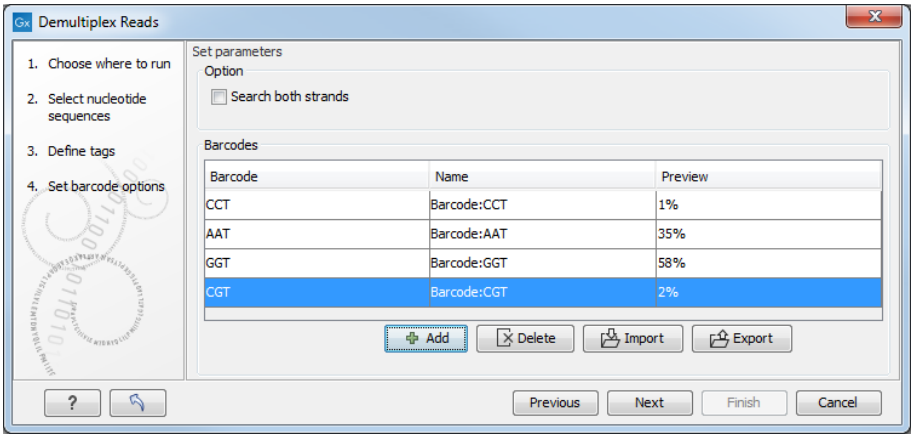


Figure 25.29: *A preview of the result.*

With this data set we got the four groups as expected (shown in figure 25.30). The **Not grouped**

list contains 445,560 reads that will have to be discarded since they do not have any of the barcodes.

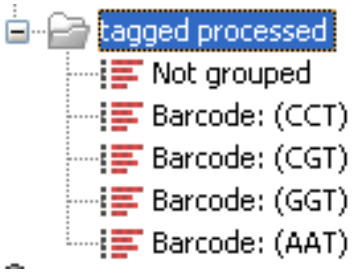


Figure 25.30: *The result is one sequence list per barcode and a list with the remainders.*

Chapter 26

Quality control for resequencing analysis

Contents

26.1 QC for Targeted Sequencing 627
26.1.1 Coverage summary report 630
26.1.2 Per-region statistics 634
26.1.3 Coverage table 635
26.1.4 Coverage graph 636
26.2 QC for Read Mapping 637
26.2.1 References 638
26.2.2 Mapped read statistics 640
26.2.3 Statistics table for each mapping 642
26.3 Whole Genome Coverage Analysis 643
26.4 Combine Reports 645
26.4.1 Combine Reports output 646
26.4.2 Report types supported 647
26.5 Create Sample Report 649
26.5.1 Create Sample Report output 650

26.1 QC for Targeted Sequencing

This tool is designed to report the performance (enrichment and specificity) of a targeted re-sequencing experiment. Targeted re-sequencing is due to its low costs, very popular and several companies provide platforms and protocols (learn more at http://en.wikipedia.org/wiki/Exome_sequencing#Target-enrichment_strategies). Array-based approaches are offered by Agilent (SureSelect) and Roche Nimblegen. Furthermore, amplicon sequencing with PCR primers is offered by RainDance, Fluidigm and others.

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.

Note! This tool is for re-sequencing data only; if you have RNA-seq data, please see section 30.

To create the target regions statistics:

Toolbox | **Quality Control** ( ) | **QC for Targeted Sequencing ( )**

This opens a wizard where you can select mapping results ( )/ ( )/ ( ) as seen in figure 26.1.

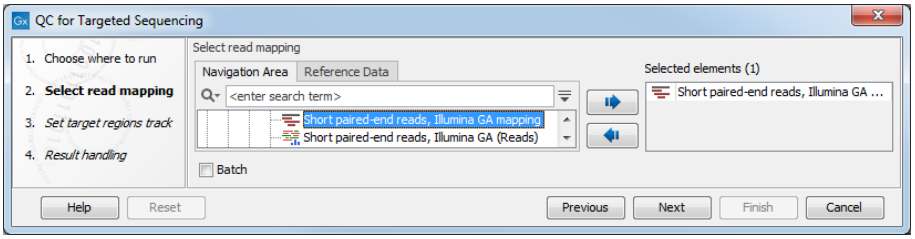


Figure 26.1: *Select a read mapping.*

Clicking **Next** will take you to the wizard shown in figure 26.2.

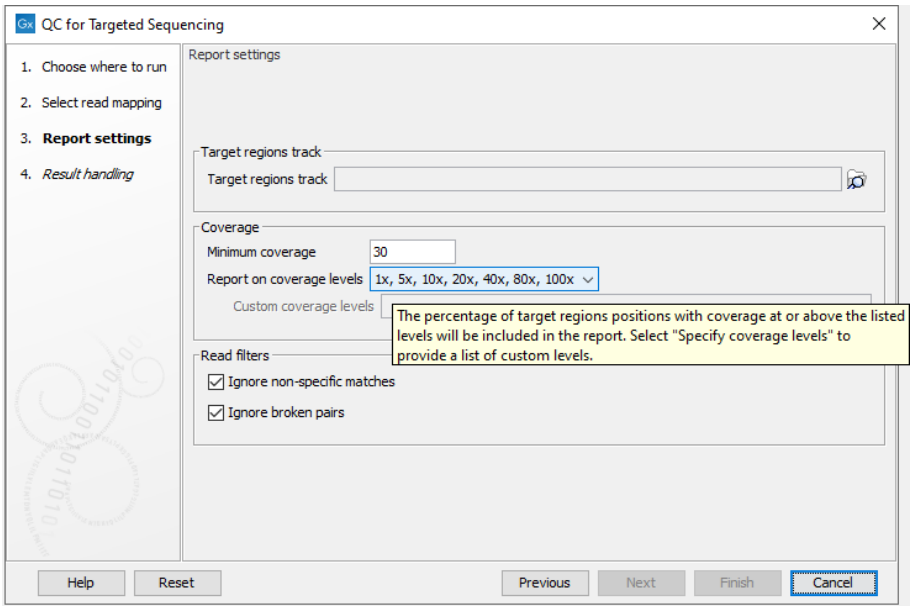


Figure 26.2: *Specifying the track of target regions.*

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 6.2) or convert (see section 24.7) from annotations on a reference genome that is already stored in the **Navigation Area**.

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 26.3. By selecting Specify coverage levels you get the option to add a list of comma-separated custom coverage levels. As shown in figure 26.3 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.

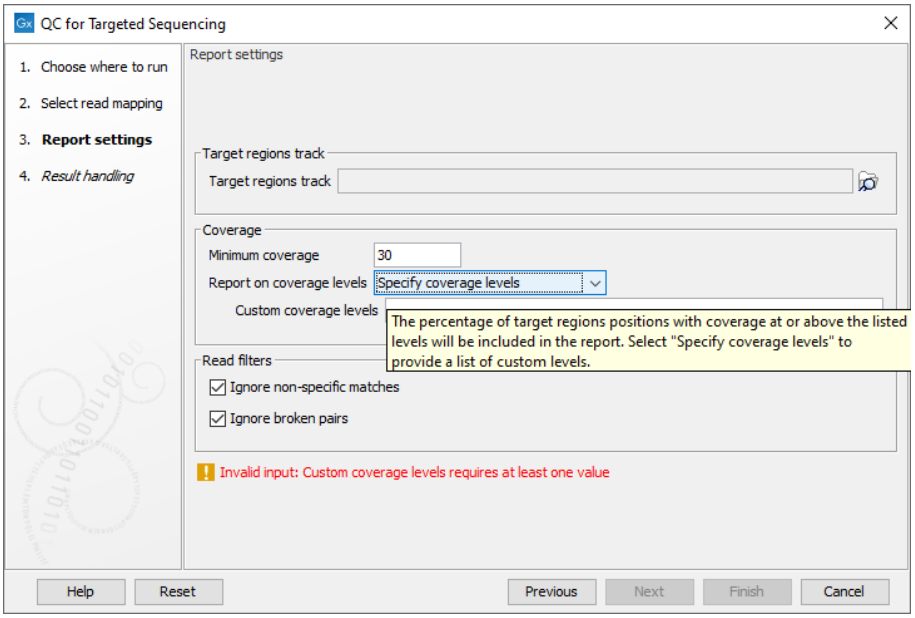


Figure 26.3: *Selecting Specify coverage levels from the drop-down list will allow you to add your own custom coverage levels in the text field by typing in the desired coverage levels. Numbers should be comma-separated.*

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 26.4.

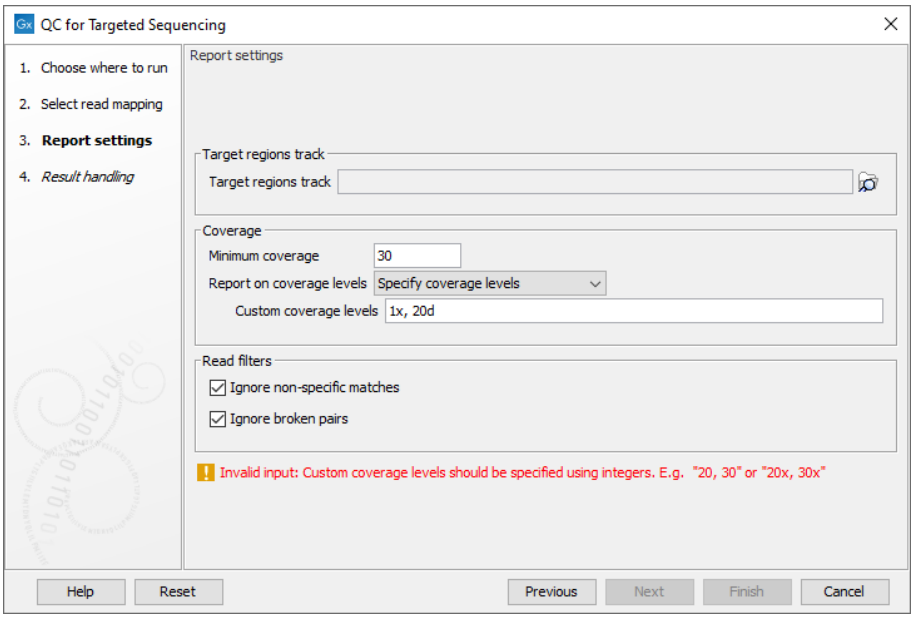


Figure 26.4: *When adding a list of custom coverage levels, numbers should be comma-separated and provided in the format 20, 30, 40 or 20x, 30x, 40x.*

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 pairs will be ignored.

Click **Next** to specify the type of output you want (see figure 26.5).

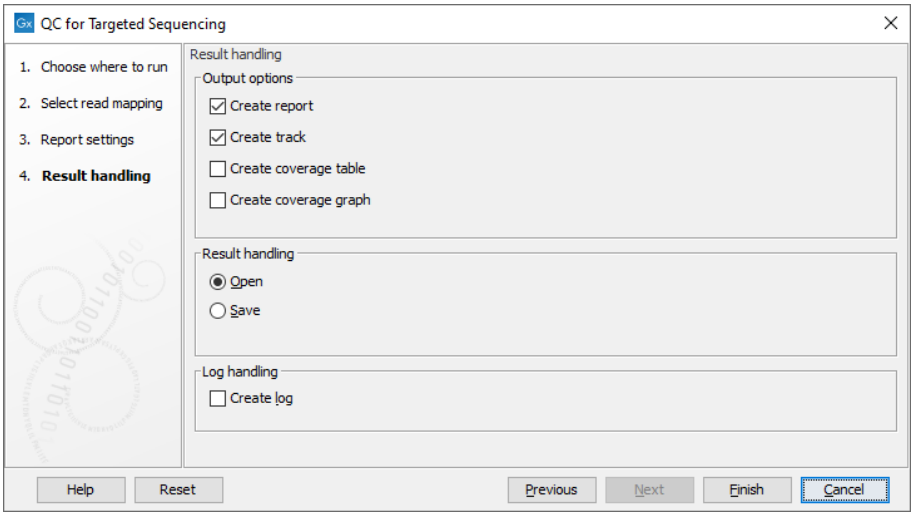


Figure 26.5: *Specifying how the result should be reported.*

There are three options:

- The report gives an overview of the whole data set as explained in section 26.1.1.
- The track gives information on coverage for each target region as described in section 26.1.2.
- The coverage table outputs coverage for each position in all the targets as described in section 26.1.3.
- 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.

Click **Finish** to create the selected reports.

26.1.1 Coverage summary report

An example of a coverage report is shown in figure 26.6.

This figure shows only the top of the report. The full content is explained below:

- **Target regions**
  - **Summary** This table shows overall coverage information.
    - **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).

1 Target regions

1.1 Summary

Number target regions	5
Total length of targeted regions	2,025
Average coverage	40.4
Median coverage	38.0
Number of target regions with coverage below 30	5
Total length of target regions containing positions with coverage below 30	2,025
Total length of target region positions with a coverage below 30	562

There are overlapping targets in the target region track. A read/base may contribute to the coverage of multiple targets.

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	0	0.00
≥90% of the targeted region has coverage at least 30	1	20.00
≥80% of the targeted region has coverage at least 30	2	40.00
≥70% of the targeted region has coverage at least 30	3	60.00
≥60% of the targeted region has coverage at least 30	3	60.00
≥50% of the targeted region has coverage at least 30	3	60.00
≥40% of the targeted region has coverage at least 30	3	60.00
≥30% of the targeted region has coverage at least 30	3	60.00
≥20% of the targeted region has coverage at least 30	4	80.00
≥10% of the targeted region has coverage at least 30	5	100.00
≥0% of the targeted region has coverage at least 30	5	100.00

Figure 26.6: *The report with overviews of mapped reads.*

  * **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 26.11). 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 +- 3SDs from the median.
- **Minimum coverage of target regions** 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 26.7 this means that 26.58 % of the positions on the target are covered by at least 40 bases.

1.5 Minimum coverage of target regions

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 26.7: *Minimum coverage of target regions of the report.*

- **Targeted regions 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 26.8).

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 26.8: *Targeted regions overview of the report: mapped bases.*

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 26.9 and figure 26.10).

  - **Reference** The name of the reference sequence.
  - **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 ingored** 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 ingored** 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 ingored** 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.

26.1.2 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 26.9:

- **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 26.9 and figure 26.10).
- **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 26.9 and figure 26.10).
- **%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.

In the shown figure, the coverage table is shown in split view with a track list. The coverage track has been opened from the track list by clicking once on the name found in the left side in the track list. When opening a read mapping in split view from a track list, the two views are linked, this means that when you click on an entry in the table, this position will be brought into focus in the track list.

For information about how to create a track list, please see section 24.3.

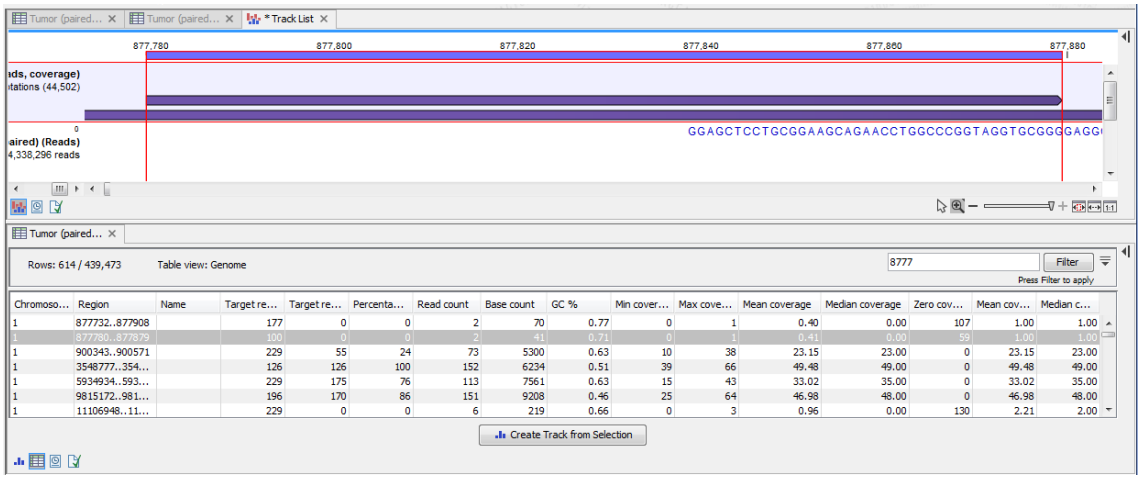


Figure 26.9: *A track list containing the target region coverage track and reads track. The target region coverage track has been opened from the track list and is shown in table view. Detailed information on each region is displayed. Only one paired read maps to the region selected.*

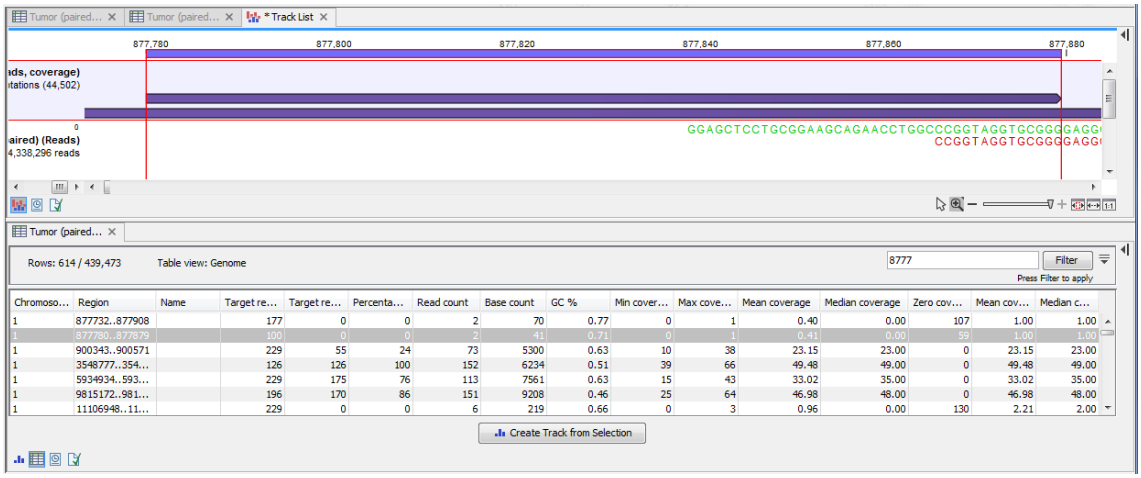


Figure 26.10: *The same data as shown in figure 26.9, but now the Disconnect paired reads option in the side-panel of the reads track has been ticked, so that the two reads in the paired read are shown disconnected.*

26.1.3 Coverage table

Besides standard information such as position etc, the coverage table (figure 26.11) 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 26.2.3).

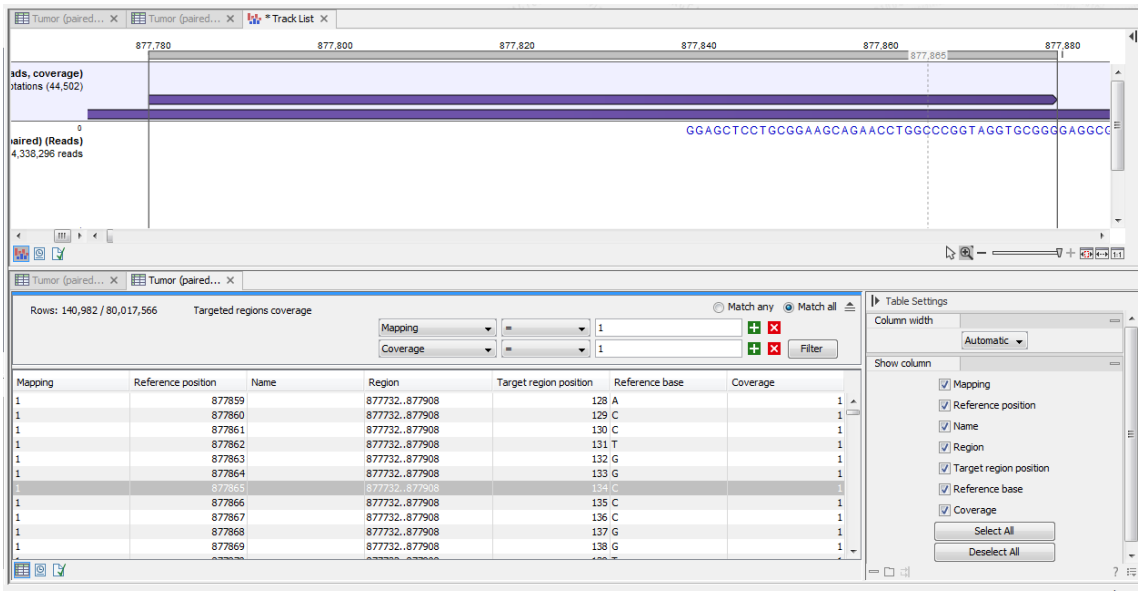


Figure 26.11: *The targeted region coverage table for the same region as shown in same as shown in figures 26.9 and figure 26.10.*

In the shown figure, the coverage table is shown in split view with a track list. The coverage track has been opened from the track list by clicking once on the name found in the left side in the track list. When opening a read mapping in split view from a track list, the two views are linked, this means that when you click on an entry in the table, this position will be brought into focus in the track list.

For information about how to create a track list, please see section 24.3.

26.1.4 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 coverage on the y-axis (see figure 26.12). The x-axis and y-axis values are identical to those found in the columns of the coverage table.

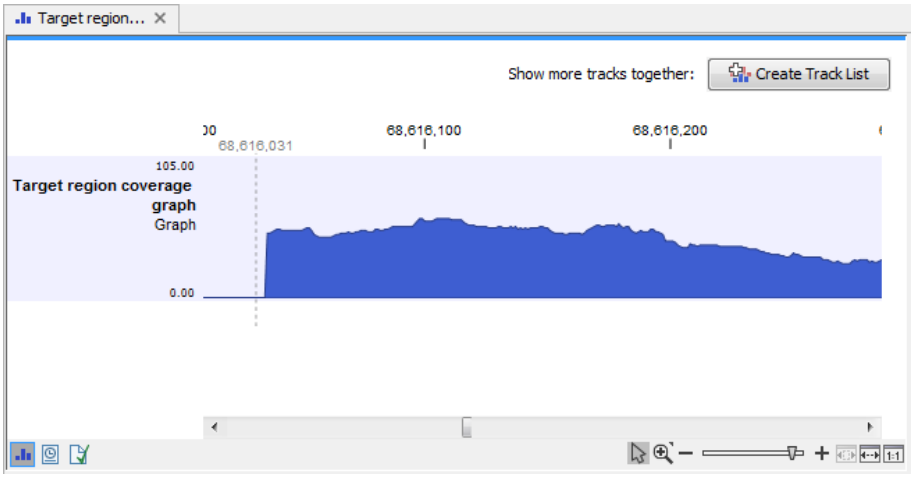


Figure 26.12: *An example of a targeted region coverage graph.*

26.2 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:

Toolbox | **Quality Control** ( ) | **QC for Read Mapping ( )**

This opens a dialog where you can select mapping results ( )/ ( )/ ( ), simples contigs from a de novo assembly, or RNA-Seq analysis results ( ).

In the next wizard window "Set contig group" (figure 26.13), 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.

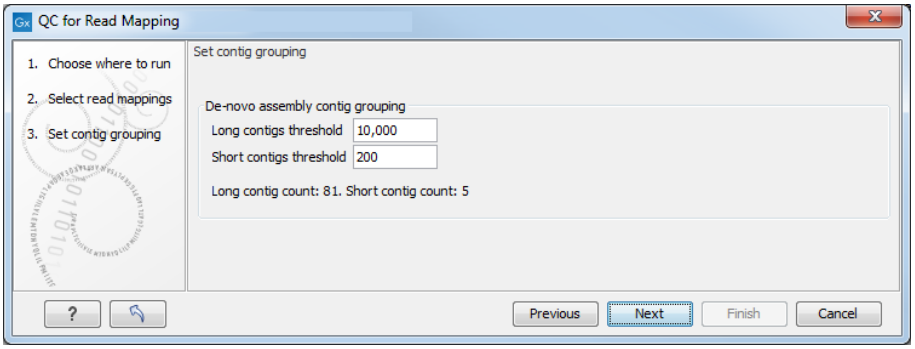


Figure 26.13: *Parameters for mapping reports.*

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 26.14), 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).

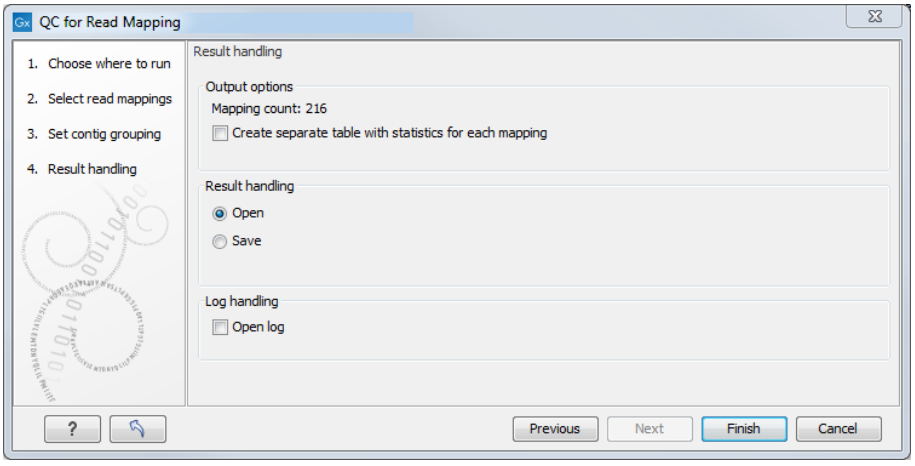


Figure 26.14: *Result handling options.*

The first section of the detailed mapping report is a summary of the statistics:

- Reference count
- Type
- Total reference length
- 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.

26.2.1 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 26.15, 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".

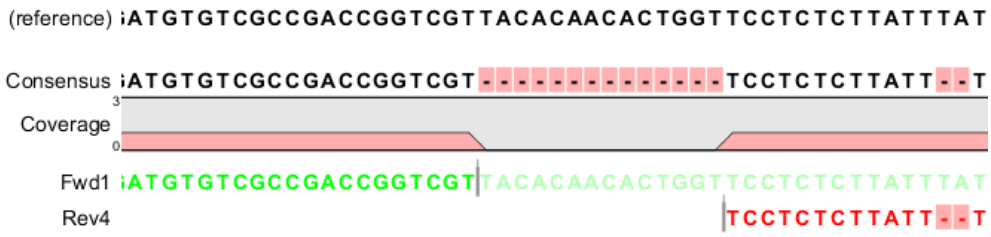


Figure 26.15: *A region of zero coverage in the middle and one time coverage on each side. Note that the gaps to the very right are within the same read which means that these two positions on the reference sequence are still counted as "covered".*

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 26.16).

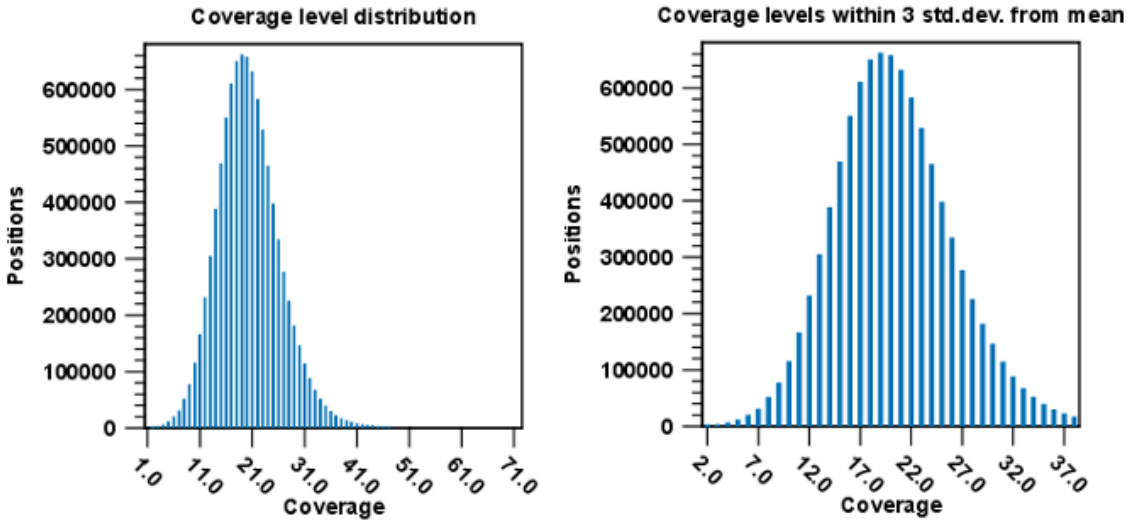


Figure 26.16: *Distribution of coverage - to the left for all the coverage levels, and to the right for coverage levels within 3 standard deviations from the mean.*

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 26.17). 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.

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 26.2.2 below).

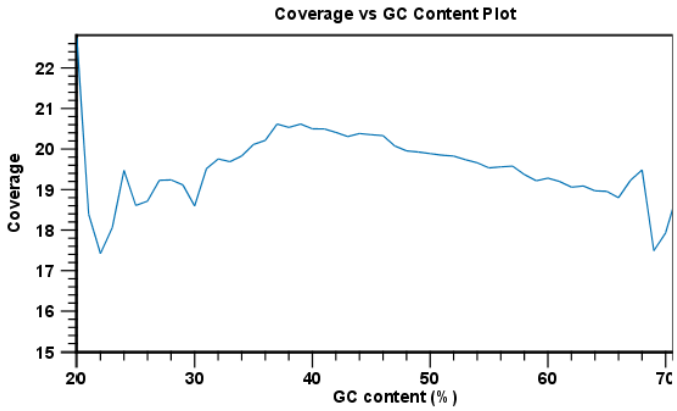


Figure 26.17: *The plot displays, for each GC content level (0-100 %), the mean read coverage of 100bp reference segments with that GC content.*

26.2.2 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 26.18), the distance includes both the read sequence and the insert between them as explained in section 6.3.7.

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 26.19 and figure 26.20.

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

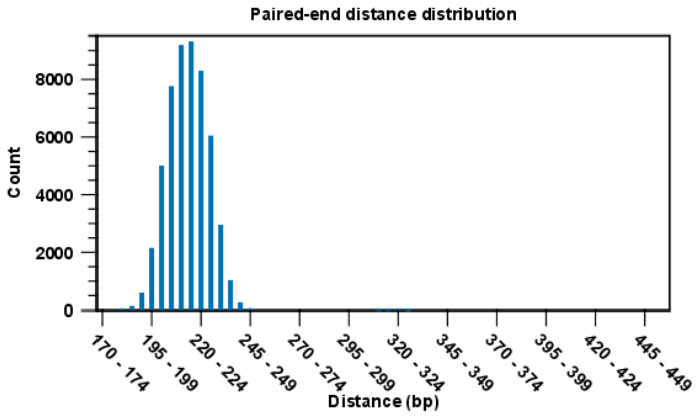


Figure 26.18: *A bar plot showing the distribution of distances between intact pairs.*

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 26.19.

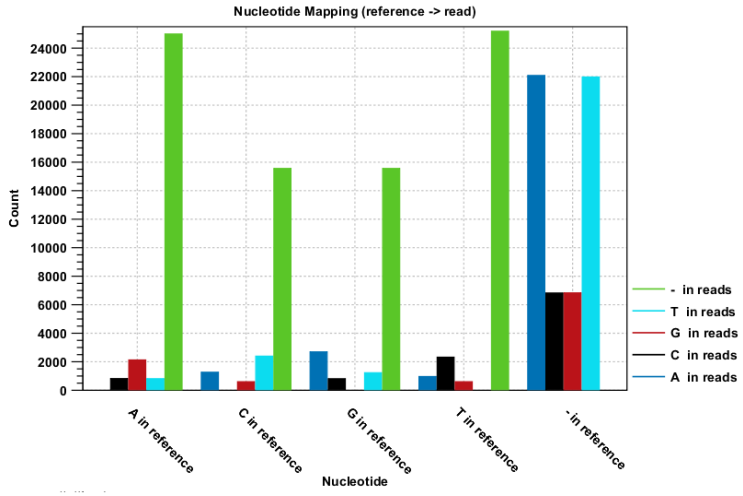


Figure 26.19: *The As and Ts are more often substituted with a gap in the sequencing reads than C and G.*

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 26.20 shows.

The last plots section deals with unaligned read lengths.

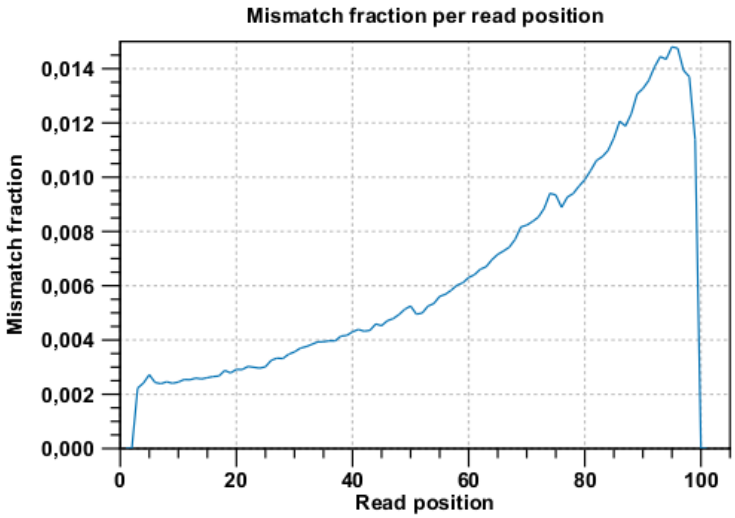
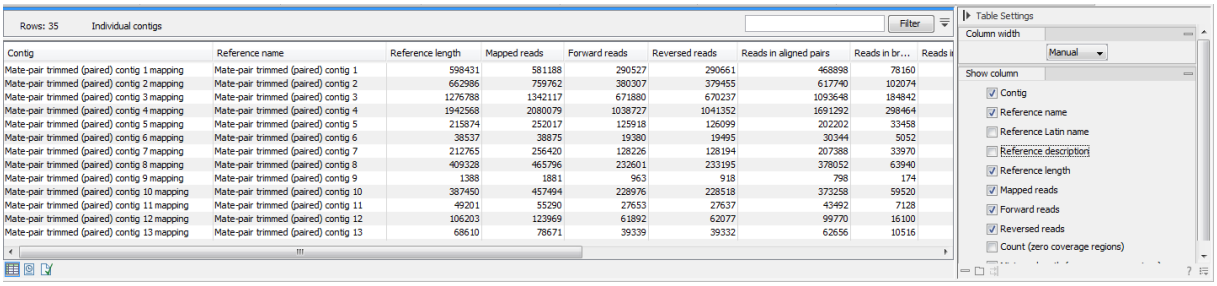


Figure 26.20: *There are mismatches towards the end of the reads.*

26.2.3 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 26.21).



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	670237	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	1388	1881	963	918	798	174
Mate-pair trimmed (paired) contig 10 mapping	Mate-pair trimmed (paired) contig 10	387450	457494	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 26.21: *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.

26.3 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.

To run the Whole Genome Coverage Analysis tool:

Toolbox |**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 26.22.

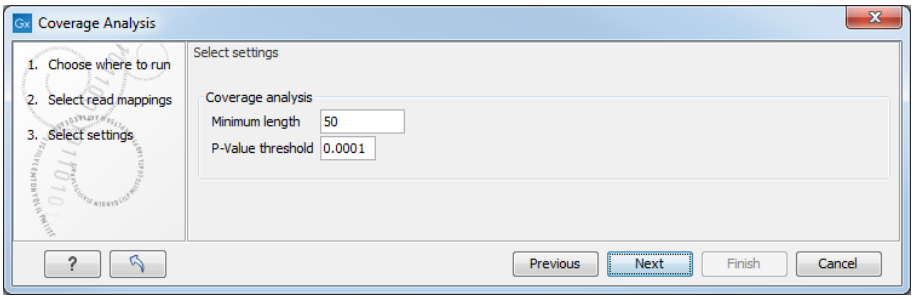


Figure 26.22: *Specify the p-value cutoff.*

Set the p-value and minimum length cutoff. Click **Next** and specify the result handling (figure 26.23).

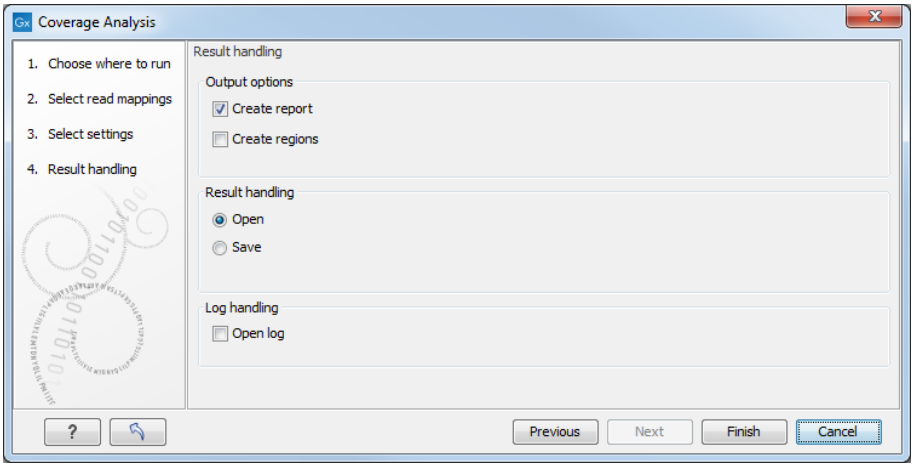


Figure 26.23: *Specify the output.*

Selecting the "Create report" will generate the report made of 2 tables (figure 26.24). 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. 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.

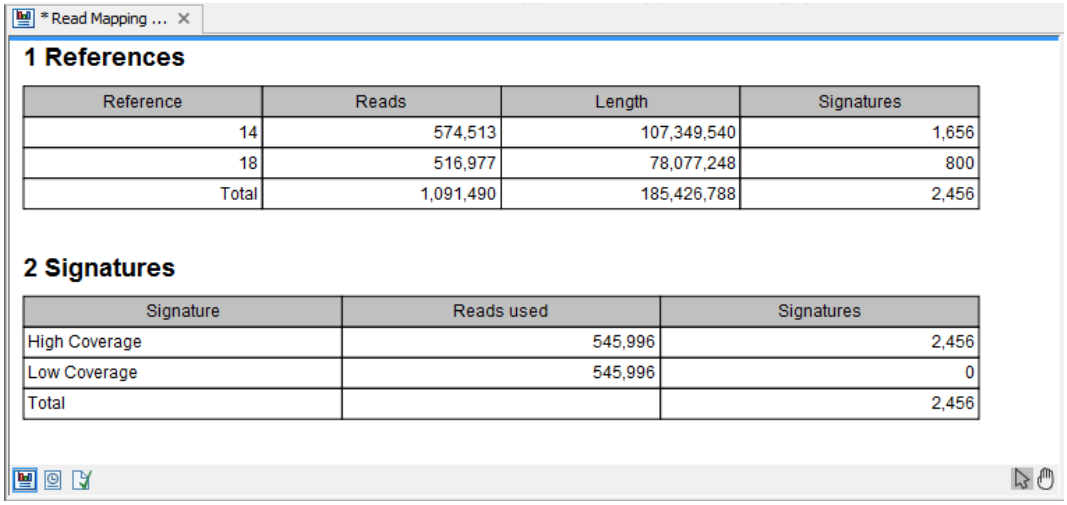


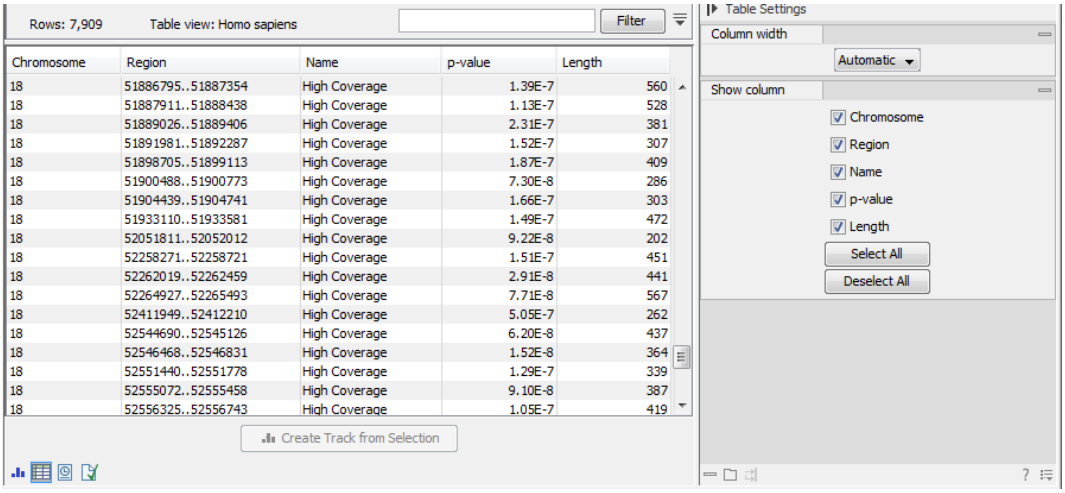
Figure 26.24: *The report output.*

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 26.25.

The Whole Genome Coverage Analysis table includes the following columns (figure 26.25):

- **Chromosome** The name is taken from the reference sequence used for mapping



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

Figure 26.25: *The table output with detailed information on each region.*

- **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.

26.4 Combine Reports

The **Combine Reports** tool makes it easy to get a cross-sample overview by summarizing reports from multiple samples. The tool takes in multiple reports and generates a single report containing summaries and other selected information from the original reports with outliers highlighted.

To create a sample report for a single sample, use the tool **Create Sample Report** described in section 26.5. The individual sample reports, which combine information from multiple report types for a single sample, can then be used as input to the **Combine Reports** tool to generate a combined report that provides a comprehensive overview of results.

Reports produced by the *CLC Genomics Workbench* tools listed in section 26.4.2 can be used as input, as can reports generated by some tools delivered by plugins developed by QIAGEN.

Creating a combined report

To create a combined report, go to:

Toolbox |**Quality Control** ( ) | **Combine Reports** ( )

In the dialog that opens, select the reports to be combined (figure 26.26).

In the next dialog, configuration options are presented, as shown in figure 26.27:

The configuration options are:

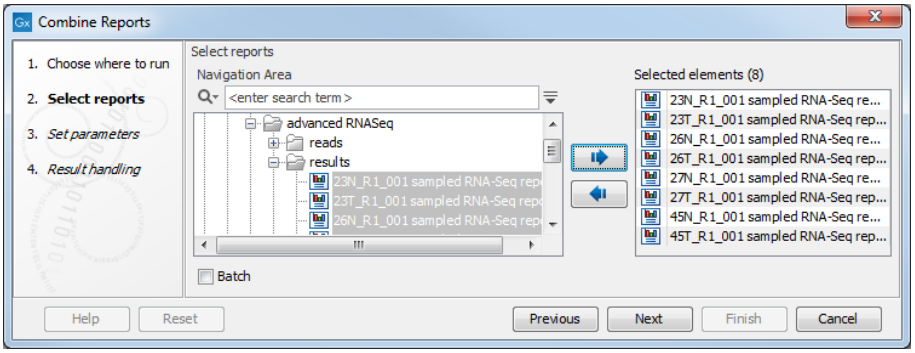


Figure 26.26: *The reports to be combined are selected as input.*

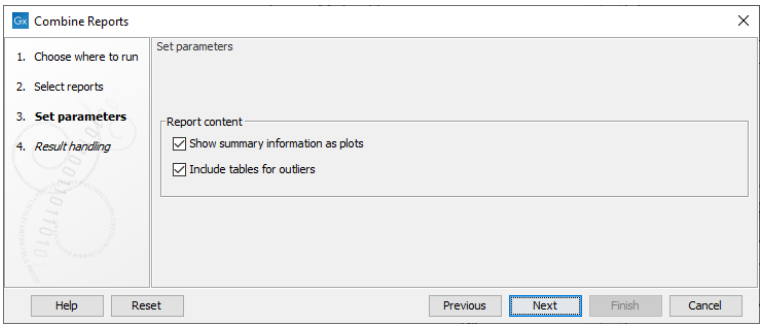


Figure 26.27: *Configuration options for combining reports.*

- **Show summary information as plots** Enable this to display information as box plots instead of tables, if possible for a given data value.
- **Include tables for outliers** Enable this to add a table after each summary table or box plot, containing the samples that are outliers for that data type, if possible for a given data value.

  See section 26.4.1 for further details.

26.4.1 Combine Reports output

The **Combine Reports** tool generates a single report, (), containing summaries and other selected information from the reports provided as input.

The report is divided into one or more sections depending on the input report type.

By default, summary information is displayed in table format. Each table presents a summary of the corresponding information from the input reports.

The tables will contain one row per input report, the first column indicating the sample same taken from the corresponding input report, and additional rows, shaded in pale gray, which report the minimum, median, maximum, mean and standard deviation for all numeric columns (figure 26.28).

If the **Show summary information as plots** option was enabled, tables will be displayed as box plots, wherever possible. Each numeric column will be presented as one box in the plot (figure 26.28).

Combined reports contain only data from report types supported by the **Combined Reports** tool. Where some of the reports supplied as inputs are supported and some are not, the combined

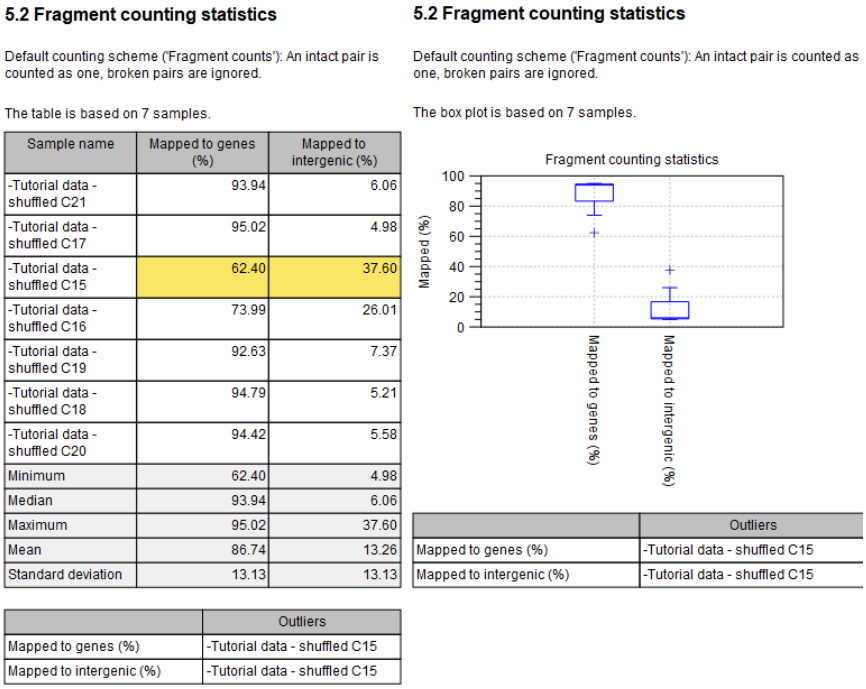
report will contain information only from the supported ones. Supported report types are listed in section 26.4.2. If none of the reports provided are of supported types, the report will contain a statement saying this.

Outliers and other highlighted entries in combined reports

Table cells are colored if the data they contain is considered an outlier or problematic.

Cells containing outliers are highlighted in yellow. Outliers are those outside the range: lower quartile - 1.5 IQR, upper quartile + 1.5 IQR.

If the **Include tables for outliers** option was enabled, any table or plot containing outliers will be followed by an additional table containing the names of the outliers. A row is provided for each column containing outliers (figure 26.28).



5.2 Fragment counting statistics

Default counting scheme ('Fragment counts'): An intact pair is counted as one, broken pairs are ignored.

The table is based on 7 samples.

Sample name	Mapped to genes (%)	Mapped to intergenic (%)
-Tutorial data - shuffled C21	93.94	6.06
-Tutorial data - shuffled C17	95.02	4.98
-Tutorial data - shuffled C15	62.40	37.60
-Tutorial data - shuffled C16	73.99	26.01
-Tutorial data - shuffled C19	92.63	7.37
-Tutorial data - shuffled C18	94.79	5.21
-Tutorial data - shuffled C20	94.42	5.58
Minimum	62.40	4.98
Median	93.94	6.06
Maximum	95.02	37.60
Mean	86.74	13.26
Standard deviation	13.13	13.13

	Outliers
Mapped to genes (%)	-Tutorial data - shuffled C15
Mapped to intergenic (%)	-Tutorial data - shuffled C15

5.2 Fragment counting statistics

Default counting scheme ('Fragment counts'): An intact pair is counted as one, broken pairs are ignored.

The box plot is based on 7 samples.

	Outliers
Mapped to genes (%)	-Tutorial data - shuffled C15
Mapped to intergenic (%)	-Tutorial data - shuffled C15

Figure 26.28: *By default, summary information is reported in table format (left). Tables of numerical data in cross-sample reports contain summary rows, shaded in pale gray. Cells containing values identifed as outliers are highlighted in yellow. Box plots are generated instead of tables for numerical data if the "Show summary information as plots" option was enabled (right). The tables at the bottom, containing summary information about outliers, are present because the reports were generated with the "Include tables for outliers" option enabled.*

Cells highlighted in pink indicate that a problem has been identified. This is explained underneath the table for each column that contains pink cells. If a cell is both an outlier and problematic, it will be highlighted in red (figure 26.29).

26.4.2 Report types supported

Reports generated by the following tools in *CLC Genomics Workbench 21.0.1* can be combined using the **Combine Reports** and **Create Sample Report** tools. In addition to the list below, reports generated by some tools delivered by plugins are also supported.

5.4 Strand specificity

The table is based on 7 samples.

Sample name	Strand specific setting	Forward reads mapped (%)	Reverse reads mapped (%)	Ignored reads (wrong strand) (%)
-Tutorial data - shuffled C21	Forward	100	0	5.83
-Tutorial data - shuffled C17	Forward	100	0	4.01
-Tutorial data - shuffled C15	Forward	100	0	39.39
-Tutorial data - shuffled C16	Forward	100	0	25.46
-Tutorial data - shuffled C19	Forward	100	0	4.05
-Tutorial data - shuffled C18	Forward	100	0	6.27
-Tutorial data - shuffled C20	Forward	100	0	6.24
Minimum	-	100.00	0.00	4.01
Median	-	100.00	0.00	6.24
Maximum	-	100.00	0.00	39.39
Mean	-	100.00	0.00	13.04
Standard deviation	-	0.00	0.00	13.88

Ignored reads (wrong strand) (%): > 25% of reads were filtered away due to the strand specific setting.
- If a strand-specific protocol has not been used, re-run the tool with strand specific setting "Both".
- If a strand-specific protocol has been used, library preparation may have failed.

	Outliers
Ignored reads (wrong strand) (%)	-Tutorial data - shuffled C15

Figure 26.29: *Table with both problematic cells and outliers. The table is followed by an explanation for the column "Ignored reads (wrong strand) (%)" containing problematic cells.*

In addition, **Combine Reports** can also take as input reports previously combined reports, as well as sample reports created by the **Create Sample Report**, allowing the generation of comprehensive, cross-sample summary reports[1].

- Call Methylation Levels
- Copy Number Variant Detection (CNVs)
- Create Variant Track Statistics Report
- Demultiplex Reads
- De Novo Assembly
- Indels and Structural Variants
- Map Bisulfite Reads to Reference
- Map Reads to Reference
- QC for Read Mapping
- QC for Sequencing Reads
- QC for Targeted Sequencing
- RNA-Seq Analysis

[1]Combine Reports and Create Sample Report supports reports generated using *CLC Genomics Workbench* version 20.0 and corresponding versions of plugins and server software. Information from reports generated by earlier software versions will not appear in the output.

- Remove Duplicate Mapped Reads
- Trim Reads
- Trim Sequences
- Whole Genome Coverage Analysis

If you wish to include information from other report types, or combine information in a customized way, reports can be exported to JSON format, as described in section 6.6.8.

26.5 Create Sample Report

Create Sample Report takes in multiple reports relating to a single sample and creates a summary report that includes selected information from the original reports.

In addition, specific types of information can be selected for inclusion in the Quality Control section of the sample report. The type of information to include is specified when launching the tool. Thresholds can also be configured. When this is done, the Quality Control section of the report can be quickly scanned to see if thresholds were met or not for each value type.

Reports produced by the *CLC Genomics Workbench* tools listed in section 26.4.2 can be used as input, as can reports generated by some tools delivered by plugins developed by QIAGEN.

Creating a sample report

To create a sample report, go to:

Toolbox | **Quality Control** ( ) | **Create Sample Report** ( )

In the dialog that opens, select the reports to use as input (figure 26.30).

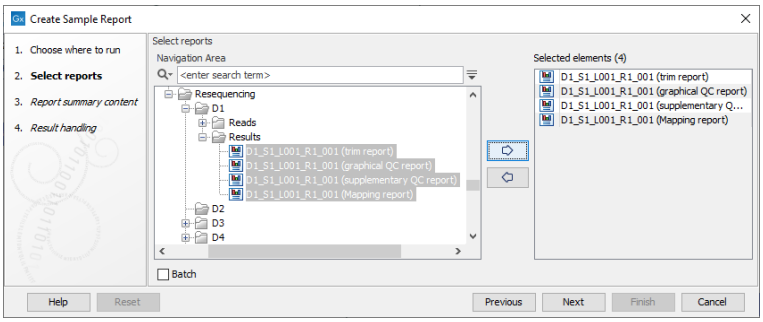


Figure 26.30: *Reports about a particular sample have been selected as input.*

In the next dialog, contents to include in the optional Quality Control section of the report can be configured (figure 26.31).

Check the boxes in the **Include** column for the information types to include. If no boxes are checked, a Quality Control section will not be included in the sample report. The information types listed reflect the tools of the *CLC Genomics Workbench* and any installed plugins that generate supported report types.

For each information type, quality control criteria can, optionally, be specified by entering a value in the **QC threshold** column. When this is done, the threshold value specified is included in the

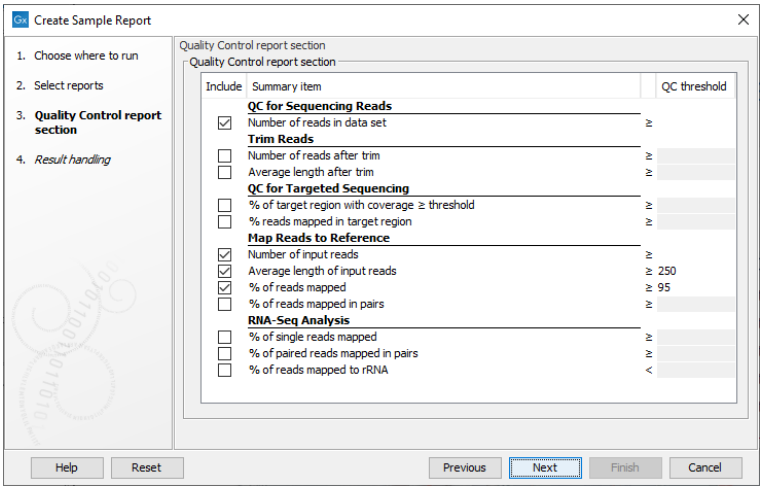


Figure 26.31: *Items selected in this dialog are included in the Quality Control section of the sample report.*

report, and its cell is colored green if the threshold is met or exceeded, or yellow if it is not (figure 26.32).

26.5.1 Create Sample Report output

Create Sample Report generates a single report, ( ) that includes a section for each of the supported report types selected as input.

If information types were selected for inclusion in a Quality Control section, this section will appear at the start of the report. If QC thresholds were set, these will be reported in the Threshold column, with cells colored green if the threshold was met or exceeded, and yellow if it was not (figure 26.32).

Sample reports contain only information from report types supported by the **Create Sample Report** tool, listed in section 26.4.2. If unsupported report types are supplied as input, information from these is not included in the report and a note of this is written to the log. If none of the reports provided are of supported types, the report will contain a statement saying this.

If you wish to combine information from reports not supported by this tool, or combine information in a customized way, reports can be exported to JSON format. See section 6.6.8.

1 Summary

Summary for D1_S1_L001_R1_001 (trim report)

1.1 Quality Control

Summary item	Value	Threshold
Number of reads in data set	402,431	N/A
Number of input reads, Map Reads to Reference	402,428	N/A
Average length of input reads, Map Reads to Reference	30.73	250.00
Percentage of reads mapped	98.89	95.00

2 Reads summary

2.1 Summary statistics

Sample name	Data sets (#)	Reads (#)	Paired reads (%)	Bases (#)
D1_S1_L001_R1_001 (trim report)	1	402,431	0	13,280,223

2.2 Read length distribution

Sample name	Minimum	25th percentile	Median	75th percentile	Maximum	Mean
D1_S1_L001_R1_001 (trim report)	33	33	33	33	33	33

Figure 26.32: *In the Quality Control report section. Here, QC thresholds were specified for 2 of the types of information requested. In one case (green), the threshold was met or exceeded, while in the other (yellow), it was not.*

Chapter 27

Read mapping

Contents

27.1 Map Reads to Reference 653
27.1.1 Selecting the reads 653
27.1.2 References and masking 653
27.1.3 Mapping parameters 654
27.1.4 Mapping paired reads 657
27.1.5 Non-specific matches 659
27.1.6 Gap placement 659
27.1.7 Mapping computational requirements 660
27.1.8 Reference caching 661
27.1.9 Mapping output options 661
27.1.10 Summary mapping report 662
27.2 Reads tracks and stand-alone read mappings 664
27.2.1 Coloring of mapped reads 664
27.2.2 Reads tracks 665
27.2.3 Stand-alone read mapping 673
27.3 Local Realignment 680
27.3.1 Method 680
27.3.2 Realignment of unaligned ends 680
27.3.3 Guided realignment 681
27.3.4 Multi-pass local realignment 682
27.3.5 Known limitations 683
27.3.6 Computational requirements 684
27.3.7 Run the Local Realignment tool 685
27.4 Merge Read Mappings 688
27.5 Remove Duplicate Mapped Reads 688
27.5.1 Algorithm details and parameters 689
27.5.2 Running the duplicate reads removal 691
27.6 Extract Consensus Sequence 692

27.1 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 special modes for handling PacBio reads and reads longer than 500bp. Before the Map Reads to Reference tool starts to map the reads, it checks the input sequence list(s) to decide on the mapping algorithm to use:

- If the input sequence list(s) have the read group set to "PacBio", then the specialized mapping algorithm which is better suited for mapping long reads with many sequencing errors is applied.
- If the input sequence list(s)' read group is not set to "PacBio", then the reads are mapped using our standard mapping algorithm. The standard mapping algorithm uses the same seeding method for all input reads, but different extension methods for long (>500 bp) and short reads.

It is possible to mix sequence lists that have the platform field of the read group "PacBio" with sequence lists that have a different read group for the same mapping. In this case the appropriate mapping algorithm will be applied to each of the sequence lists.

27.1.1 Selecting the reads

To start the read mapping:

Toolbox | **Resequencing Analysis** ( ) | **Map Reads to Reference ( )**

In the first dialog, select the sequences or sequence lists containing the sequencing data (figure 27.1). Please note that reads longer than 100,000 bases are not supported.

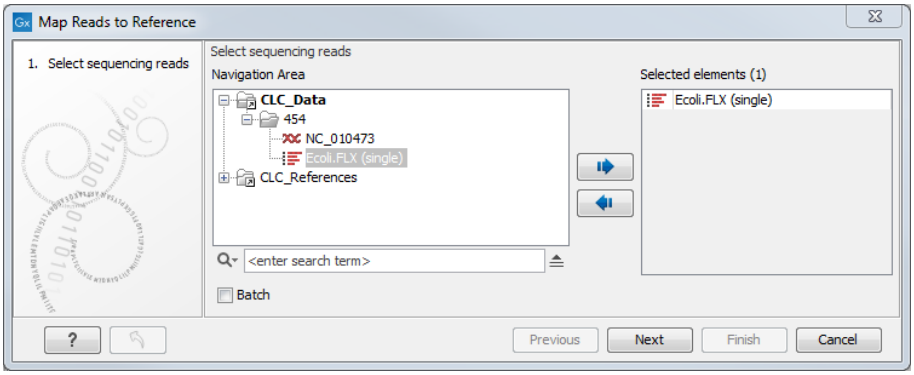


Figure 27.1: *Specifying the reads as input. You can also choose to work in batch.*

27.1.2 References and masking

When the sequences are selected, click **Next**, and you will see the dialog shown in figure 27.2.

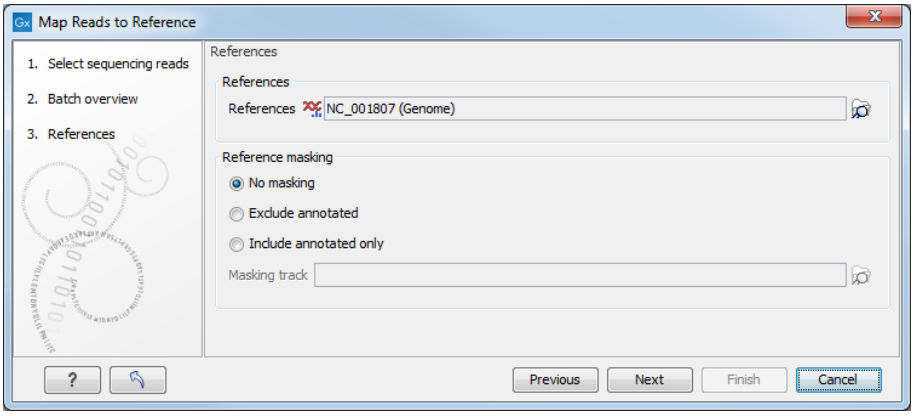


Figure 27.2: *Specifying the reference sequences and masking.*

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)

The next part of the dialog shown in figure 27.2 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 24.7).

27.1.3 Mapping parameters

Clicking **Next** leads to the parameters for the read mapping (see figure 27.3).

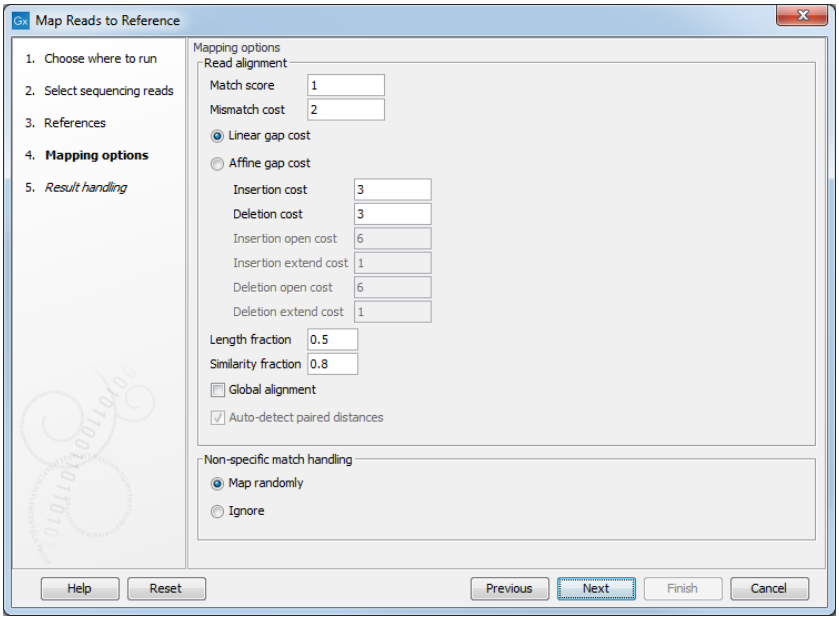


Figure 27.3: *Setting parameters for the mapping.*

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, 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 $\ell \cdot$ `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 $\ell \cdot$ `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
$\texttt{Insertion open cost} + \ell \cdot \texttt{Insertion extend cost}$
and a deletion of length ℓ is penalized by
$\texttt{Deletion open cost} + \ell \cdot \texttt{Deletion extend cost}$
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.

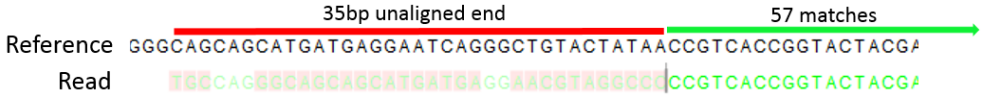


Figure 27.4: *An alignment of a read where a region of 35bp at the start of the read is unaligned while the remaining 57 nucleotides matches the reference.*

Figure 33.20 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 33.21 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 33.20 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 33.21 would become 61 and 58, respectively, and thus make it optimal.

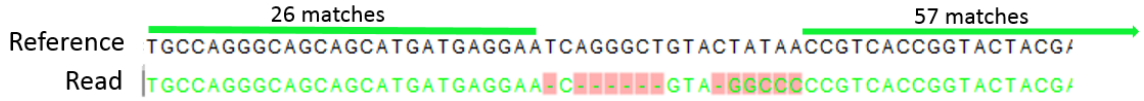


Figure 27.5: *An alignment of a read containing a region with several mismatches and deletions. By reducing the default cost of either mismatches or deletions the read mapper can make an alignment that spans the full length of the read.*

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 33.22).

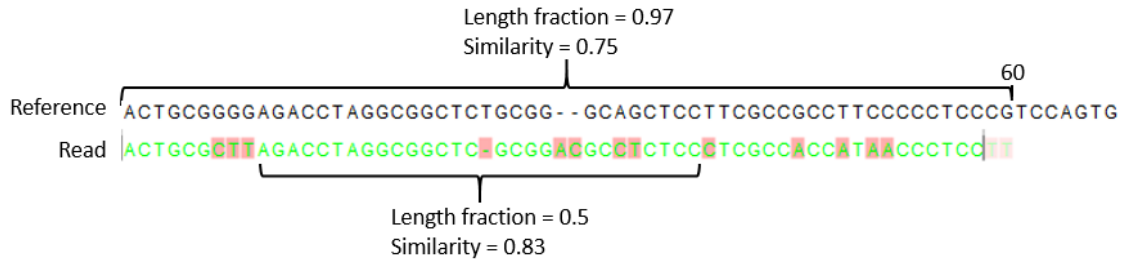


Figure 27.6: *A read containing 59 nucleotides where the total alignment length is 60. The part of the alignment that gave rise to the optimal score has length 58 which excludes 2 bases at the left end of the read. The length fraction of the matching region in this example is therefore 58/60 = 0.97. 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 27.3.

27.1.4 Mapping paired reads

- **Auto-detect paired distances** At the bottom of the dialog shown in figure 27.3 you can specify how **Paired reads** should be handled. You can read more about how paired data is imported and handled in section 6.3.7. 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 33.23 shows an example of the distribution of intervals with and without automatic pair distance interval estimation.

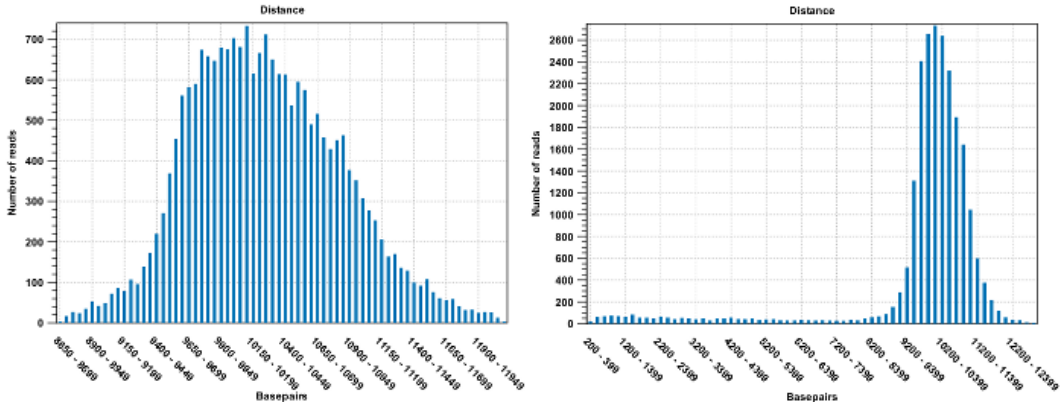


Figure 27.7: *To the left: mapping with a narrower distance interval estimated by the workbench. To the right: mapping with a large paired distance interval (note the large right tail of the distribution).*

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 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 (see section 6.3.7).

If a large portion of pairs are flagged 'Broken' we recommend the following:

1. Inspect the detailed mapping report (see section 26.2) 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 26.2 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 27.1.5 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.

27.1.5 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 27.1.4).

When looking at the mapping, the default color for non-specific matches is yellow.

27.1.6 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 33.24.

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 33.25.



Figure 27.8: *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.*



Figure 27.9: *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.*

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 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 29.5.1.

27.1.7 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.

27.1.8 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**

27.1.9 Mapping output options

Clicking **Next** lets you choose how the output of the mapping should be reported (see figure 27.10).

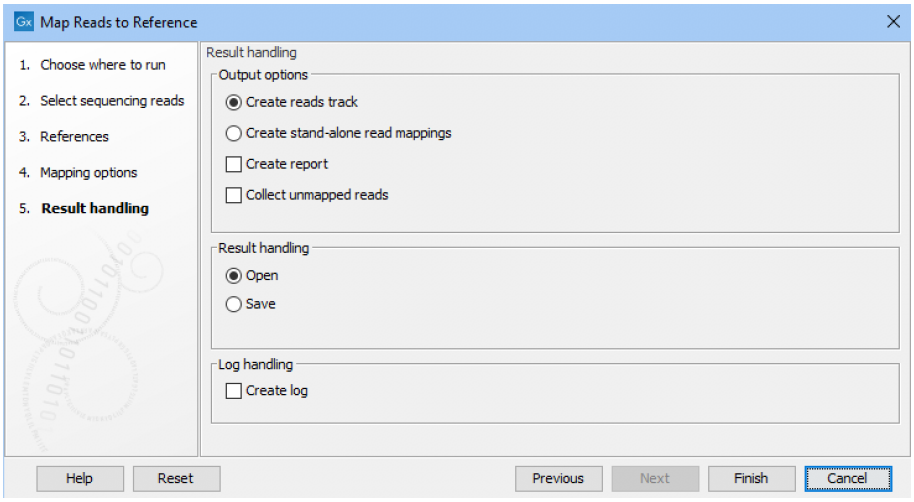


Figure 27.10: *Mapping output options.*

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 24. 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 19.7). 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 27.2.3).

Read more about both output types in section 27.2. 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 24.7).

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 27.1.10.
- **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.

27.1.10 Summary mapping report

If you choose to create a report as part of the read mapping (see section 27.2), this report will summarize the results of the mapping process. An example of a report is shown in figure 27.11. 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

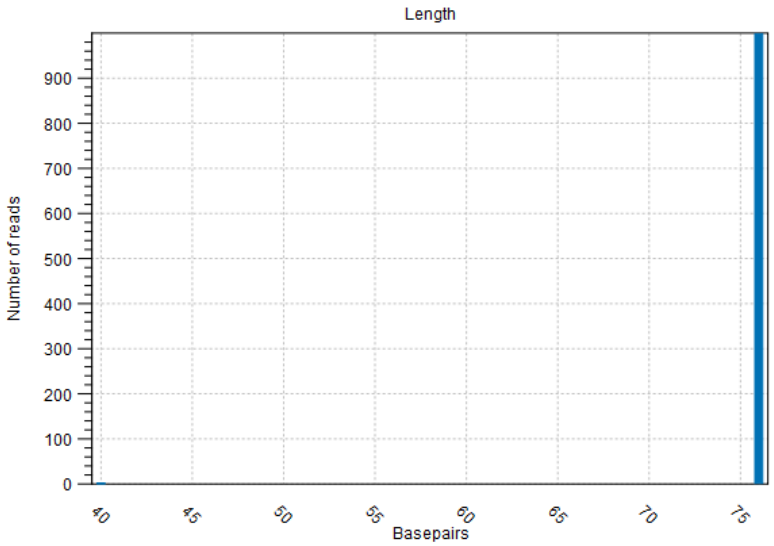


Figure 27.11: *The summary mapping report.*

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.

27.2 Reads tracks and stand-alone read mappings

This section describes the common features and differences that exist between the two types of output that are generated by the Map Reads to Reference tool.

As a reminder, here are the main differences between tracks and mappings.

Reads track

- seamlessly integrates with subsequent analyses
- best viewed in a Track List for comparison with other tracks
- an **overflow graph** above the reads in the reads track shows reads that cannot be displayed in the track

Stand-alone read mapping

- rich visualization and editing features (see section 19.7)
- includes a **consensus sequence** and a **coverage graph**

Remember that it is possible to convert to and from tracks (see section 24.7).

27.2.1 Coloring of mapped reads

The mapped reads are colored by default according to the following color code (see figure 27.12 for a simplified illustration of the color scheme):

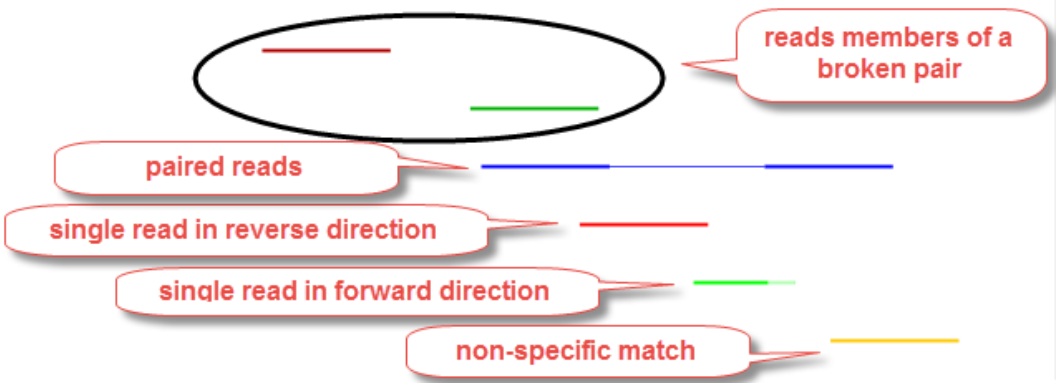


Figure 27.12: *Default reads color in a stand-alone read mapping where the read layout is not set to "Packed": you can see that members of a broken pairs are in darker shades than the single reads.*

- Single reads mapping in their **forward** direction are **green**.
- Single reads mapping in their **reverse** direction are **red**.
- **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 "overrule" 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 27.13). **Insertions** with a frequency lower than 1% are shown with a **black vertical line**.

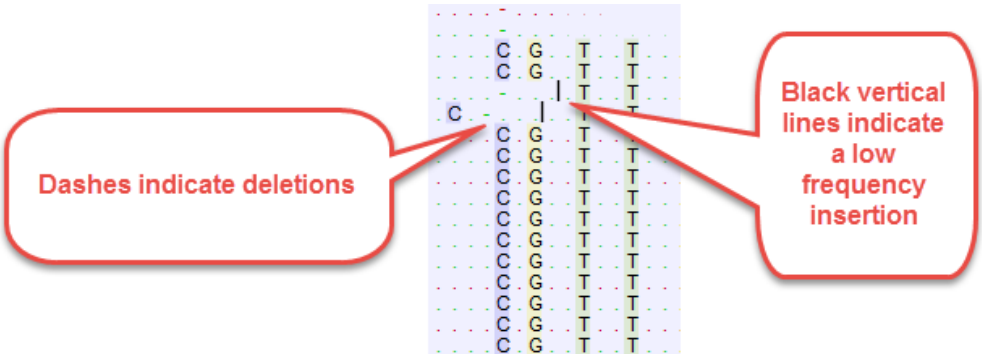


Figure 27.13: *Reads track showing deletions as dashed lines, and low frequency insertions as black vertial lines.*

- **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 27.14). Ambiguous bases are in gray.

These default colors can be changed using the side panel as shown in figure 27.15.

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.

27.2.2 Reads tracks

The main advantage of having the output of the read mapping process represented as a track is that this way it seamlessly integrates with other downstream analysis tools. So unless any specific functionality of the stand-alone read mapping is required, we recommend to use the

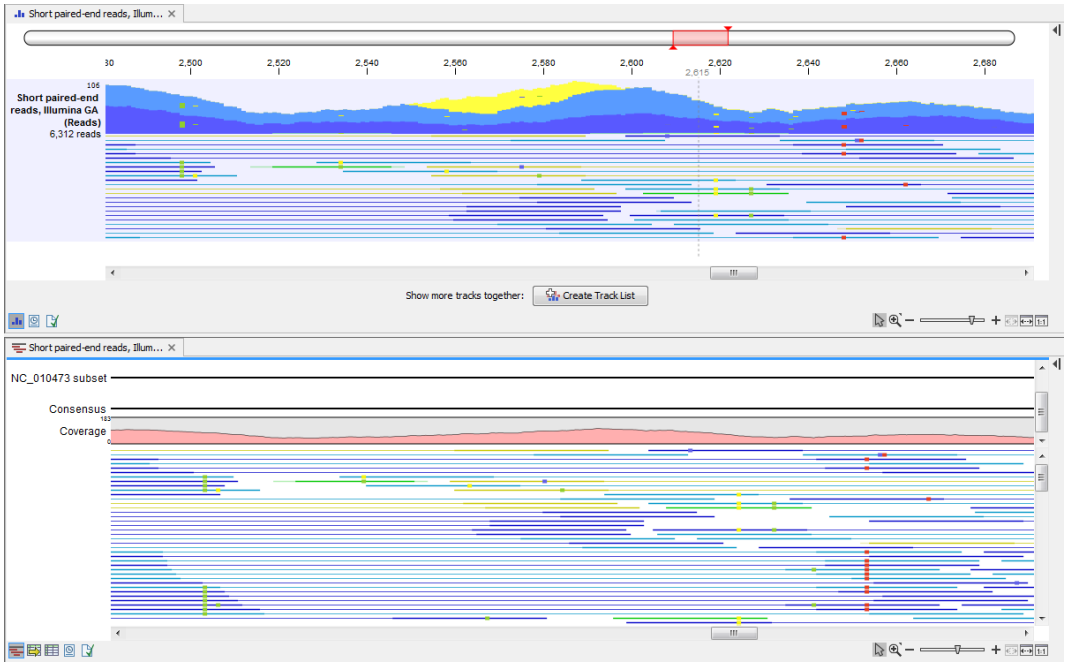


Figure 27.14: *Mismatches between the reads and reference are shown as narrow vertical traits following the Rasmol color scheme. A reads track is shown above, a read mapping below.*

tracks output for the additional flexibility in further analysis. Later it is possible to convert to and from tracks (see section 24.7).

When the option "Create reads track" is chosen when running the tool Map Reads to Reference, only the reads themselves will be saved. It is always possible to add information about the reference, consensus sequence or annotations later by creating a Track List.

A reads track will display two types of different information depending on the zooming level at which it is set.

When zoomed out to the point where the amount of data in the view reaches the data aggregation setting, the track shows aggregated mapped reads (figure 27.16), so the height of the graph reflect the coverage of a particular region on the mapping. In addition, three blue shades differentiate from top to bottom:

- The maximum coverage value (read count) of the positions included in that region.
- The average coverage value of the positions included in that region.
- The minimum coverage value of the positions included in that region.

The data aggregation view is an alternative way of displaying a large amount of data (mapped reads) on the screen in order to shorten the data display time. This aggregated view allows you to navigate the view more smoothly.

In figure 27.17 we have zoomed in on a reads track displayed in a Track List together with targeted regions and variants annotations shown below.

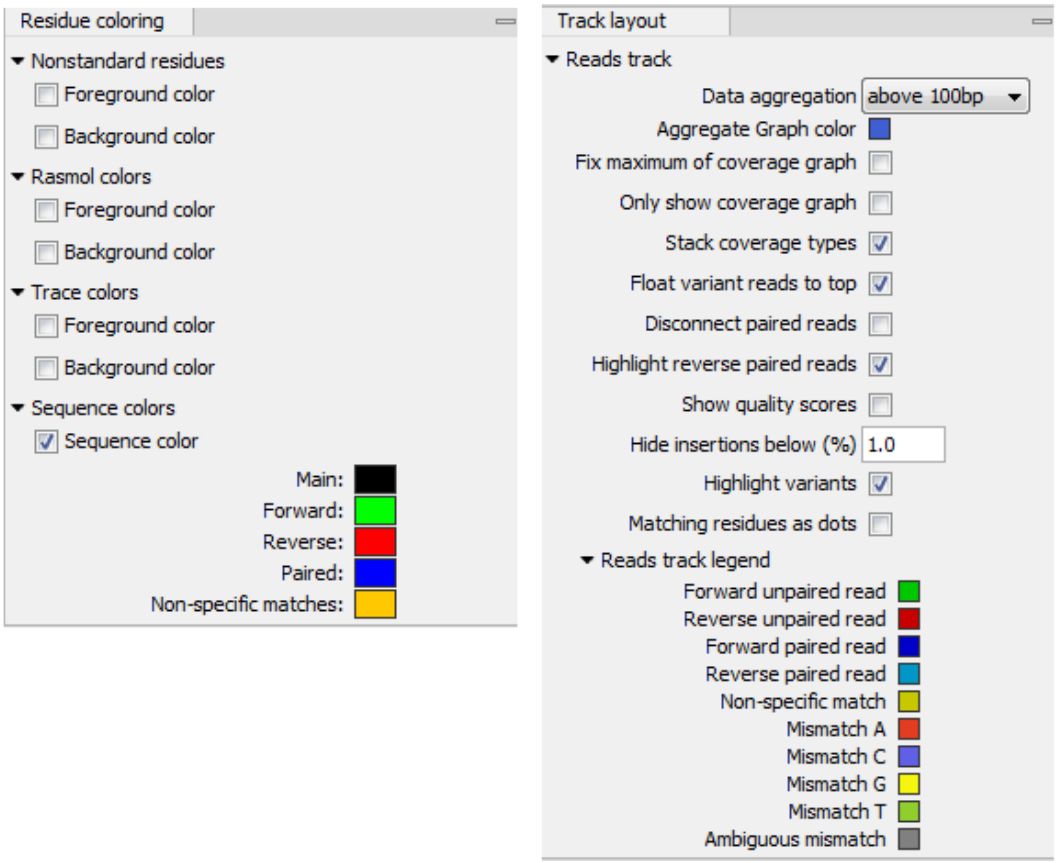


Figure 27.15: *Coloring of mapped reads legends for read mappings (left) and reads track (right). Clicking on a color allows you to change it (except for read mappings at the Packed compactness level.*

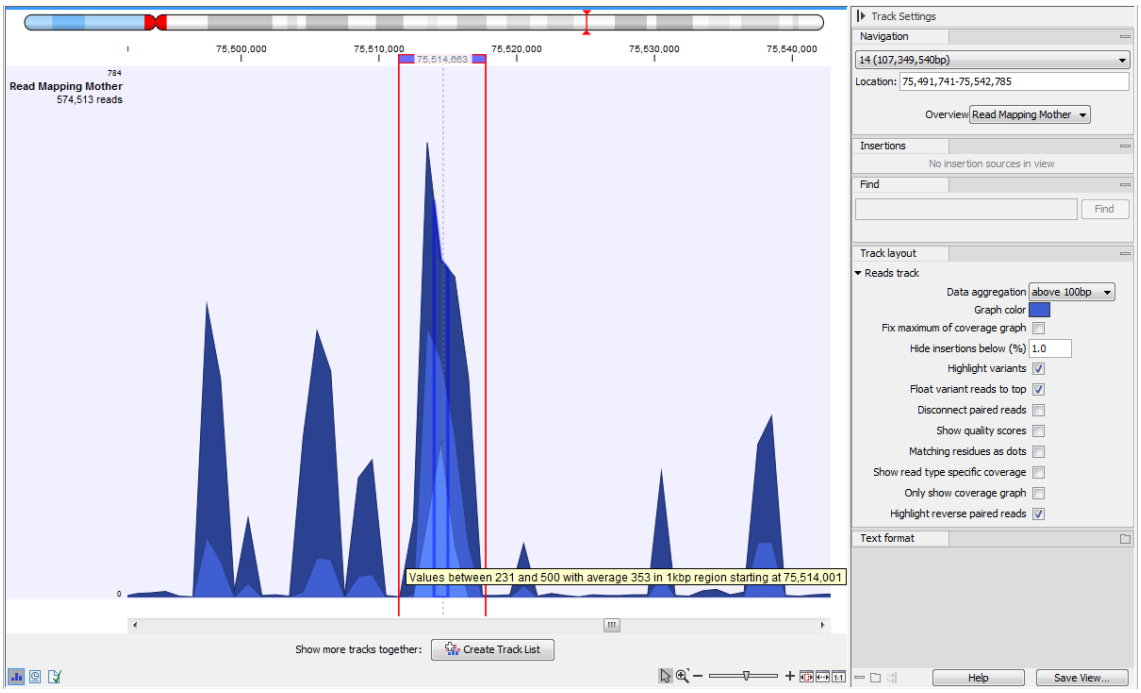


Figure 27.16: *In the aggregated Reads Track, the three blue shades represent minimum, average and maximum coverage values for the aggregated mapped reads, here a 1kb region as stated in the tooltip.*

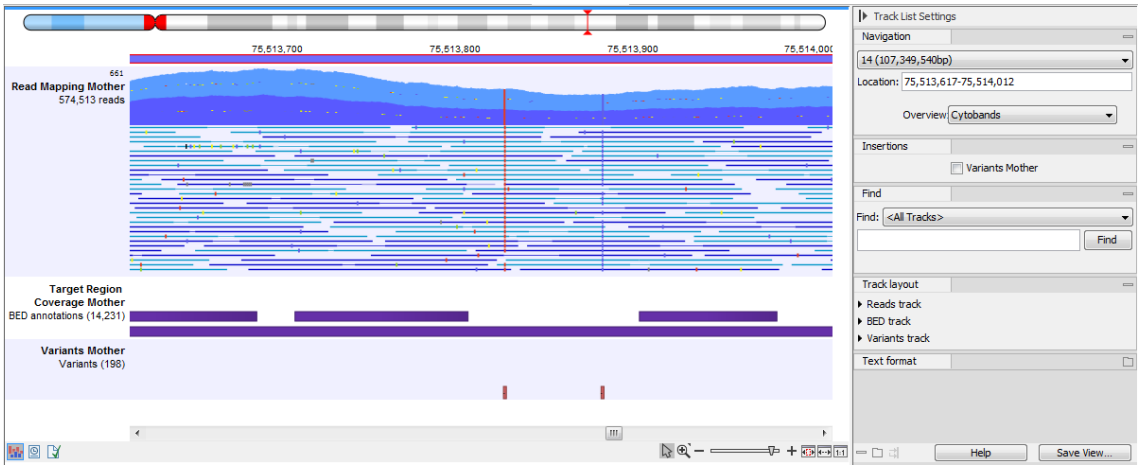


Figure 27.17: *Zooming in on the tracks reveals details.*

If you zoom in further the alignment of the reads and the reference sequence can be viewed at single nucleotide level (see figure 27.18).

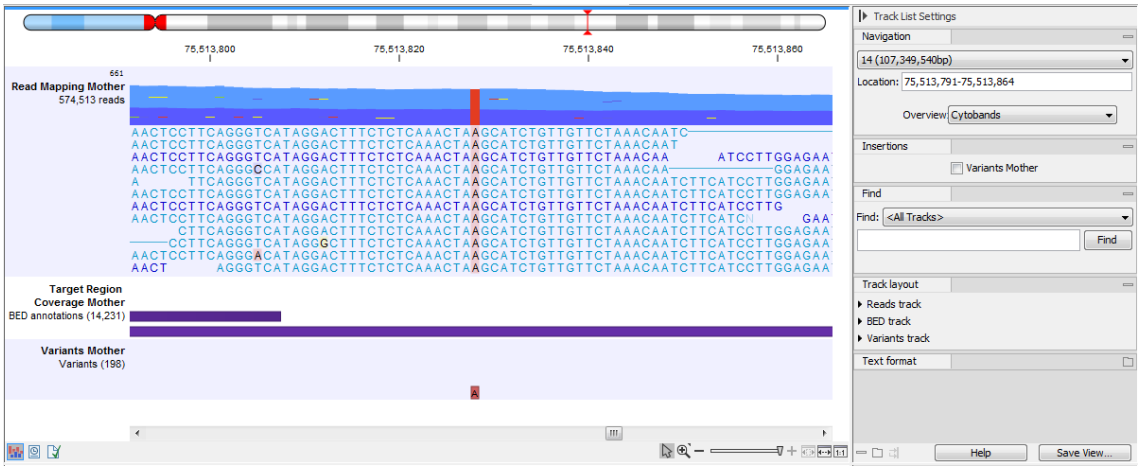


Figure 27.18: *Zoom in to see the bases of the reads and the reference sequence.*

In this case only a certain amount of reads are visible. In order to see more reads, increase the height of the reads track by dragging down the lower part of the track with the mouse. Also, a vertical scroll bar will appear to the right of the reads when hovering on them to navigate through high coverage regions.

Read mappings to circular genomes, and 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 choose in the **Selected Read** menu the following options: Copy, BLAST or Open in a New View the selected portion of the read (figure 27.19).

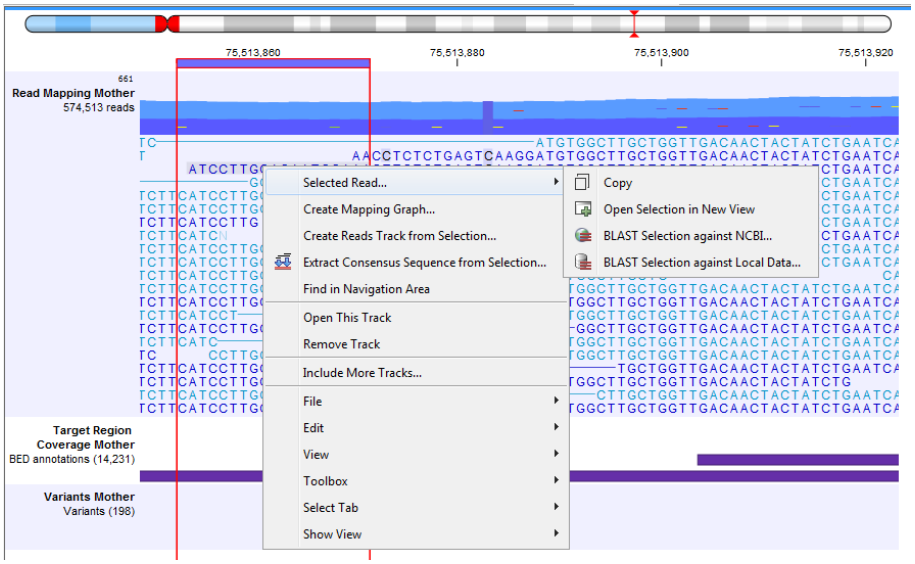


Figure 27.19: *Adjusting the height of the track.*

When hovering over a position in a Reads track shown in non-aggregated view, a tooltip displays

the read counts for each observed nucleotide in that position, together with the directions of the reads with that nucleotide (figure 27.20). Tooltips usually appear after a small wait time once the mouse is on a certain position, but you can hold shift while moving the mouse over the reads to make tooltips appear without any delay.

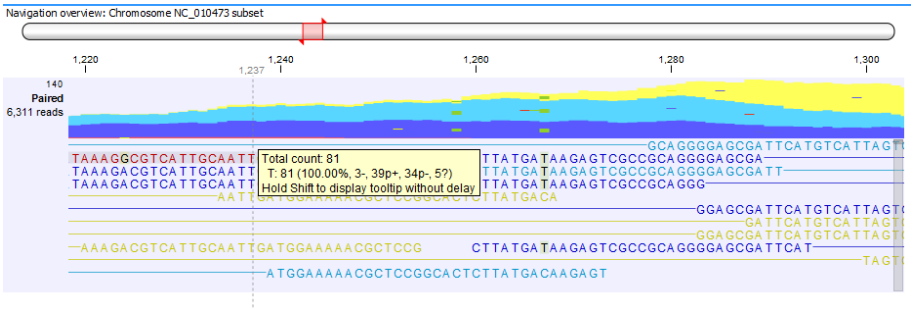


Figure 27.20: *Example of a tooltip available in a non aggregated view of a reads track.*

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

Reads tracks Side Panel settings

The options for the **Side Panel** vary depending on which track is shown in the View Area. In figure 27.21 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 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 27.22). 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.

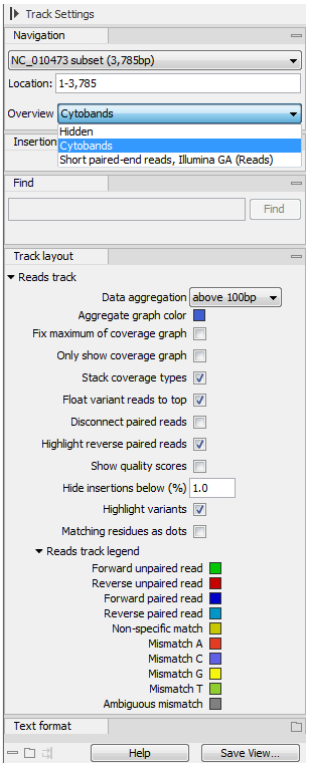


Figure 27.21: *The Side Panel for reads tracks.*

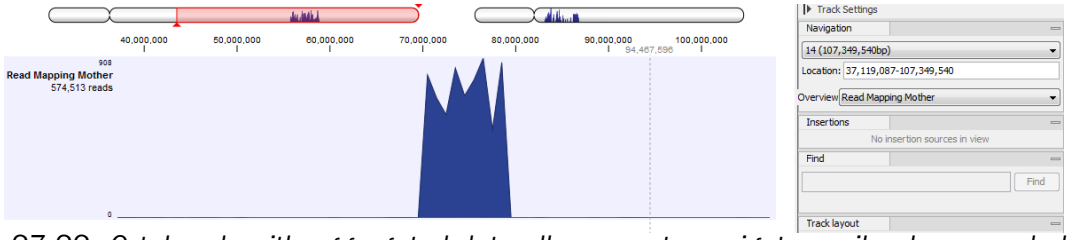


Figure 27.22: *Cytobands with aggregated date allows you to navigate easily where needed based on the data location.*

- **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 27.21 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 as on top of each other.
- **Float variant reads to top.** When checked, reads with variations will appear at the top of the view.
- **Disconnect paired reads.** Disconnects paired end reads (see section 27.2.2).
- **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.
- **Reads track legend.** Shows the coloring of the mapped reads, and makes it possible to change the coloring of selected reads types.

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 either NNN's (gap) or a IUPAC code 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 'Disconnect paired reads'. An example is shown in figure 27.23.



Figure 27.23: *Overlapping paired reads containing discrepancies in the overlapping region. The divergent area in the overlapping read pair is marked with N's and IUPAC codes in the top part of the figure. Disconnecting the paired reads by ticking the box "Disconnect Paired Reads" in the Side Panel allows you to see the individual reads.*

27.2.3 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 27.24.



Figure 27.24: *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 option 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. This section appears at the top of the **Side Panel** when viewing a stand-alone read mapping:

- **Compactness**. The compactness setting options let you control the level of detail to be displayed. This setting affects many of the other settings in the **Side Panel** as well as the general behavior of the view. For example: if the compactness is set to **Compact**, you will not be able to see quality scores or annotations on the reads, even if these are turned on via the "Nucleotide info" palette of the Side Panel. You can change the Compactness setting in the Side Panel directly, or you can use the shortcut: press and hold the Alt key while you scroll 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 this is relevant. To view such information, additional viewing options under the **Nucleotide info** view settings must also selected. For further details on these, please see section 19.1.1 and section 12.
  - **Low**. Hides trace data, quality scores and puts the reads' annotations on the sequence. You can see on figure 27.25 the Edit function that are available when right-clicking on a nucleotide.

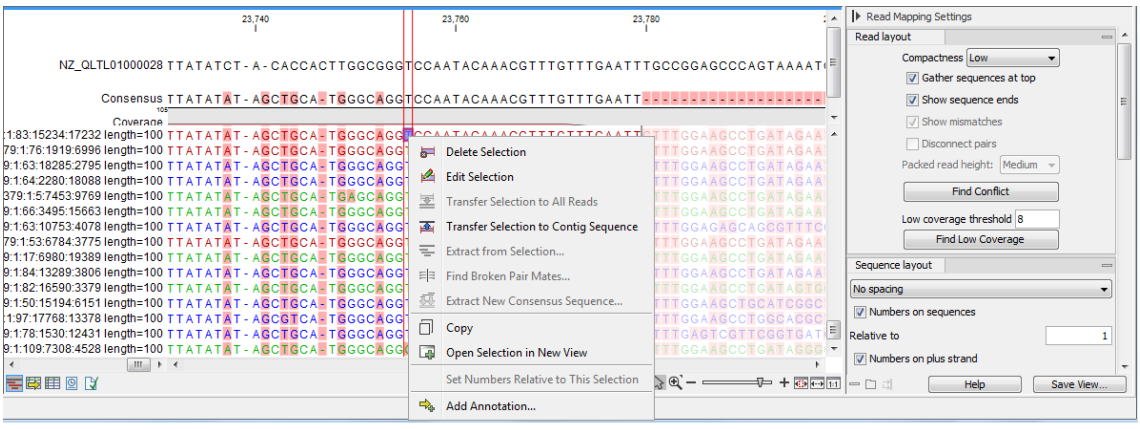


Figure 27.25: *An example of the low compactness setting.*

  - **Medium**. The labels of the reads and their annotations are hidden, and the residues of the reads cannot be seen.
  - **Compact**. Even less space between the reads.

– **Packed**. All the other compactness settings will stack the reads on top of each other, but the packed setting will use all space available for displaying the reads. When zoomed in to 100%, you can see the residues but when zoomed out the reads will be represented as lines just as with the Compact setting. The packed mode is very useful when viewing large amounts of data. However certain functionality possible with other views are not available in packed view. For example, no editing of the read mapping or selections of it can be done and color coding changes are not possible. An example of the packed setting is shown in figure 27.26.

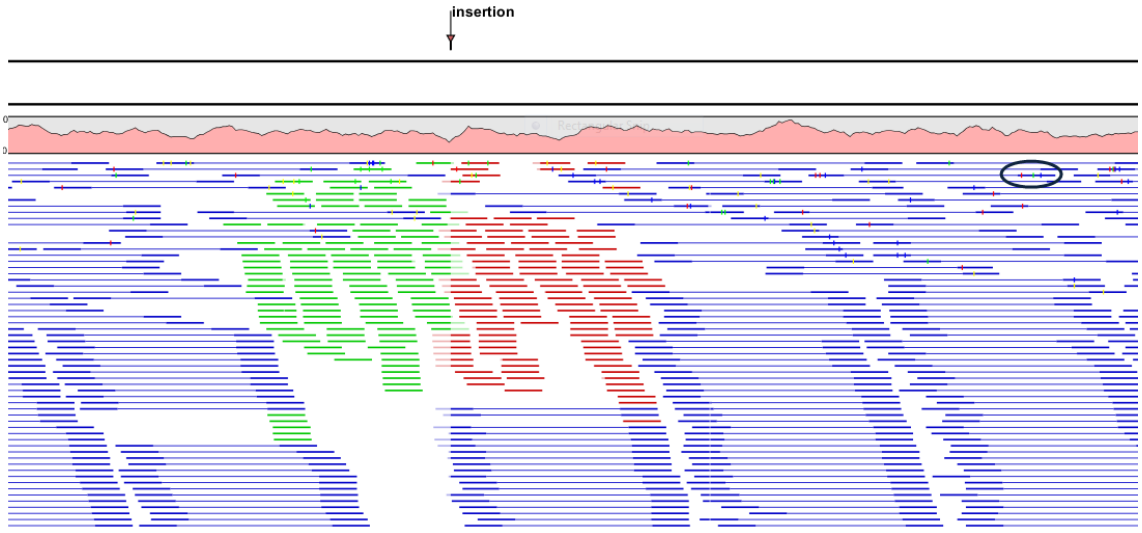


Figure 27.26: *An example of the packed compactness setting. We highlighted in black an example of 3 narrow vertical lines representing mismatching residues.*

- **Gather sequences at top**. If selected, the contributing sequence reads will automatically be placed right below the reference. This setting is not relevant when the compactness is packed.
- **Show sequence ends**. Regions that have been trimmed are shown with faded traces and residues. This illustrates that these regions have been ignored during the assembly.
- **Show mismatches**. When the compactness is packed, you can highlight mismatches which will get a color according to the Rasmol color scheme. A mismatch is whenever the base is different from the reference sequence at this position. This setting also causes the reads that have mismatches to be floated at the top of the view.
- **Disconnect paired reads**. This option will break up the paired reads in the display (they are still marked as pairs - this just affects the visualization). The reads are marked with colors for the direction (default red and green) instead of the color for pairs (default blue). This is particularly useful when investigating overlapping pairs in packed view and when the strand / read orientation is important.
- **Packed read height**. When the compactness is set to "packed", you can choose the height of the visible reads. When there are more reads than the height specified, an overflow graph will be displayed below the reads. The overflow graph is shown in the same colors as the sequences, and mismatches in reads are shown as narrow vertical lines (see figure 27.26). The colors of the small lines represent the mismatching

residue. The color codes for the horizontal lines correspond to the color used for highlighting mismatches in the sequences (red = A, blue = C, yellow = G, and green = T), meaning that a red line with half the height of the blue part of the overflow graph will represent a mismatching "A" in half of the paired reads at this particular position.

- **Find Conflict**. Clicking this button selects the next position where there is an conflict between the sequence reads. Residues that are different from the reference are colored (as default), providing an overview of the conflicts. Since the next conflict is automatically selected it is easy to make changes. You can also use the Space key to find the next conflict.
- **Low coverage threshold**. All regions with coverage up to and including this value are considered low coverage. When clicking the 'Find low coverage' button the next region in the read mapping with low coverage will be selected.

Sequence layout. There is one additional parameter to those described in section 12.1.1

- **Matching residues as dots**. Matching residues will be presented as dots. Only the top sequence will be preserved in its original format.

Residue coloring. There is one additional parameter to those described in section 12.1.1

- **Sequence colors**. This option lets you use different colors for the reads.
  - **Main**. The color of the consensus and reference sequence. Black per default.
  - **Forward**. The color of forward reads (single reads). Green per default.
  - **Reverse**. The color of reverse reads (single reads). Red per default.
  - **Paired**. The color of paired reads. Blue per default. Note that reads from **broken pairs** are colored according to their Forward/Reverse orientation or as a Non-specific match, but with a darker nuance than ordinary single reads.
  - **Non-specific matches**. When a read would have matched equally well another place in the mapping, it is considered a non-specific match. This color will "overrule" the other colors. Note that if you are mapping with several reference sequences, a read is considered a double match when it matches more than once *across all the contigs/references*. A non-specific match is yellow per default.

Alignment info. There are additional parameters to the ones described in section 21.2.

- **Coverage**: Shows how many sequence reads that are contributing information to a given position in the read mapping. The level of coverage is relative to the overall number of sequence reads.
- **Paired distance**: Plots the distance between the pairs 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 as follow:

- **Foreground color**. Colors the letters 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 letters using a gradient, where the left side color is used for low coverage and the right side is used for maximum coverage.
- **Graph**. The coverage is displayed as a graph (Learn how to export the data behind the graph in section 6.8).
  - **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.

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 27.27.

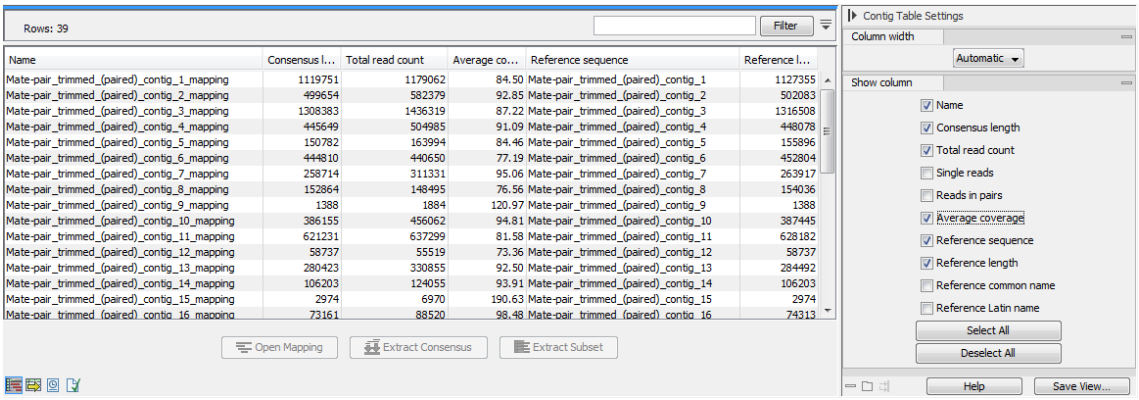


Figure 27.27: *The mapping table.*

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 27.6).
- **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 27.28 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 6.3.7), or if one of the reads do not map at all.

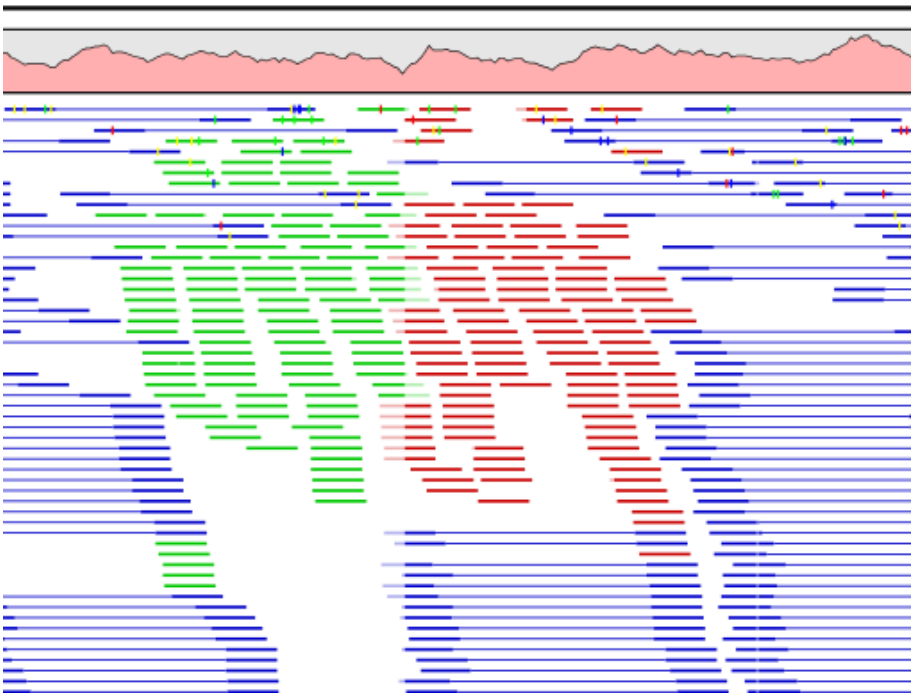


Figure 27.28: *Broken pairs.*

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 27.29.

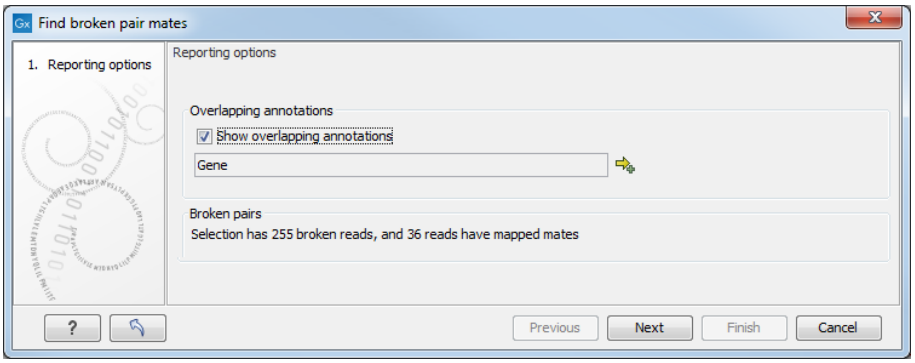


Figure 27.29: *Finding the mates of broken pairs.*

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 27.30.

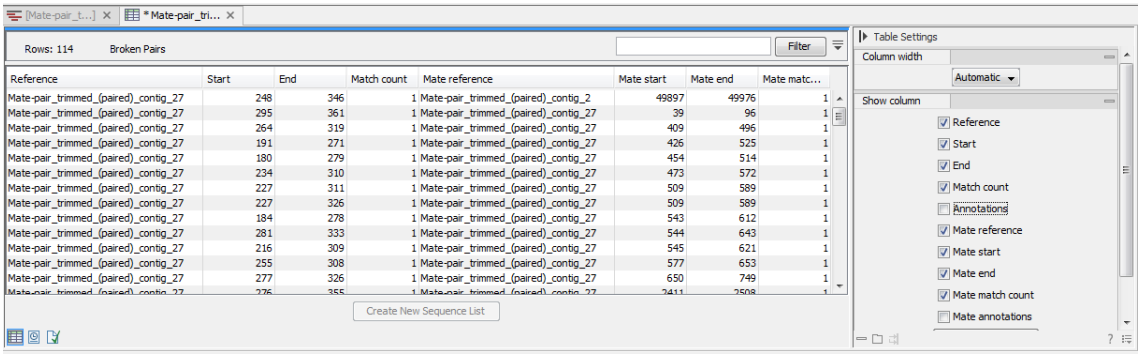


Figure 27.30: *An overview of the broken pairs.*

The table includes the following information for both parts of the pair:

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 27.29.

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.

27.3 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 27.31. An example in the case of a mapping made using the global alignment algorithm is shown in figure 27.32.

27.3.1 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-Brujn 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 realignment of each individual read. Figure 27.33 depicts a partial realignment graph for the read mapping shown in figure 27.31.

27.3.2 Realignment of unaligned ends

A typical error in read alignments is the occurrence of unaligned ends (also known as soft-clipped read ends). These unaligned ends are introduced by the read mapper as a consequence of an unresolved indel towards the end of a read. Those unaligned ends can be realigned in many cases, after the read itself has been locally realigned according to the indel that prevented the read mapper from aligning the read ends correctly. Figure 27.34 depicts such an example.

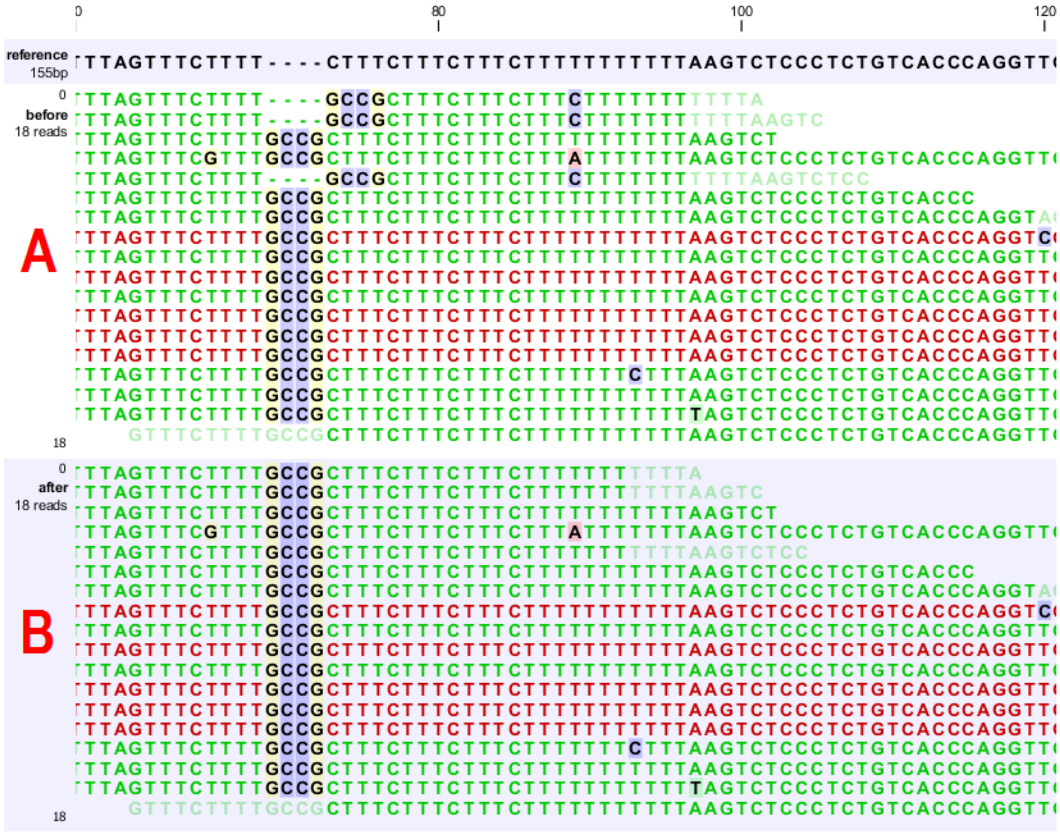


Figure 27.31: *Local realignment of a read mapping produced with the 'local' option. [A] The alignments of the first, second, and fifth read in this read mapping do not support the four-nucleotide insertion supported by the remaining reads. A variant detection tool might be tempted to call a heterozygous insertion of four nucleotides in one allele and heterozygous replacement of four nucleotides in a second allele. [B] After applying local realignment, the first, second, and fifth read consistently support the four-nucleotide insertion.*

27.3.3 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:

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. 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

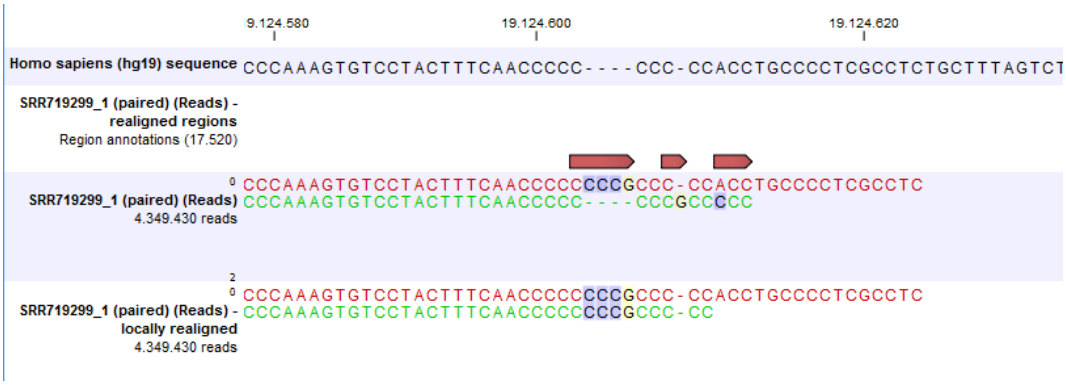


Figure 27.32: *Local realignment of a read mapping produced with the 'global' option. Before realignment the green read was mapped with two mismatches. After realignment it is mapped with the inserted 'CCCG' sequence (seen in the alignment of the red read) and no mismatches.*

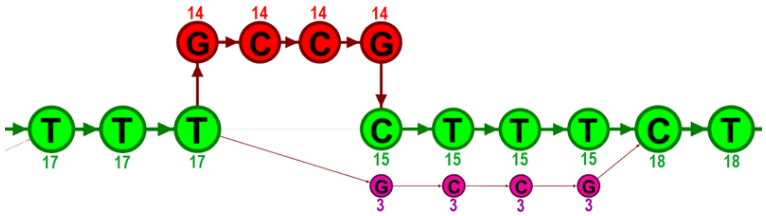


Figure 27.33: *The green nodes represent nucleotides of the reference sequence. The four red nodes represent the four-nucleotide insertion observed in fourteen mapped reads. The four violet nodes represent the four mismatches to the reference sequence observed in three mapped reads. During realignment 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.*

in the data (e.g., the 'InDel' track from the Structural Variants' tool (see Section 28.10) produced on the 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 27.35 and figure 27.36 show examples that can be improved by guiding the local realignment algorithm.

27.3.4 Multi-pass local realignment

As described in section 27.3.1 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.

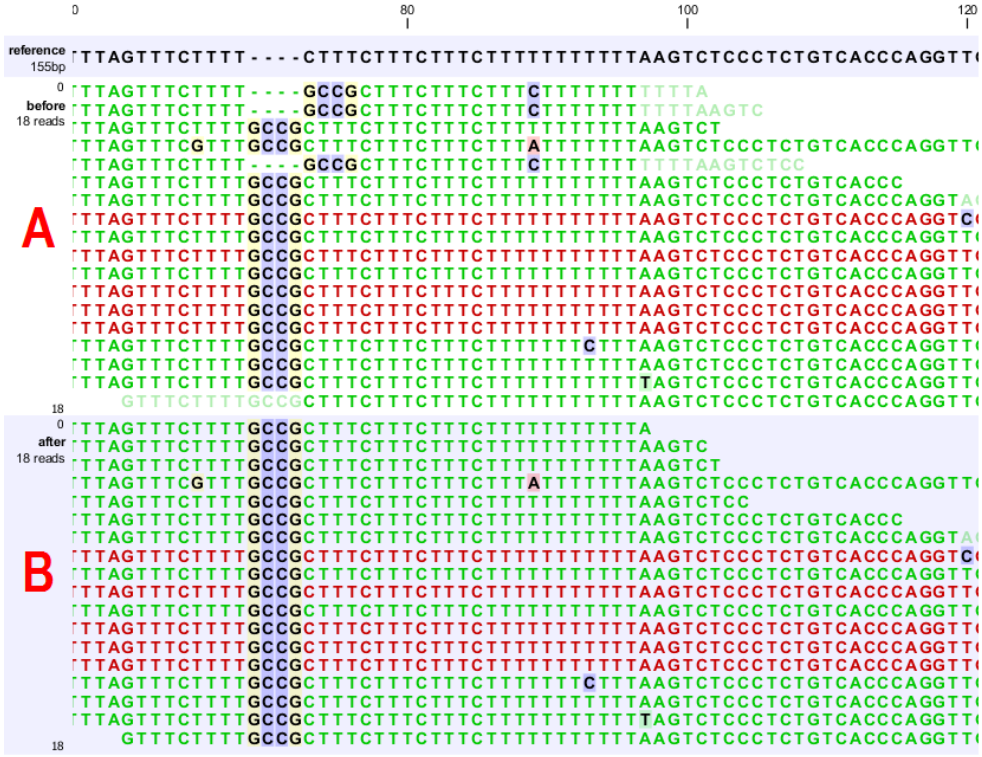


Figure 27.34: *[A] The alignments of the first, second, and fifth read in this read mapping do not support the four-nucleotide insertion supported by the remaining reads. Additionally, the first, second, fifth and the last reads have unaligned ends. [B] After applying local realignment the first, second 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 27.31).*

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 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.

27.3.5 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.



Figure 27.35: *[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.*

- 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".
- 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.

27.3.6 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.



Figure 27.36: *[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).*

27.3.7 Run the Local Realignment tool

The tool is found in the Toolbox:

Toolbox | **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 27.37.

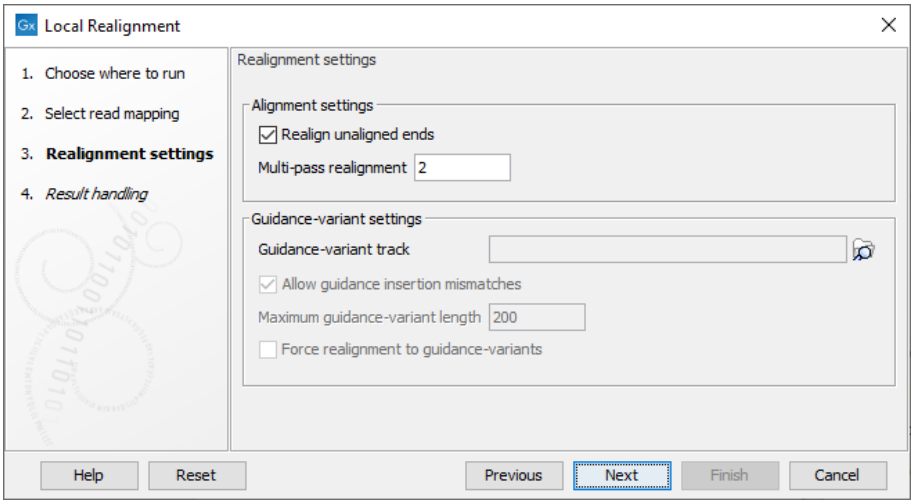


Figure 27.37: *Set the realignment options.*

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: each guidance variant will be represented by a pseudo-read, allowing the local realignment to explore the alignments that the guidance variants suggest. 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 28.10), 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 27.38).

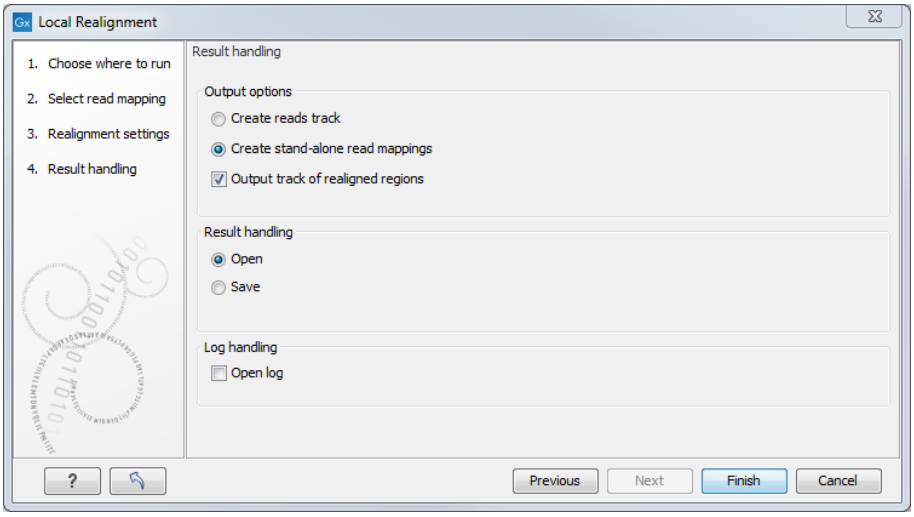


Figure 27.38: *An output track of realigned regions can be created.*

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.

27.4 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:

Toolbox | **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 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 27.1). 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.

27.5 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 27.39.

When sequencing library preparation involves a PCR amplification step, it is common to observe multiple reads where identical nucleotide sequences are disproportionably 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 disproportionally 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

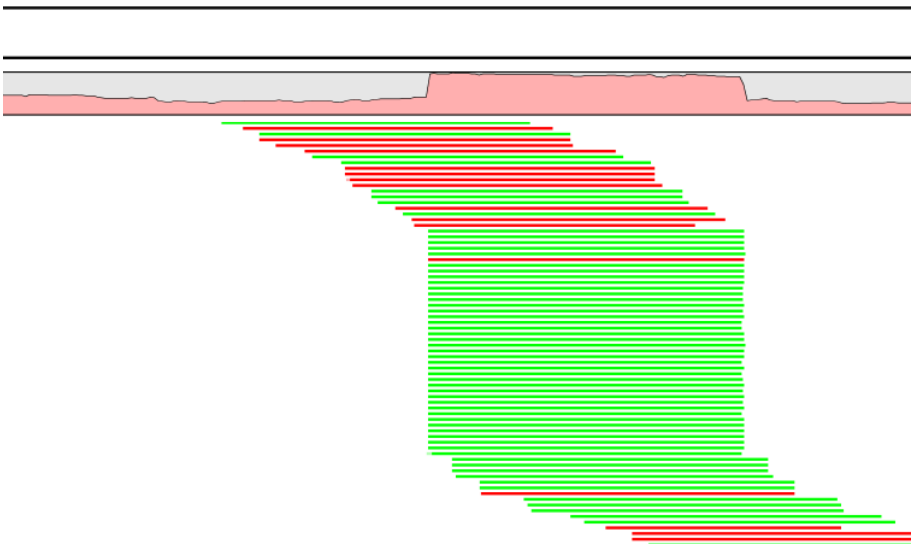


Figure 27.39: *Mapped reads with a set of duplicate reads, the colors denote the strand (green is forward and red is reverse).*

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 reccomended 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 and end 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.

27.5.1 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).

Based on this assumption, a group of reads that share identical start and end coordinates (or start coordinate and length for single end reads) and also share identical sequences can be considered as potential duplications of the same DNA fragment. These reads are then investigated to find reads to be removed and reads to be kept. In order to explain how this works, we will use a small example shown in figure 27.40.

The example shows 165 reads that share the same start position and orientation and are considered for duplicate read removal. 60 reads share the sequence shown at the top, 5 reads

```
ACGGACTGCTT   60
ACTGACTGCTT    5
ACTGACTGATT  100
```

Figure 27.40: *An alignment of three different sequence. The numbers are read counts, e.g. the read at the top occurs 60 times.*

share the middle sequence, and 100 reads share the sequence at the bottom. The differences are at position 3 and 9 (underlined).

The tool will now create a tree structure out of these reads as illustrated in figure 27.41.

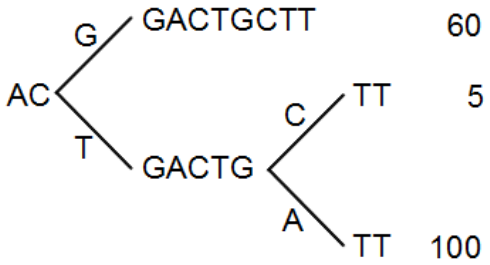


Figure 27.41: *The reads from figure 27.40 represented as a Patricia tree [Morrison, 1968].*

The first branch point in the tree is at the third position, where sequence number one has a G and the other sequences have a T. The other two sequence disagree at position nine, where one has a C and another has an A.

The next step is to iteratively merge the branches, starting from the end of the tree. The first branch point to consider is at position nine. Since only 5 reads have a C and 100 reads have an A, the C branch is collapsed. This is shown in figure 27.42.

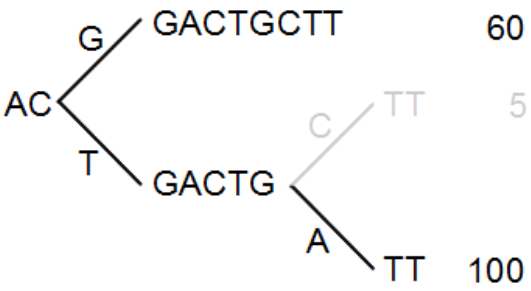


Figure 27.42: *Merging the sequences.*

As a user, you can specify the **threshold** for when the reads should be merged. The default is 20 %: when the minority branch has less than 20 % of the read count of the both branches, it is collapsed.

The next branch to consider is at the third position, where there are now 105 reads that have a T and 60 reads that have a G (see figure 27.43).

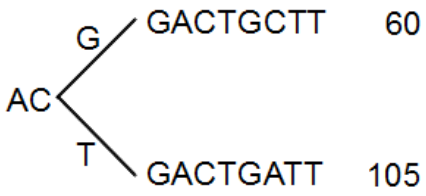


Figure 27.43: *Merging the sequences.*

With the default setting at 20%, these two branches will not be collapsed, because there are too many reads on the minority branch (60 reads versus 105 reads). Since this process is aimed at collapsing reads that are only distinguished apart by sequencing errors, you would not expect this situation to be caused by sequencing errors, but rather true biological variation (or PCR errors in the early cycles that are indistinguishable from true variation).

If we raised the threshold to 60%, the two branches above would be merged into one if it was not for the second rule governing the merging of branches: The sequences have to be identical except for the difference at the branch point. Looking at the sequences in figure 27.43, there is a difference at position 9 which means that these two branches would never be merged, regardless the threshold and the read counts.

The result of the duplicate reads removal in this example would be that the 165 reads are reduced to two in the result.

27.5.2 Running the duplicate reads removal

The tool is found in the Toolbox:

Toolbox | **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 27.44.

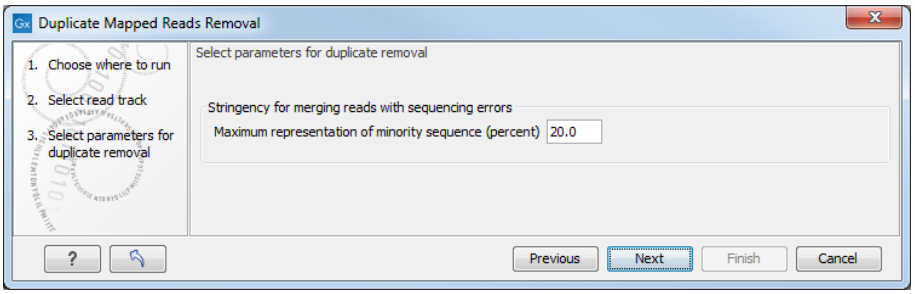


Figure 27.44: *Setting the stringency for merging similar reads.*

The parameter is explained in detail in section 27.5.1.

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 27.45).

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.

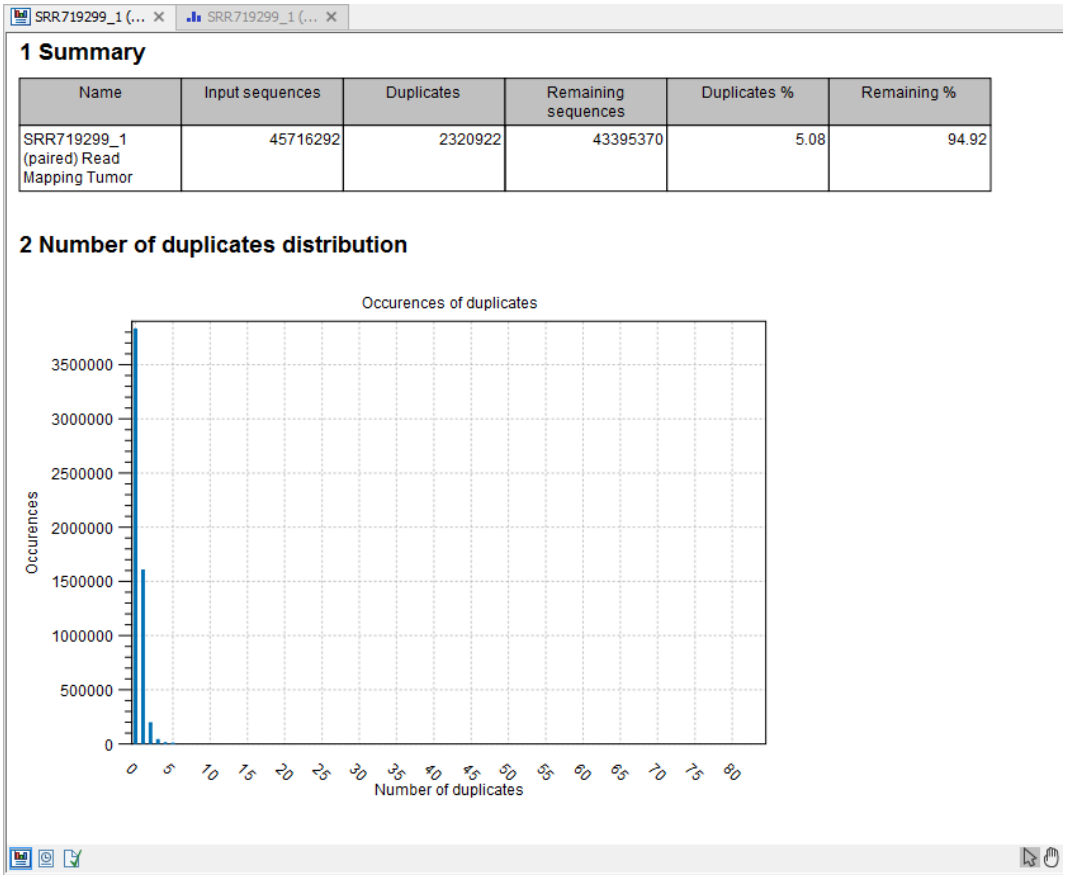


Figure 27.45: *Summary statistics on the duplicate mapped reads.*

27.6 Extract Consensus Sequence

Using the **Extract Consensus Sequence** tool, a consensus sequence can be extracted from all kinds of read mappings, including those generated from *de novo* assembly or RNA-seq analyses. In addition, you can extract a consensus sequence from nucleotide BLAST results.

Note: Consensus sequences can also be extracted when viewing a read mapping by right-clicking on the name of the consensus or reference sequence, or a selection of the reference sequence, and selecting the option **Extract New Consensus Sequence (** **)** from the menu that appears. The same option is available from the graphical view of BLAST results when right-clicking on a selection of the subject sequence.

To start the **Extract Consensus Sequence** tool, go to:

Toolbox | **Resequencing Analysis** ( ) | **Extract Consensus Sequence (** **)**

In the first step, select the read mappings or nucleotide BLAST results to work with.

In the next step, options affecting how the consensus sequence is determined are configured (see figure 27.46).

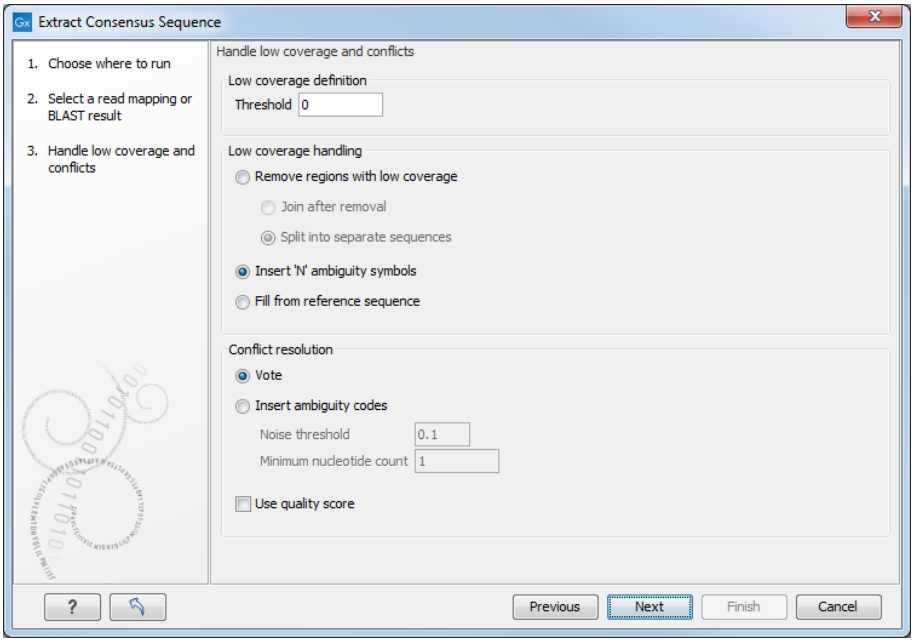


Figure 27.46: *Specifying how the consensus sequence should be extracted.*

Handling low coverage regions

The first step is to define a **low coverage threshold**. Consensus sequence is not generated for reference positions with coverage at or below the threshold specified.

The default value is 0, which means that a reference base is considered to have low coverage when no reads cover this position. Using this threshold, if just a single read covered a particular position, only that read would contribute to the consensus at that position. Setting a higher threshold gives more confidence in the consensus sequence produced.

There are several options for how low coverage regions should be handled:

- **Remove regions with low coverage**. When using this option, no consensus sequence is created for the low coverage regions. There are two ways of creating the consensus sequence from the remaining contiguous stretches of high coverage: either the consensus sequence is **split** into separate sequences when there is a low coverage region, or the low coverage region is simply ignored, and the high-coverage regions are directly **joined**. In this case, an annotation is added at the position where a low coverage region is removed in the consensus sequence produced (see below).
- **Insert 'N' ambiguity symbols**. This simply adds Ns for each base in the low coverage region. An annotation is added for the low coverage region in the consensus sequence produced (see below).
- **Fill from reference sequence**. This option uses the sequence from the reference to construct the consensus sequence for low coverage regions. An annotation is added for the low coverage region in the consensus sequence produced (see below).

Handling conflicts

Settings are provided in the lower part of the wizard for configuring how conflicts or disagreement

between the reads should be handled when building a consensus sequence in regions with adequate coverage.

- **Vote** When reads disagree at a given position, the base present in the majority of the reads at that position is used for the consensus.
  - If the most common symbol is a gap, then the consensus symbol is a gap.
  - If there are equal numbers of gaps and non-gaps, then it is one of the other symbols.
  - When choosing between symbols, we choose in the order A - C - G - T.
  - Ambiguous symbols cannot be chosen.

  If the **Use quality score** option is also selected, quality scores are used to decide the base to use for the consensus sequence, rather than the number of reads. The quality scores for each base at a given position in the mapping are summed, and the base with the highest total quality score at a given position is used in the consensus. If two bases have the same total quality score at a location, we follow the order of preference listed above.

  Information about biological heterozygous variation in the data is lost when the **Vote** option is used. For example, in a diploid genome, if two different alleles are present in an almost even number of reads, only one will be represented in the consensus sequence.

- **Insert ambiguity codes** When reads disagree at a given position, an ambiguity code representing the bases at that position is used in the consensus. (The IUPAC ambiguity codes used can be found in Appendix H and G.)

  Unlike the Vote option, some level of information about biological heterozygous variation in the data is retained using this option.

  To avoid the situation where a different base in a single read could lead to an ambiguity code in the consensus sequence, the following options can be configured:

  - **Noise threshold** The percentage of reads where a base must be present at given position for that base to contribute to an ambiguity code. The default value is 0.1, i.e. for a base to contribute to an ambiguity code, it must be present in at least 10 % of the reads at that position.
  - **Minimum nucleotide count** The minimum number of reads a particular base must be present in, at a given position, for that base to contribute to the consensus.

  If no nucleotide passes these two thresholds at a given position, that position is omitted from the consensus sequence.

  If the **Use quality score** option is also selected, summed quality scores are used, instead of numbers of reads for conflict handling. To contribute to an ambiguity code, the summed quality scores for bases at a given position must pass the noise threshold.

In the next step, output options are configured (figure 27.47).

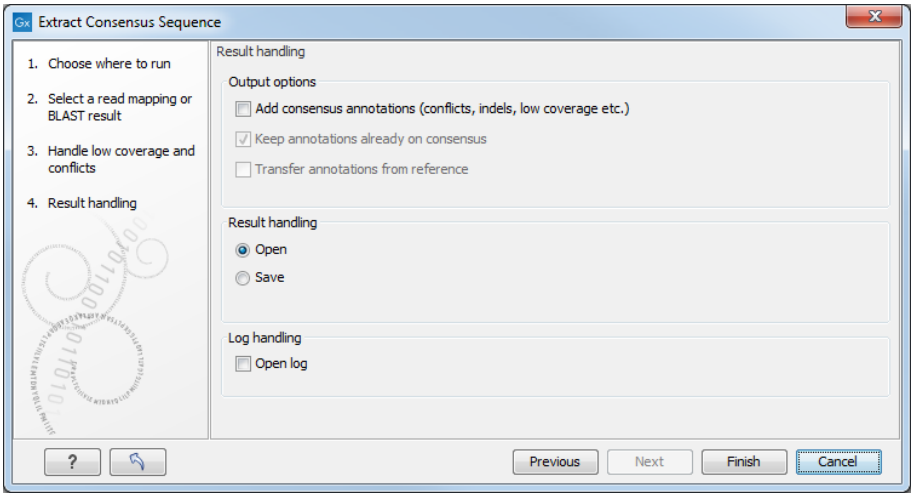


Figure 27.47: *Choose to add annotations to the consensus sequence.*

Consensus annotations

The annotations that can be added to the consensus sequence provide information about conflicts that have been resolved and about low coverage regions (unless you have chosen to split the consensus sequence). Note that for large data sets, a very high number of annotations may be generated, which will cause the tool to take longer to complete. The results will also take up much more disk space.

For stand-alone read mappings, it is possible to transfer existing annotations to the consensus sequence. Since the consensus sequence produced may be broken up, the annotations will also be broken up, and thus may not have the same length as before. In some cases, gaps and low-coverage regions will lead to differences in the sequence coordinates between the input data and the new consensus sequence. The annotations copied will be placed in the region on the consensus that corresponds to the region on the input data, but the actual coordinates might have changed.

Track-based read mappings do not themselves contain annotations and thus the options related to transferring annotations, "Transfer annotations from the reference sequence" and "Keep annotations already on consensus", cannot be selected for this type of input.

Copied/transferred annotations will contain the same qualifier text as the original. That is, 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, not the new sequence, which may differ.

Quality scores on the consensus sequence

The resulting consensus sequence (or sequences) will have quality scores assigned if quality scores were found in the reads used to call the consensus. For a given consensus symbol X we compute its quality score from the "column" in the read mapping. Let Y be the sum of all quality scores corresponding to the "column" below X, and let Z be the sum of all quality scores from that column that supported X[1]. Let $Q = Z - (Y - Z)$, then we will assign X the quality score of q where

[1]By supporting a consensus symbol, we understand the following: when conflicts are resolved using voting, then only the reads having the symbol that is eventually called are said to support the consensus. When ambiguity codes

$$q = \begin{cases} 64 & \text{if } Q > 64 \\ 0 & \text{if } Q < 0 \\ Q & \text{otherwise} \end{cases}$$

are used instead, all reads contribute to the called consensus and thus $Y = Z$.

Chapter 28

Variant detection

Contents

28.1 Variant Detection tools 698
28.1.1 Differences in the variants called by the different tools 699
28.1.2 How the variant detection tools work 702
28.1.3 Detailed information about overlapping paired reads 702
28.2 Fixed Ploidy Variant Detection 703
28.3 Low Frequency Variant Detection 705
28.4 Basic Variant Detection 705
28.5 Variant Detection - filters 706
28.5.1 General filters 706
28.5.2 Noise filters 708
28.6 Variant Detection - the outputs 714
28.6.1 Variant tracks 714
28.6.2 The annotated variant table 721
28.6.3 The variant detection report 722
28.7 Fixed Ploidy and Low Frequency Detection tools: detailed descriptions . . . 723
28.7.1 Variant Detection - error model estimation 723
28.7.2 The Fixed Ploidy Variant Detection tool: Models and methods 724
28.7.3 The Low Frequency Variant Detection tool: Models and methods 728
28.8 Copy Number Variant Detection 731
28.8.1 The Copy Number Variant Detection tool 732
28.8.2 Region-level CNV track (Region CNVs) 737
28.8.3 Target-level CNV track (Target CNVs) 739
28.8.4 Gene-level annotation track (Gene CNVs) 742
28.8.5 CNV results report 742
28.8.6 CNV algorithm report 743
28.9 Identify Known Mutations from Sample Mappings 746
28.9.1 Run the Identify Known Mutations from Sample Mappings tool 747
28.9.2 Output from the Identify Known Mutations from Sample Mappings tool . 749
28.10 InDels and Structural Variants 751
28.10.1 Run the InDels and Structural Variants tool 751

28.10.2 The Structural Variants and InDels output 754
28.10.3 The InDels and Structural Variants detection algorithm 758
28.10.4 Theoretically expected structural variant signatures 761
28.10.5 How sequence complexity is calculated 765

28.1 Variant Detection tools

CLC Genomics Workbench offers three tools for detecting variants.

- **Fixed Ploidy Variant Detection** () - described in detail in section 28.2
- **Low Frequency Variant Detection** () - described in detail in section 28.3
- **Basic Variant Detection** () - described in detail in section 28.4

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 28.1.

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 28.1: *An overview of the variant detection tools.*

To run one of the variant detection tool, go to:

Toolbox | **Resequencing Analysis (**  **)**

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

28.1.1 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 28.5). 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 Frequencey 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 28.2 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.

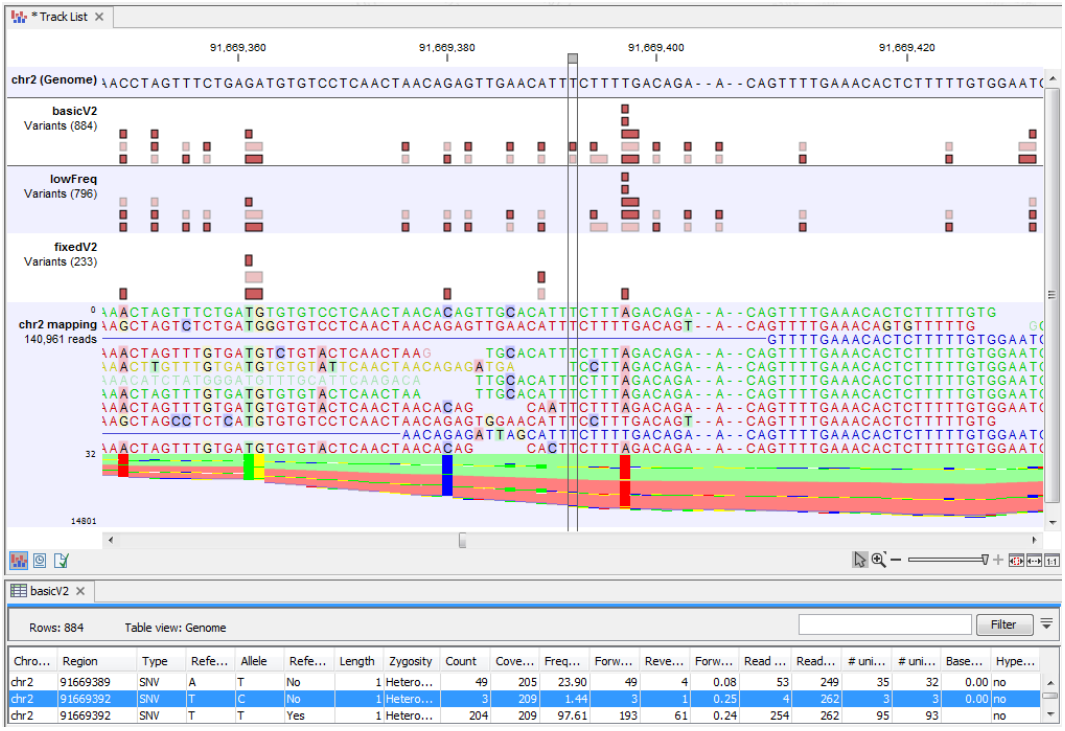


Figure 28.2: *Case where a variant is detected only using the Basic Variant Detection tool.*

Figure 28.3 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 28.3, 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 28.3: *Case where variants can be detected as SNV by a tool and MNV by another.*

Figure 28.4 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.

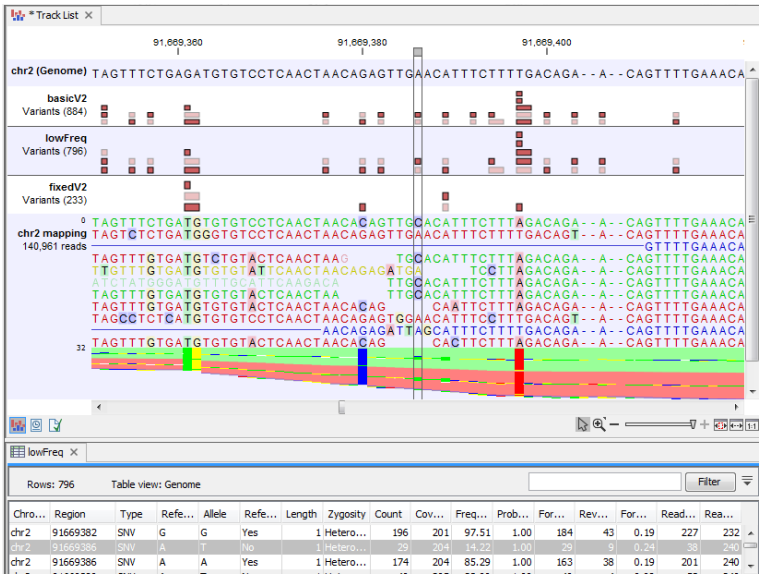


Figure 28.4: *Case where a variant does not fit the ploidy assumption.*

28.1.2 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 polymorphims 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 28.5 and section 28.6.

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 28.2 and 28.3.
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 28.5).
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.

28.1.3 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.

28.2 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 28.7.1).

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 28.5):

- 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 28.2.
- 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.

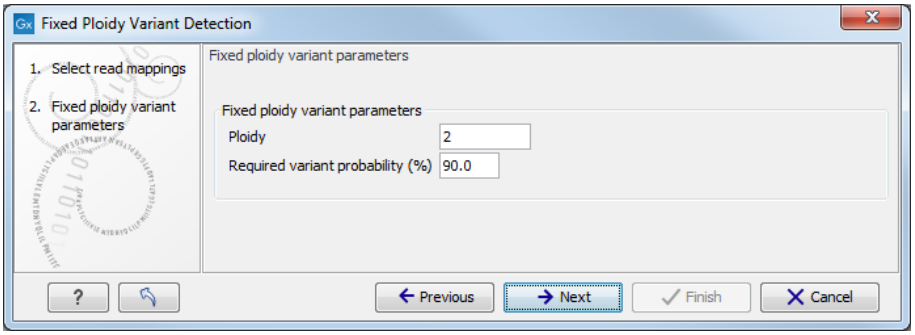


Figure 28.5: *The Fixed Ploidy Variant Detection parameters.*

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.

28.3 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 28.6):

- **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).

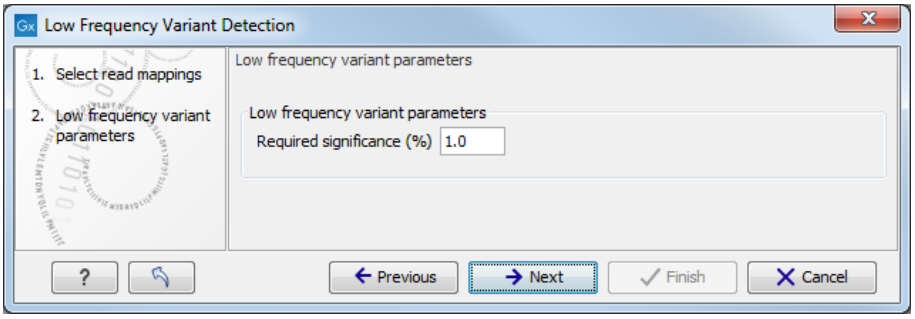


Figure 28.6: *The Low Frequency Variant Detection parameters.*

For a more in depth description of the Low Frequency Variant Detection tool see section 28.7).

28.4 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 28.5). The tool has a single parameter (figure 28.7) 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 28.6 for a description of the outputs).

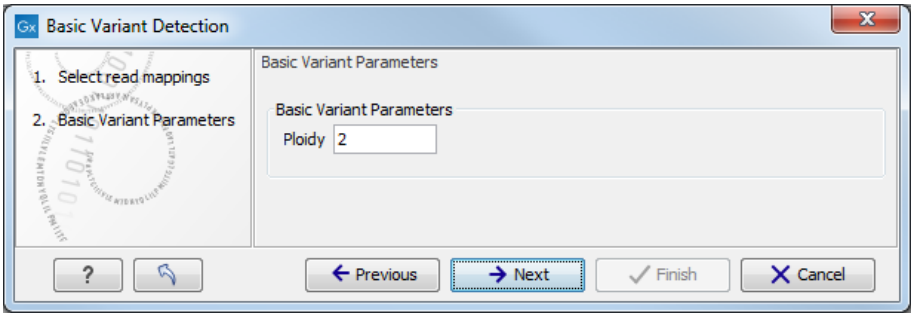


Figure 28.7: *The Basic Variant Detection parameters.*

28.5 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 28.6). 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.

28.5.1 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 28.8):

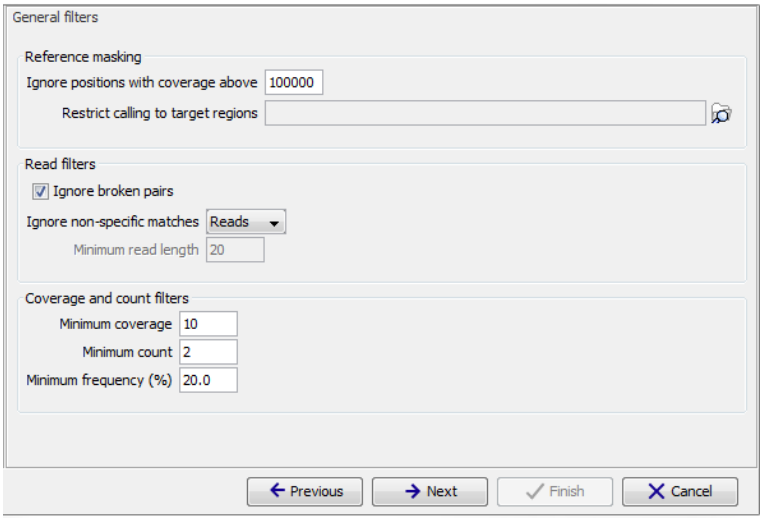


Figure 28.8: *General filters. The values shown are those that are default for Fixed Ploidy Variant detection.*

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.

28.5.2 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 28.9). 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.

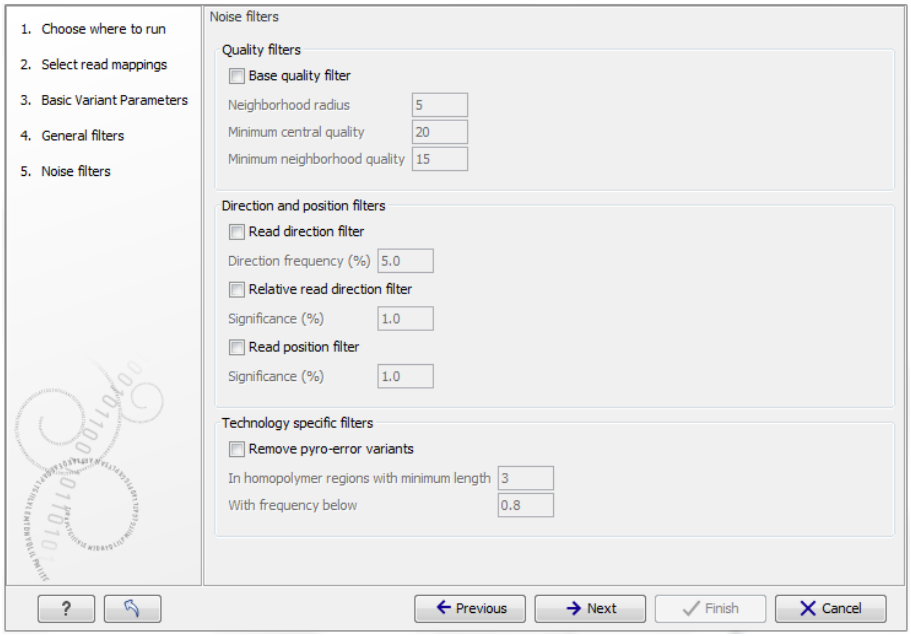


Figure 28.9: *Noise filters.*

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 28.10 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.

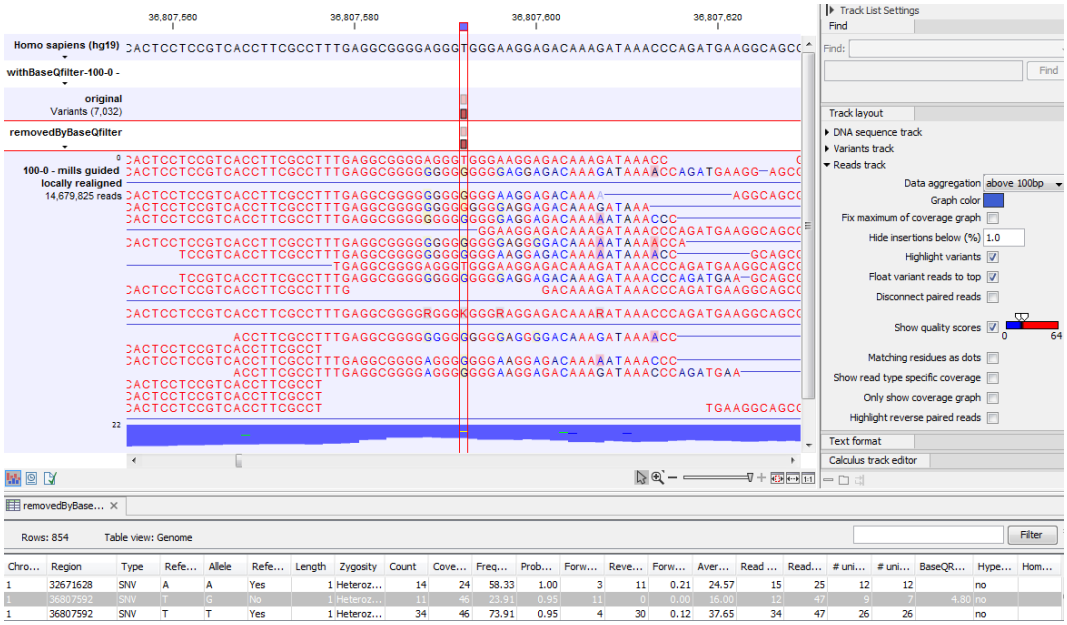


Figure 28.10: *Example of a variant called when the base quality filter is NOT applied, and not called when it is.*

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 28.11 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 'Disconnect 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.

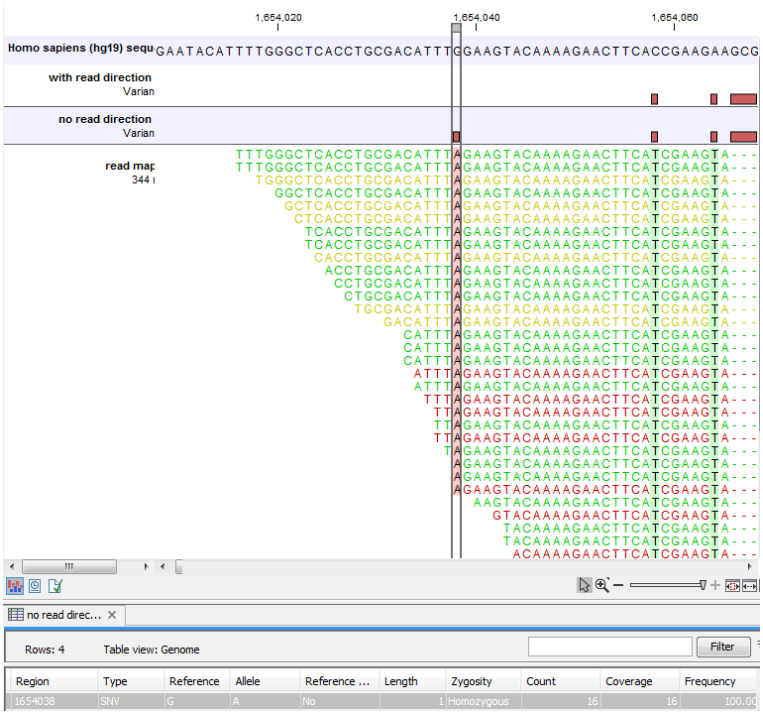


Figure 28.11: *Example of a variant that is removed by the 'Read direction' filter.*

Figure 28.12 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).

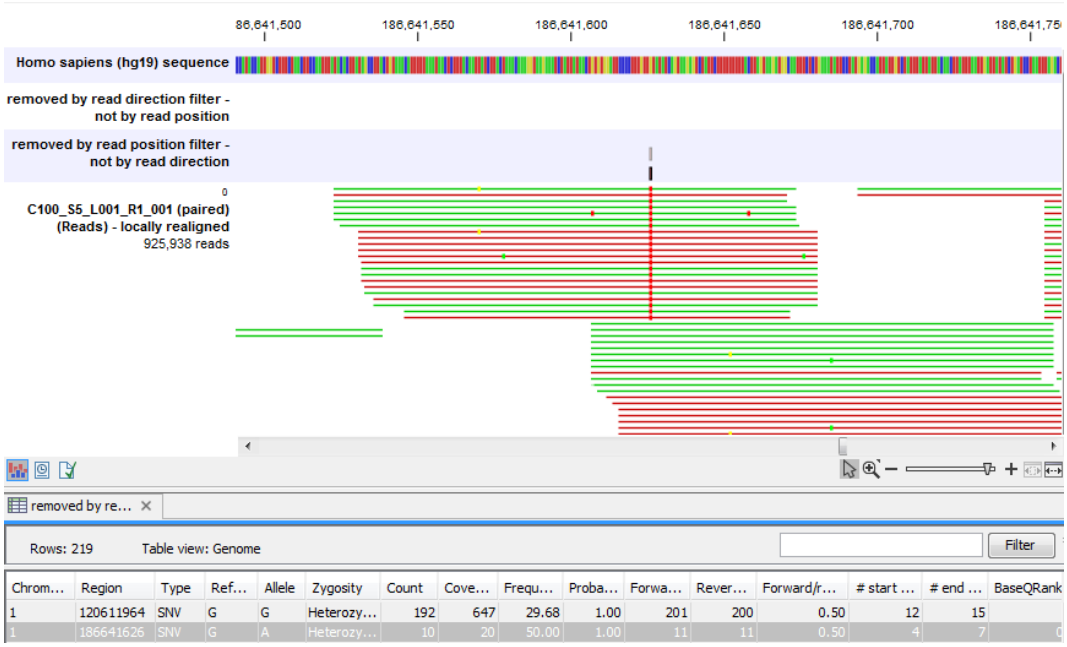


Figure 28.12: *A variant that is filtered out by the Read position filter but not by the Read direction filter.*

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 "AAAAAAAA**A**". 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 "CGAAAAAGT**A**CG". 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 28.13 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.

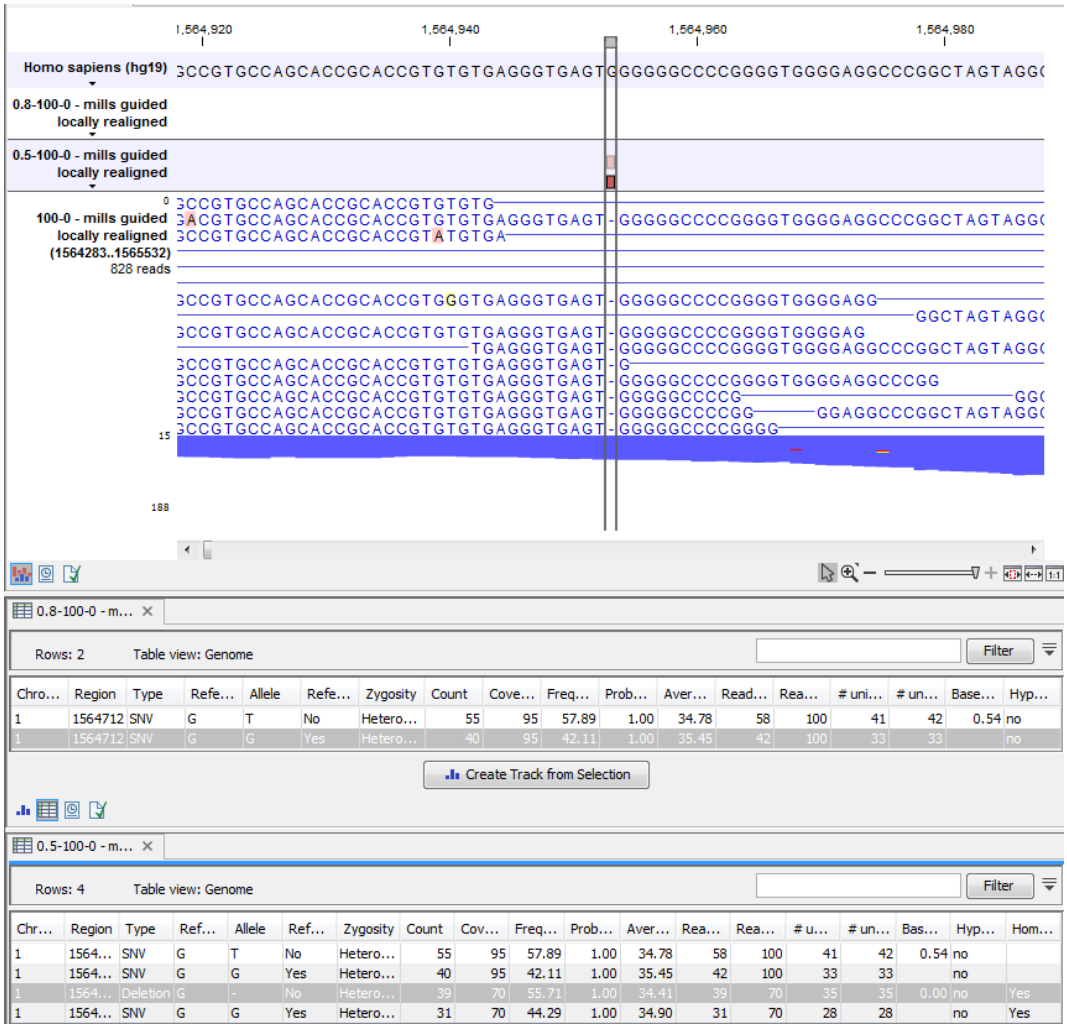


Figure 28.13: *An example of a variant that is filtered out when the pyro-error filter is applied with settings 3 and 0.8, but not with settings 3 and 0.5.*

In addition to the example above, a simple example is provided below in figure 28.14 to illustrate the difference between variant frequency and pyro-variant removal frequency (where non-reference and non-homopolymer variant reads are ignored).

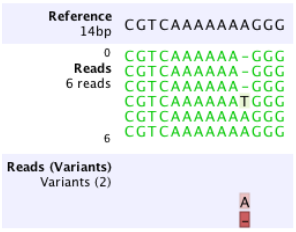


Figure 28.14: *An example of a simple read mapping with 6 mapped reads. Three of them indicate a deletion, two match the reference, and one read is an A to T SNP*

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.

28.6 Variant Detection - the outputs

The Variant Detection Tools have the following outputs: a variant track, an annotated variant table and a report (figure 28.15). 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 6.2), or downloaded from external databases (e.g. dbSNP, HapMap, or 1000genomes) - as described in section 6.2)).

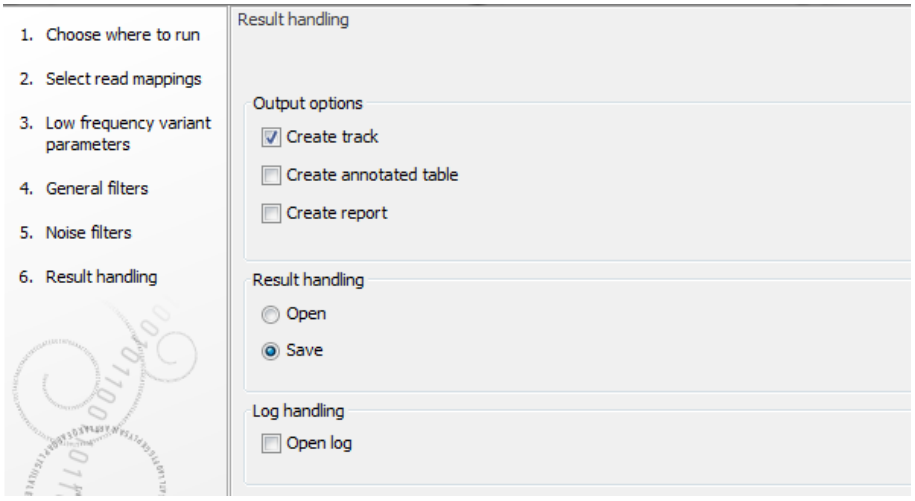
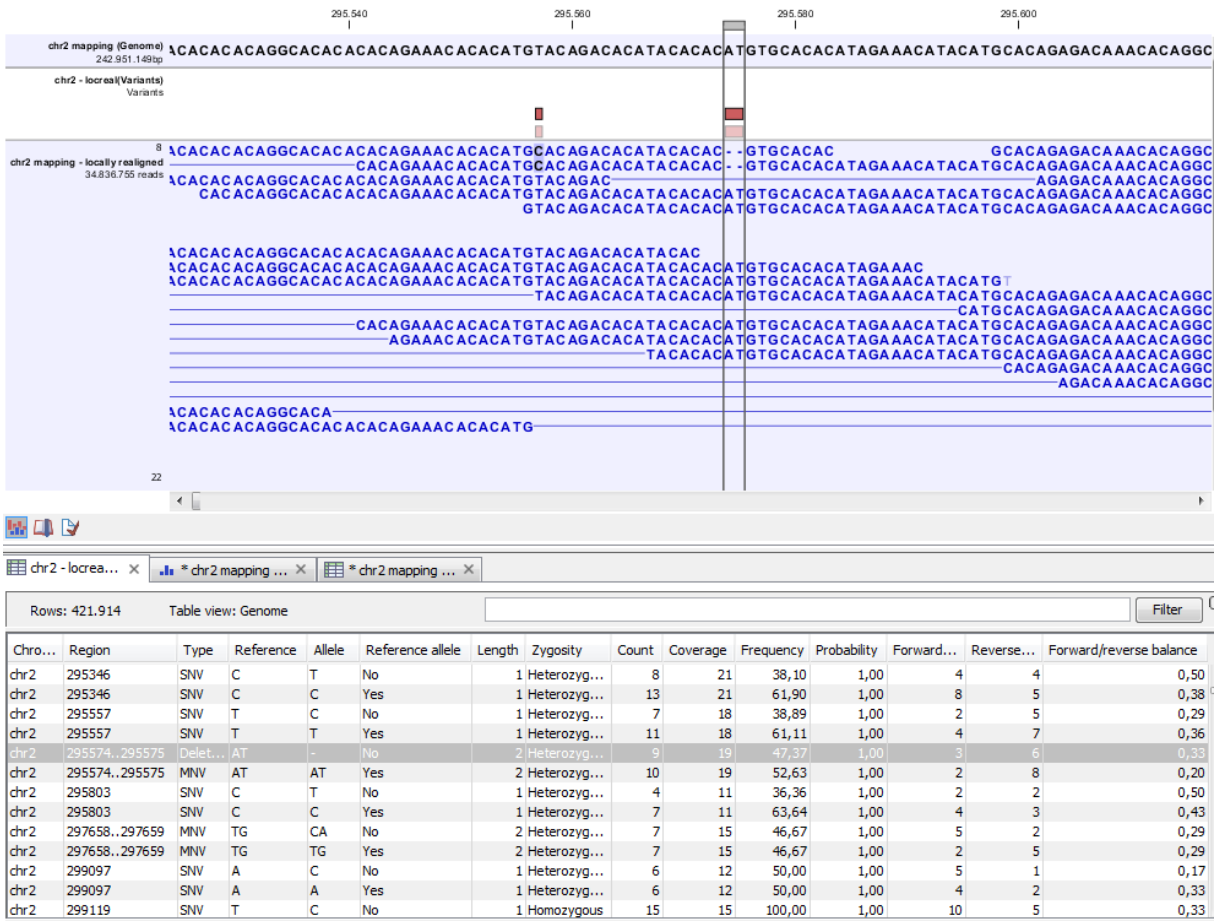


Figure 28.15: *Output options.*

28.6.1 Variant tracks

A variant track (figure 28.16) usually contains the following information for each variant:



Chro...	Region	Type	Reference	Allele	Reference allele	Length	Zygosity	Count	Coverage	Frequency	Probability	Forward...	Reverse...	Forward/reverse balance
chr2	295346	SNV	C	T	No	1	Heterozyg...	8	21	38,10	1,00	4	4	0,50
chr2	295346	SNV	C	C	Yes	1	Heterozyg...	13	21	61,90	1,00	8	5	0,38
chr2	295557	SNV	T	C	No	1	Heterozyg...	7	18	38,89	1,00	2	5	0,29
chr2	295557	SNV	T	T	Yes	1	Heterozyg...	11	18	61,11	1,00	4	7	0,36
chr2	295574..295575	Delet...	AT	-	No	2	Heterozyg...	9	19	47,37	1,00	3	6	0,33
chr2	295574..295575	MNV	AT	AT	Yes	2	Heterozyg...	10	19	52,63	1,00	2	8	0,20
chr2	295803	SNV	C	T	No	1	Heterozyg...	4	11	36,36	1,00	2	2	0,50
chr2	295803	SNV	C	C	Yes	1	Heterozyg...	7	11	63,64	1,00	4	3	0,43
chr2	297658..297659	MNV	TG	CA	No	2	Heterozyg...	7	15	46,67	1,00	5	2	0,29
chr2	297658..297659	MNV	TG	TG	Yes	2	Heterozyg...	7	15	46,67	1,00	2	5	0,29
chr2	299097	SNV	A	C	No	1	Heterozyg...	6	12	50,00	1,00	5	1	0,17
chr2	299097	SNV	A	A	Yes	1	Heterozyg...	6	12	50,00	1,00	4	2	0,33
chr2	299119	SNV	T	C	No	1	Homozygous	15	15	100,00	1,00	10	5	0,33

Figure 28.16: *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 was produced by running the Fixed Ploidy Variant Detection tool on the reads track. 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 28.17.

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.

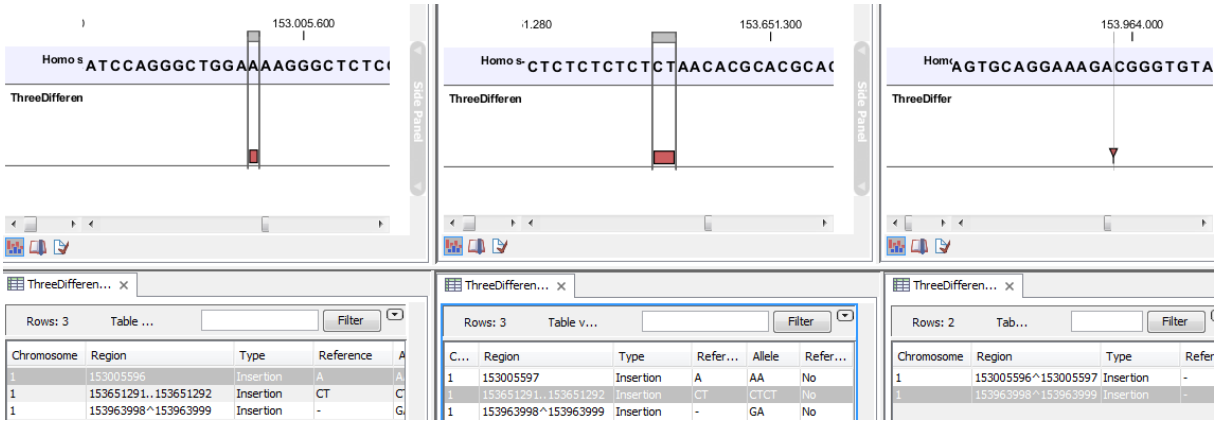


Figure 28.17: *Examples of variants with different types of 'Region' column contents. The left-most variant has a 'single position' region, the middle variant has a 'region' region and the right-most has a 'between positions' region.*

- 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 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".

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.

Linkage

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 28.1.3 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 29.1.2 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 28.1.3.

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 28.1.3.)

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 29.1.2).

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 6.3).

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 'Disconnect 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 'Disconnect 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 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.

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.

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 is 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 99.999999999999999999%

Please note that the variants in the variant track can be enriched with information using the annotation tools in section 29.2.

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 28.17 is given in figure 28.18.

```
##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 28.18: *A gvf file giving rise to the variants in the figure above.*

28.6.2 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 28.19).

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 28.19: *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 `http://varnomen.hgvs.org/`.

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 `http://varnomen.hgvs.org/`.

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 29.2).

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 2.1.5), 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 28.6.1).

28.6.3 The variant detection report

In addition to the estimated error rates of the different types of errors shown in figure 28.21, 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 28.20).

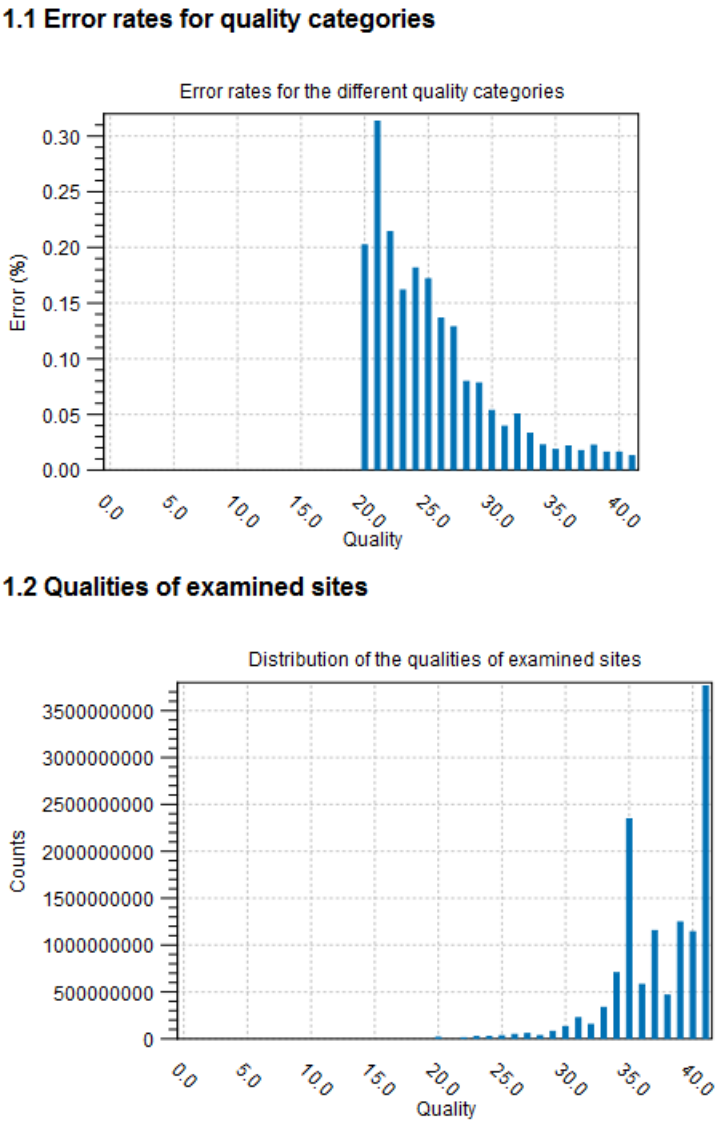


Figure 28.20: *Part of the contents of the report on the variant calling.*

28.7 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 28.2 and 28.3.

28.7.1 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. Information on the estimated error rates can be found in the Reports (see section 28.6 and figure 28.21).

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 28.21: *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.

28.7.2 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 **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:

$$\{e(N \to M)|N, M \in \{A, C, G, T, -\}\}.$$

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:

$$\begin{aligned} P(t|data) &= \frac{P(data|t)P(t)}{P(data)} \\ &= \frac{P(data|t)P(t)}{\sum_{s \in S} P(data|s)P(s)}, \end{aligned} \tag{28.1}$$

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:

$$P(data|t) = P(n_1, ..., n_k|t).$$

To derive the likelihood of the data, $P(n_1, ..., n_k|t)$, 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 $n_1, ..., n_k$ in a site t as:

$$P(n_1, ..., n_k|t) = \prod_{i=1}^{k} \sum_{N \in \{A,C,G,T,-\}} P_t(N) \times e_q(N \to n_i). \tag{28.2}$$

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 (28.1), we thus have the following equation for the posterior probabilities of the site types:

$$\begin{aligned} P(t|n_1, ..., n_k) &= \frac{P(n_1, ..., n_k|t)f_t}{\sum_{s \in S} P(n_1, ..., n_k|s)f_s} \\ &= \frac{\prod_{i=1}^{k} \sum_{N \in \{A,C,G,T,-\}} P_t(N) \times e_q(N \to n_i) f_t}{\sum_{s \in S} \prod_{i=1}^{k} \sum_{N \in \{A,C,G,T,-\}} P_s(N) \times e_q(N \to n_i) f_s} \end{aligned} \tag{28.3}$$

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 28.3 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 $n_1, ..., n_k$ in the reads at the site is:

$$\begin{aligned} P(t|n_1, ..., n_k) &= \frac{P(t, n_1, ..., n_k)}{\sum_{s \in S} P(s, n_1, ..., n_k)} \\ &= \frac{P(t)P(n_1, ..., n_k|t)}{\sum_{s \in S} P(s)P(n_1, ..., n_k|s)} \end{aligned} \tag{28.4}$$

Now, for $P(t)$ we use our current value for f_t, and if we further insert the expression for $P(n_1, ..., n_k|t)$ (28.2) we get:

$$P(t|n_1, ..., n_k) = \frac{f_t \prod_{i=1}^{k} \sum_{N \in \{A,C,G,T,-\}} P_t(N) \times e_q(N \rightarrow n_i)}{\sum_{s \in S} f_s \prod_{i=1}^{k} \sum_{N \in \{A,C,G,T,-\}} P_s(N) \times e_q(N \rightarrow n_i)} \tag{28.5}$$

We get the updating equation for the prior site type probabilities, $f_t, t \in S$, from equation 28.5: Let h index the sites in the alignment ($h = 1, ...H$). 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 28.5) across all sites in the alignment:

$$f_t^* = \frac{\sum_{h=1}^{H} \frac{f_t \prod_{i=1}^{k} \sum_{N \in \{A,C,G,T,-\}} P_t(N) \times e_q(N \to n_i^h)}{\sum_{s \in S} f_s \prod_{i=1}^{k} \sum_{N \in \{A,C,G,T,-\}} P_s(N) \times e_q(N \to n_i^h)}}{H} \tag{28.6}$$

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 $n_1^h, ..., n_{k_h}^h$ is:

$$\begin{aligned}
&P(r_i^h = N, n_1^h, ..., n_{k_h}^h) \\
&= \sum_{s \in S} f_s P(r_i^h = N, n_1^h, ..., n_{k_h}^h | s) \\
&= \sum_{s \in S} f_s \prod_i P(r_i^h = N, n_i^h | s) \\
&= \sum_{s \in S} f_s P(r_i^h = N, n_i^h | s) \prod_{j \neq i} P(n_j^h | s) \\
&= \sum_{s \in S} f_s (P_s(N) \times e_{q_{i^h}}(N \to n_i^h) \prod_{j \neq i} \sum_{N' \in \{A,C,G,T,-\}} P_s(N') \times e_{q_j}(N' \to n_j^h))
\end{aligned} \tag{28.7}$$

Using Bayes formula again, as we did above in 28.4, we get:

$$\begin{aligned}
P(r_i^h = N | n_1^h, ..., n_{k_h}^h) &= \frac{P(r_i^h = N, n_1^h, ..., n_k^h)}{P(n_1^h, ..., n_{k_h}^h)} \\
&= \frac{P(r_i^h = N, n_1^h, ..., n_{k_h}^h)}{\sum_{N' \in \{A,C,G,T,-\}} P(r_i^h = N', n_1^h, ..., n_{k_h}^h)}
\end{aligned} \tag{28.8}$$

and inserting the expression from equation 28.7:

$$\begin{aligned}
&P(r_i^h = N | n_1^h, ..., n_{k_h}^h) \\
&= \frac{\sum_{s \in S} f(s)(P_s(N) \times e_{q_{i^h}}(N \to n_i^h) \prod_{j \neq i} \sum_{N' \in \{A,C,G,T,-\}} P_s(N') \times e_{q_j^h}(N' \to n_j^h))}{\sum_{N' \in \{A,C,G,T,-\}} (\sum_{s \in S} f(s)(P_s(N') \times e_{q_{i^h}}(N' \to n_i^h) \prod_{j \neq i} \sum_{N' \in \{A,C,G,T,-\}} P_s(N') \times e_{q_j^h}(N' \to n_j^h)))} \\
&= \frac{\sum_{s \in S} f(s)(P_s(N) e_{q_{i^h}}(N \to n_i^h) \prod_{j \neq i} \sum_{N' \in \{A,C,G,T,-\}} P_s(N') \times e_{q_j^h}(N' \to n_j^h))}{(\sum_{s \in S} f(s) \prod_j \sum_{N' \in \{A,C,G,T,-\}} P_s(N') \times e_{q_j^h}(N' \to n_j^h)}
\end{aligned} \tag{28.9}$$

The equation 28.9 gives us the probabilities for a given read, i, and site, h, given the data $n_1^h, ..., n_k^h$, 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, $e_q^*(N \to M)$, 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:

$$e_q^*(N \to M) = \frac{\sum_h \sum_{i=1,...,k_h : n_i^h = M} P(r_i^k = N | n_1^h, ..., n_{k_h}^h)}{\sum_h \sum_{i=1,...,k_h} P(r_i^k = N | n_1^h, ..., n_{k_h}^h)}$$

28.7.3 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 28.7.2.

The multinomial models are of the kind "there are q different nucleotide alleles present at the site with frequencies f_i, $i = 1, ..., q$, $\sum_{i=1}^{q} f_i = 1$", where the number of alleles, q, differ. The models that are evaluated at each site are given in Table 28.1.

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 w at frequency f_4	f_1, f_2, f_3 and f_4

Table 28.1: 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 loglikehood 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, $m = 1...,M$. For each model we calculate the value:

$$v_m = 2\log L(H_m) - c_{df_m}(p) \tag{28.10}$$

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, $i = 1,...,k$. 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:

$$f^* = \frac{\sum_{i=1}^{k} P(r_i^h = x|n_i^h)}{k}. \tag{28.11}$$

To calculate this we use Bayes Theorem on the numerator:

$$P(r_i^h = x|n_i^h) = \frac{P(r_i^h = x, n_i^h)}{P(n_i^h)} \tag{28.12}$$

$$= \frac{P(x) \times e(x \to n_i^h)}{P(x) \times e(x \to n_i^h) + P(y) \times e(y \to n_i^h)} \tag{28.13}$$

$$= \frac{f \times e(x \to n_i^h)}{f \times e(x \to n_i^h) + (1-f) \times e(y \to n_i^h)} \tag{28.14}$$

Inserting our current values for the frequency parameter f under the model, and the error rates $e(x \to n_i^h)$ and $e(y \to n_i^h)$, in 28.12, and further inserting the obtained values in 28.11 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:

$$P(r_i^h = N, n_i^h) = P_h(N)e_{q_i^h}(N \rightarrow n_i^h). \quad (28.15)$$

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:

$$P(r_i^h = N|n_i^h) = \frac{P(r_i^h = N, n_i^h)}{\sum_{N' \in A,C,G,T,-} P(r_i^h = N', n_i^h)}. \quad (28.16)$$

Inserting 28.15 in 28.16 we get:

$$P(r_i^h = N|n_i^h) = \frac{P_h(N)e_{q_i^h}(N \rightarrow n_i^h)}{\sum_{N' \in A,C,G,T,-} P_h(N')e_{q_i^h}(N' \rightarrow n_i^h)}. \quad (28.17)$$

The equation 28.17 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:

$$e_q^*(N \rightarrow M) = \frac{\sum_h \sum_{i=1,\ldots,k_h : n_i^h = M} P(r_i^k = N|n_i^h)}{\sum_h \sum_{i=1,\ldots,k_h} P(r_i^k = N|n_i^h)}$$

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:

$$P_m = \frac{e^{v_m}}{\sum_{m'} e^{v_{m'}}}$$

28.8 Copy Number Variant Detection

The Copy Number Variant Detection 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 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 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.

28.8.1 The Copy Number Variant Detection tool

To run the Copy Number Variant Detection tool, go to:

Toolbox | **Resequencing Analysis (**  **)** | **Copy Number Variant Detection (**  **)**

Select the case read mapping and click **Next**.

You are now presented with choices regarding the data to use in the CNV prediction method, as shown in figure 28.22.

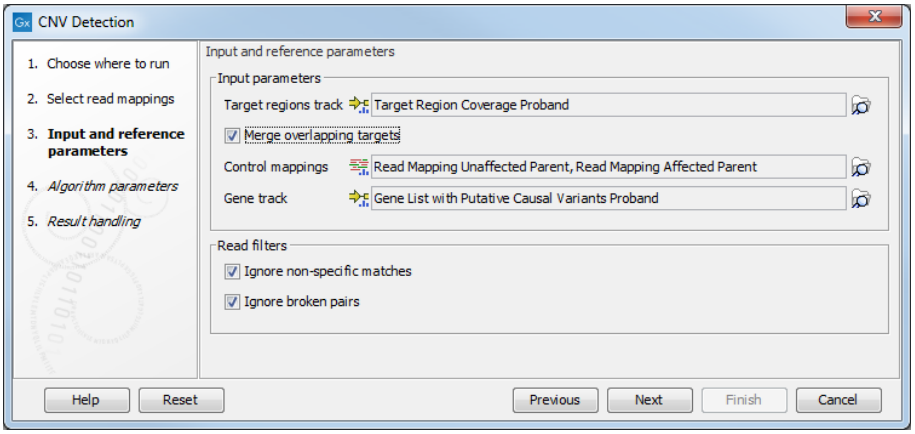


Figure 28.22: *The first step of the CNV detection tool.*

- **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.
- **Control mappings** You must specify one or more read mappings, which will be used to create a baseline by the algorithm. 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 28.23.

- **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, absolute value** You must specify the minimum fold change for a CNV call. 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, then the adjusted coverage of the CNV in the case sample must be either 1.5 times higher or 1.5

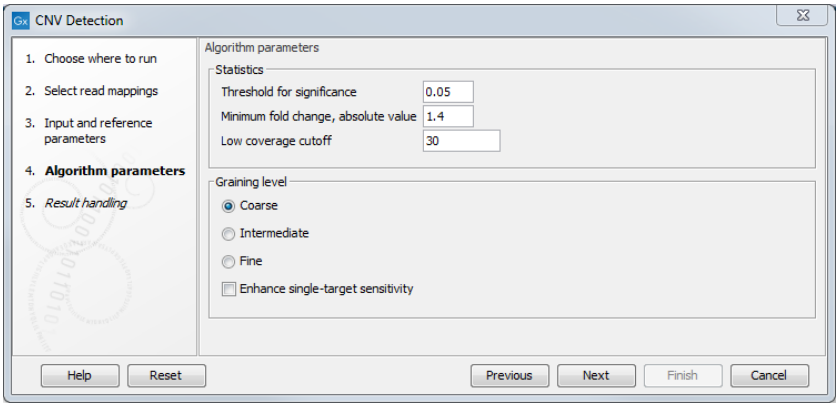


Figure 28.23: *The second step of the CNV detection tool*

times lower than the coverage in the baseline, for it to pass the filtering step. If you do not want to filter on the fold-change, enter 0.0 in this field. Also, if your sample purity is less than 100%, it is necessary to take that into account when you adjust the fold-change cutoff. This is described in more detail in section 28.8.1. 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, 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.

- **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 28.24). 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.

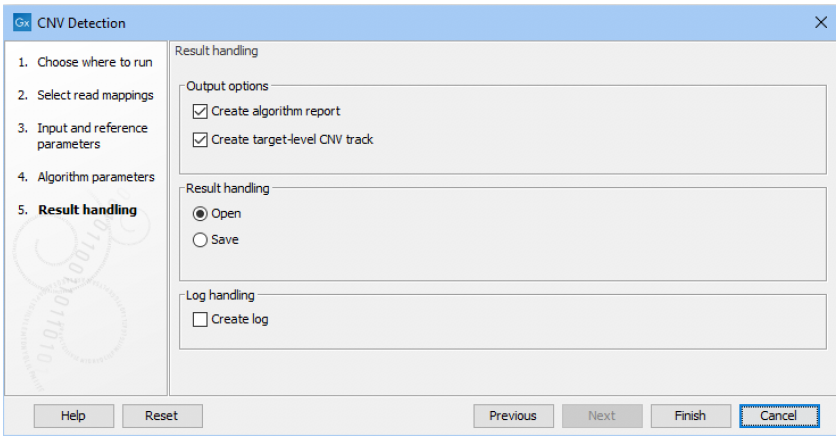


Figure 28.24: *Specifying whether an algorithm report and a target-level CNV track should be created.*

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:

$$\text{Fold change, amplifications (CN(sample) > CN(normal))} = \frac{\text{CN(sample)}}{\text{CN(normal)}} \tag{28.18}$$

$$\text{Fold change, deletions (CN(sample) < CN(normal))} = -\frac{\text{CN(normal)}}{\text{CN(sample)}} \tag{28.19}$$

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 28.2: 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:

$$\text{cutoff} = \frac{X\%}{100\%} \times T + (1 - \frac{X\%}{100\%}) \tag{28.20}$$

For example, if the sample purity is 40%, and you want to detect 6-fold amplifications or deletions (e.g. 12 copies instead of 2, or 2 copies instead of 12), then the cutoff should be:

$$\text{cutoff} = \frac{40\%}{100\%} \times 6 + (1 - \frac{40\%}{100\%}) = 3.0. \tag{28.21}$$

Figure 28.25 shows the required fold-change cutoffs in order to detect a particular degree of amplification/deletion at different sample purities. Figure 28.26 zooms in for low-level amplifications and deletions.

The Copy Number Variant Detection 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.

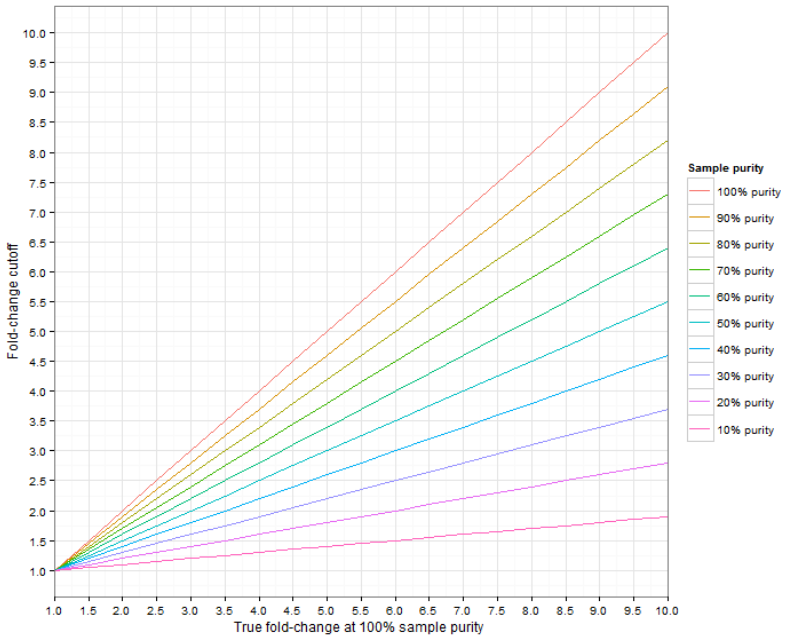


Figure 28.25: *The required fold-change cutoff to detect amplifications and deletions of different magnitudes, as a function of sample purity.*

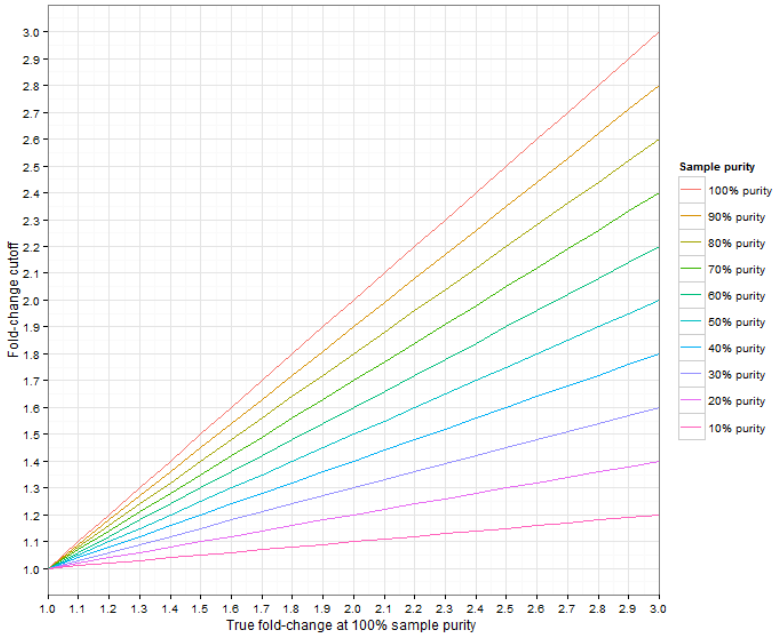


Figure 28.26: *Low-level amplifications and deletions: The required fold-change cutoff to detect amplifications and deletions of different magnitudes, as a function of sample purity.*

28.8.2 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 more 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 be 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).

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. 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 28.8.2.

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 28.8.3).

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.

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 observed fold-change of F, following formula gives the actual fold-change that would be seen if the sample were 100% pure:

$$\text{fold-change in 100\% pure sample} = \frac{F-1}{X/100\%} + 1 \tag{28.22}$$

For example, if the sample purity is 40%, and you have observed a fold-change of 3, then the fold-change in the 100% pure sample would have been:

$$\text{fold-change in 100\% pure sample} = \frac{3.0-1}{40\%/100\%} + 1 = 6.0. \tag{28.23}$$

Figures 28.27 shows the 'true' fold changes for different observed fold-changes at different sample purities. Figure 28.28 zooms in for low-level amplifications and deletions.

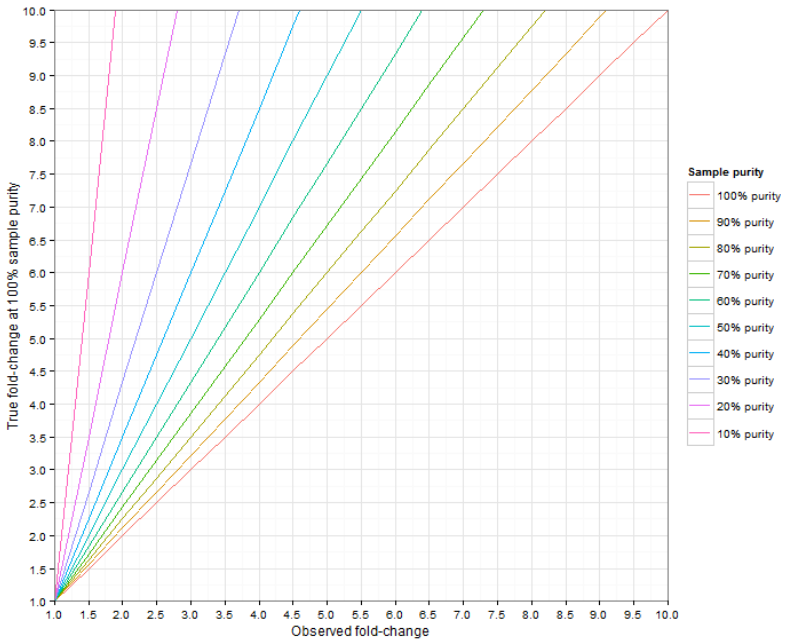


Figure 28.27: *The true fold-change in the 100% pure sample, for different observed fold-changes, as a function of sample purity.*

28.8.3 Target-level CNV track (Target CNVs)

The algorithm will produce a target-level CNV track, if you've 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 give you additional information about both the CNVs called in the region-level results, and those regions that have not been called. Note that a "statistically relevant" target is one that has a coverage higher than the specified coverage cutoff, AND is found on a a chromosome that was not identified as a coverage outlier in the chromosomal analysis step. The sample is not considered covered enough for statistical purposes if you have fewer than 49 targets that can be used to do the statistics.

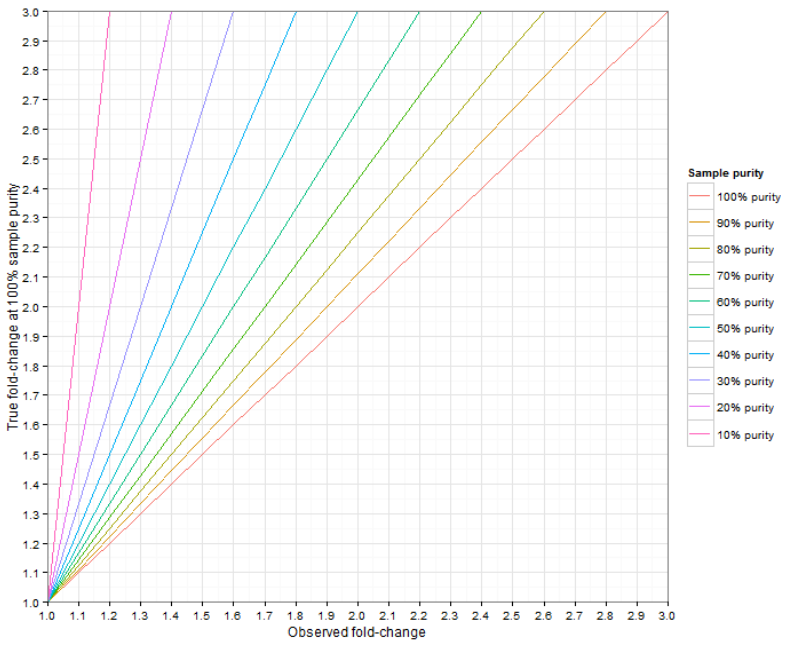


Figure 28.28: *Low-level amplifications and deletions: the true fold-change in the 100% pure sample, for different observed fold-changes, as a function of sample purity.*

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 28.8.5).

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 28.8.2.

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.

28.8.4 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 28.8.2, 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.

28.8.5 CNV results report

The report contains information about the results of the Copy Number Variant Detection 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 log-ratios 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 28.29. On the horizontal axis, the targets are placed in the order in which they appear in the genome. On the vertical axis, the adjusted coverage log-ratio of each target is plotted. The black line represents the actually observed mean adjusted log-ratio of coverage for each target. The cyan and red lines represent the 95% confidence intervals of the expected mean adjusted log-ratios of coverages, based on the statistical model. Chromosome boundaries are indicated as vertical lines.

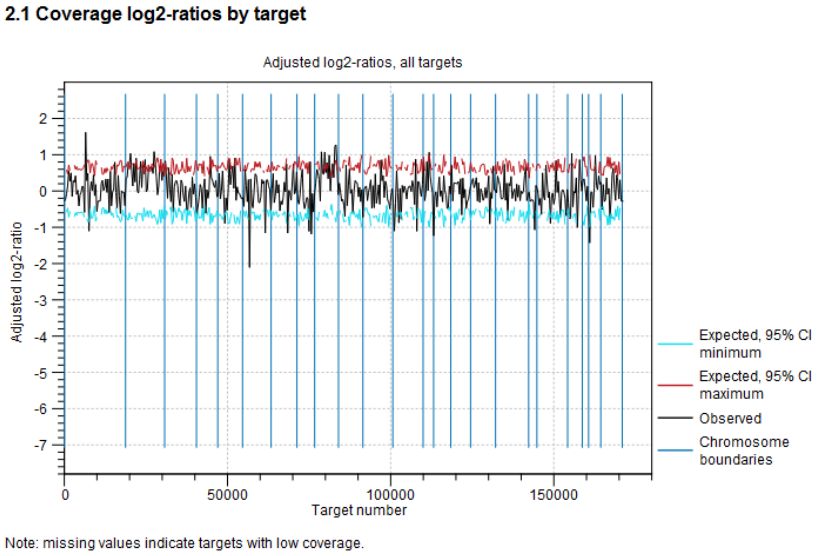


Figure 28.29: *An example graph showing the mean adjusted log-ratios of coverages in the report produced by the Copy Number Variant Detection tool. In this example, the second and ninth chromosomes are amplified, and the log-ratios of coverages of targets on these chromosome are significantly higher than for targets on other chromosomes. The black line in these regions is outside the boundaries defined by the cyan and red lines.*

CNV statistics The last section in the report provides some information about the number of CNVs called in the region-level prediction results. The number of uncalled or filtered regions are also shown.

28.8.6 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 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 28.30).

The graph is followed by a table, where the detailed chromosome coverages are shown after normalization. Chromosomes with disproportionate coverage and chromosomes without any 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.

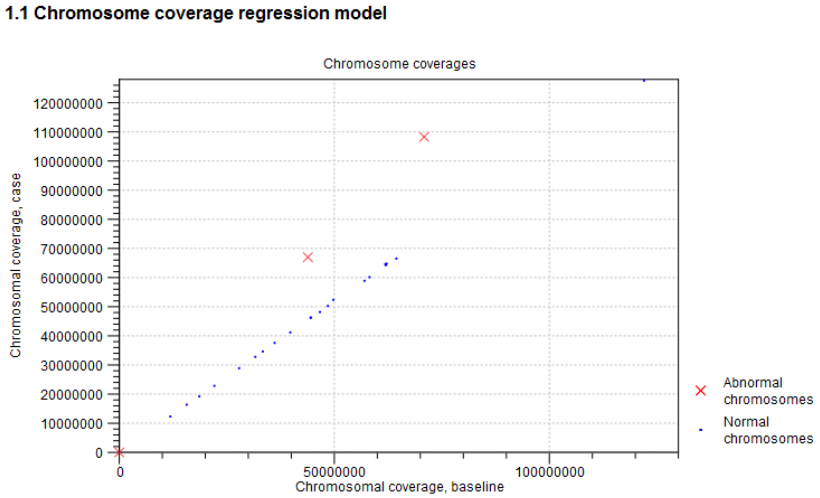


Figure 28.30: *An example graph showing the coverages of the chromosomes in the case versus the baseline. In this example, three chromosomes are marked as abnormal. Two of these chromosomes are significantly amplified, and log-ratios of coverages of many targets on these chromosome are significantly higher than for targets on other chromosomes. The third outlier chromosome had zero coverage in both the case and the baseline.*

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 28.31). 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 28.32). 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 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 28.33). A good fit of the model to the data points indicates that the variance has been modeled accurately.

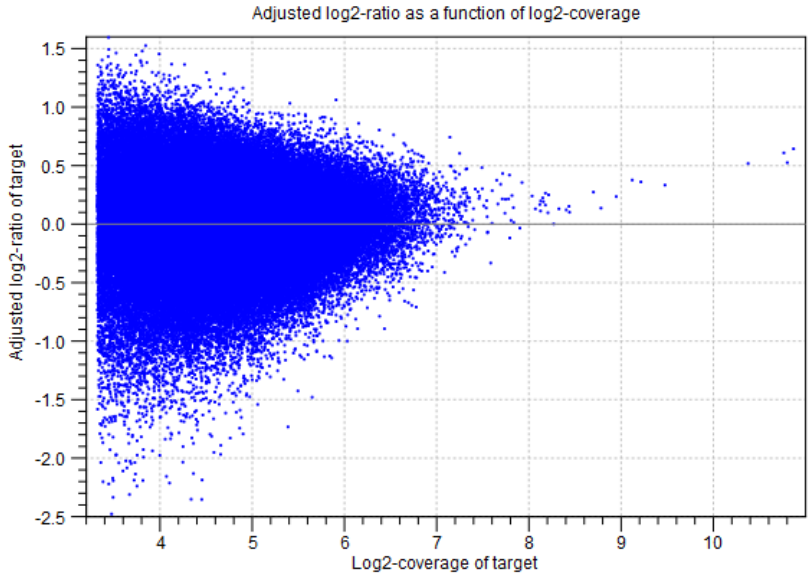


Figure 28.31: *An example graph showing the mean adjusted log-ratios of coverages plotted against the log-coverages of targets, in the algorithm report of the Copy Number Variation Detection tool. Here, the adjusted mean log-ratios are centered around 0.0 for most coverages, and the variation decreases 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.*

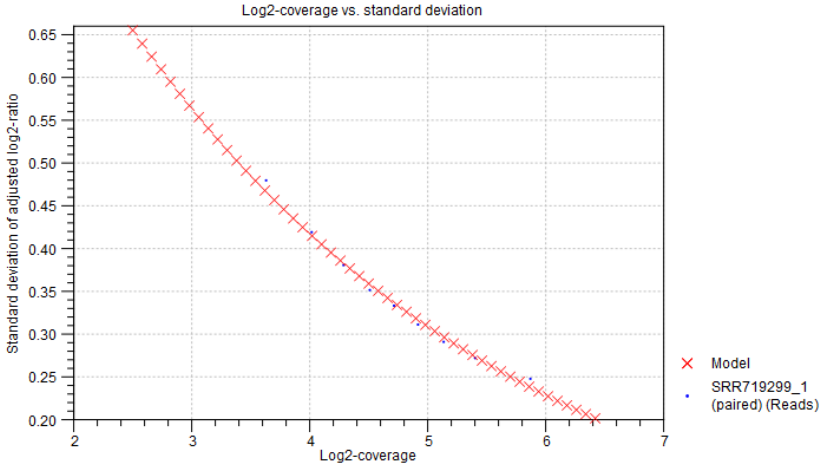


Figure 28.32: *An example graph showing how the variance in the target-level mean log-ratios was modeled in the algorithm report of the Copy Number Variation Detection tool. Here, the data points are very close to the fitted model, indicating a good fit of the model to the data.*

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,

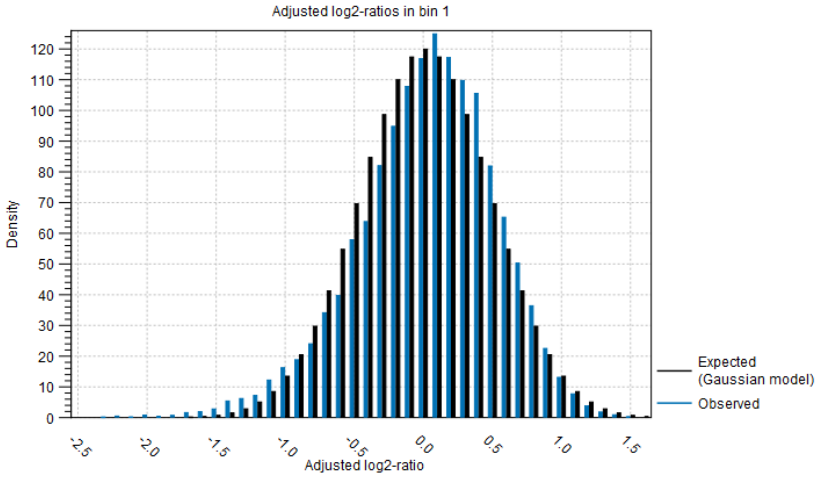


Figure 28.33: *An example bar chart from the algorithm report of the Copy Number Variation Detection tool, showing how well the normal distribution assumption was fulfilled by the adjusted coverage log-ratios. Here, there is a good correspondence between the expected distribution and the observations.*

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.

28.9 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.

28.9.1 Run the Identify Known Mutations from Sample Mappings tool

To run the "Identify Known Mutations from Sample Mappings" tool go to the toolbox:

Toolbox | **Resequencing Analysis** ( ) | **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 28.34):

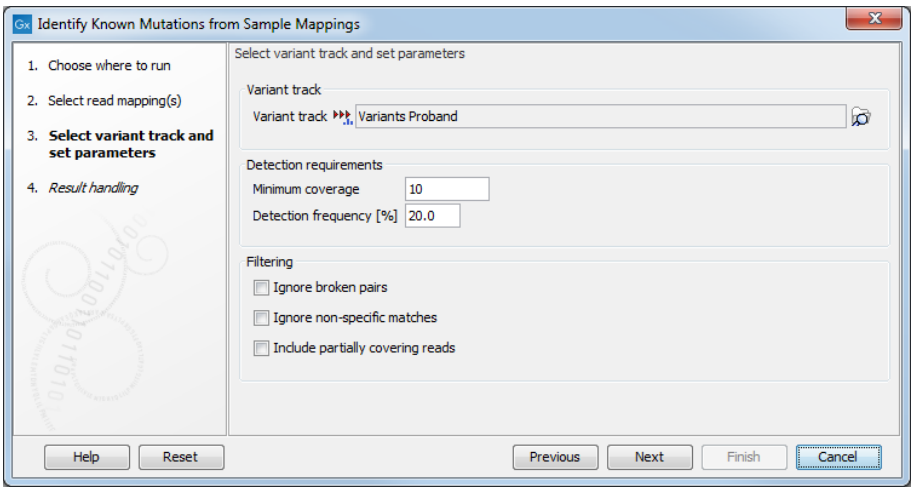


Figure 28.34: *Select the variant track with the variants that you wish to use for variant testing.*

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 loose 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 28.35). At this step the output options can be adjusted.

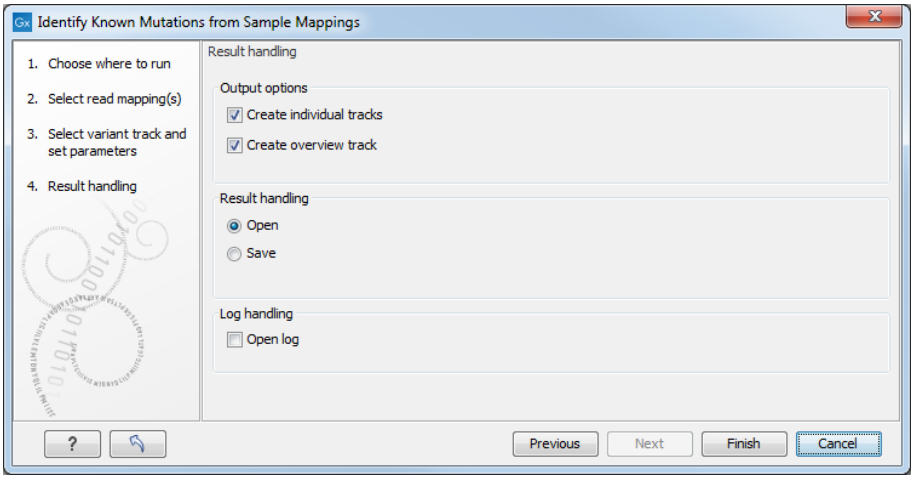


Figure 28.35: *Select the desired output format(s). If using the default settings, two types of output will be generated; individual tracks and overview tracks.*

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**.

28.9.2 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 28.36).

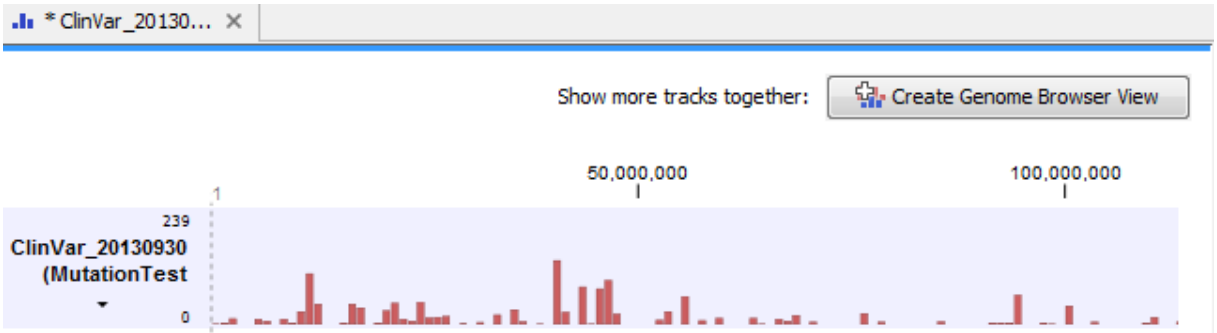


Figure 28.36: *Overview track of read mappings tested against a Clinvar variant track.*

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.

In the individual track, the variant has been annotated with most classical variant track annotations (see section 28.6.1), 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 28.37.

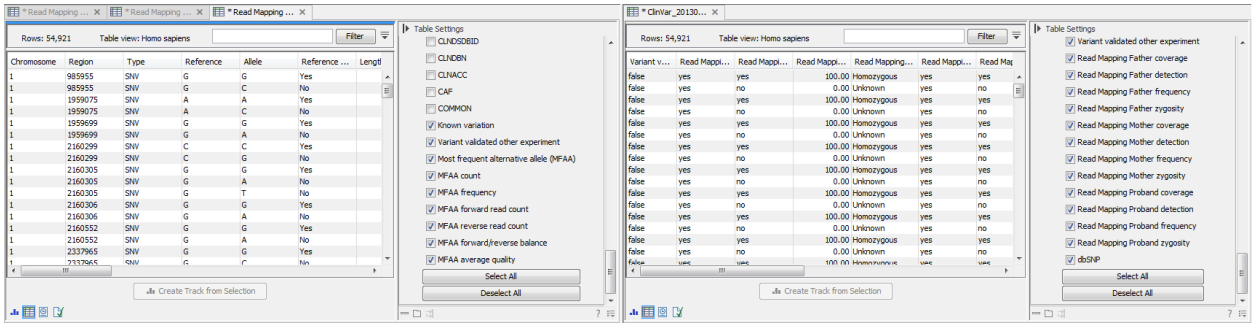


Figure 28.37: *Table views of the individual track (left) and overview track (right).*

28.10 InDels and Structural Variants

The InDels and Structural Variants tool is designed to identify structural variants such as insertions, deletions, inversions, translocations and tandem duplications in read mappings. 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 - irrespectively 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.

28.10.1 Run the InDels and Structural Variants tool

Go to:

Toolbox | **Resequencing Analysis** ( ) | **InDels and Structural Variants tool** ( )

This will open up a dialog. Select the read mapping of interest as shown in figure 28.38 and click on the button labeled **Next**.

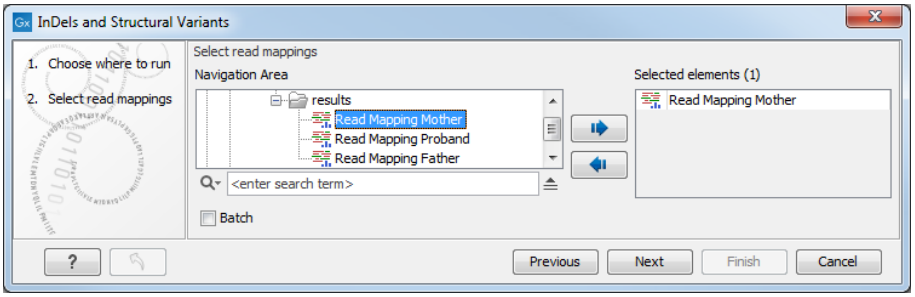


Figure 28.38: *Select the read mapping of interest.*

The next wizard step (Figure 28.39) 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.

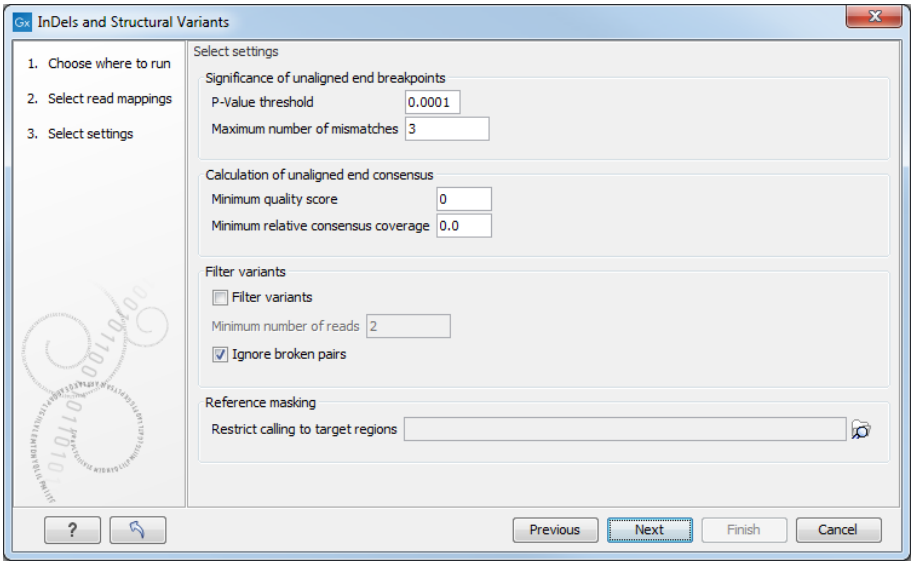


Figure 28.39: *Select the relevant settings.*

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 28.40) will be opened. The Indels and Structural Variants tool has the following output options:

- **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

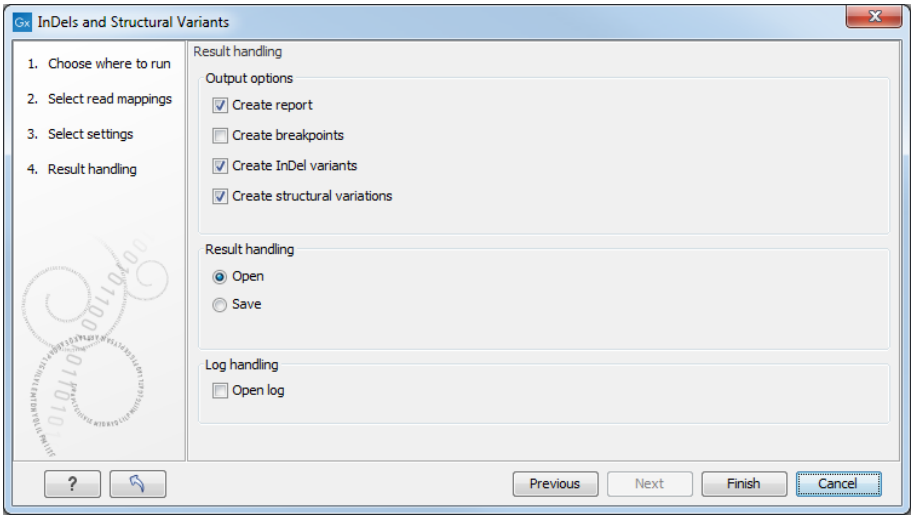


Figure 28.40: *Select output formats.*

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 28.6.1 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 28.41. The output is described in detail in section 28.10.2.

28.10.2 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 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.

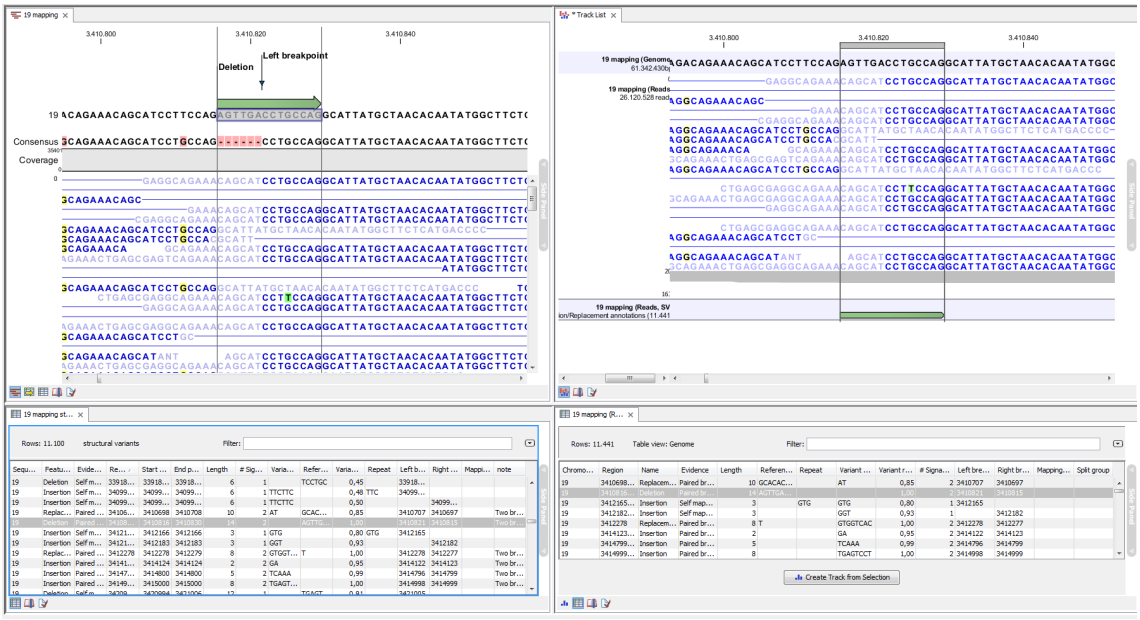


Figure 28.41: *Example of the result of an analysis on a standalone read mapping (to the left) and on a reads track (to the right).*

- A distribution of the logarithm of the sequence complexity of the unaligned ends of the left and right breakpoint signatures (see section 28.10.5 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 28.10.5 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 28.6.1). 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 of 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 28.10.5). 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', 'translocation' 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 28.10.4). 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 28.10.5).
- **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.

28.10.3 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 28.42. 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.

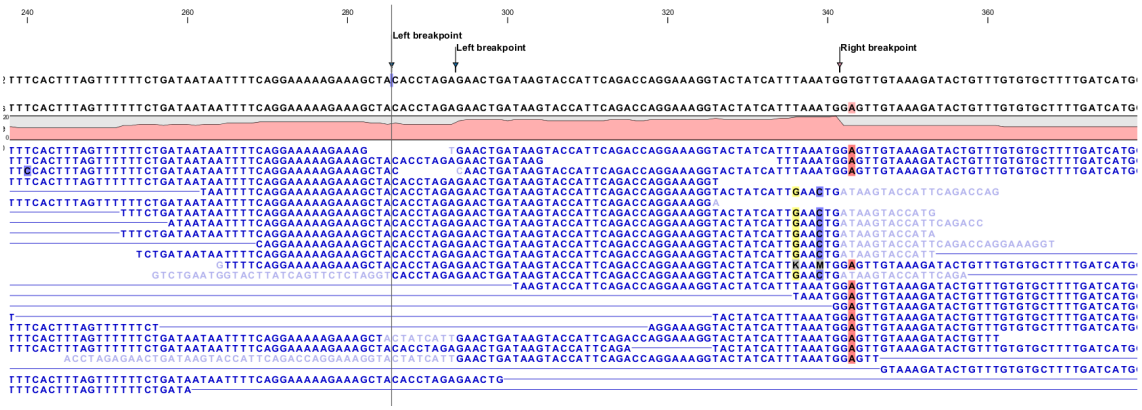


Figure 28.42: *Example of a read mapping containing unaligned ends with three unaligned end signatures.*

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 28.10.1 and figure 28.39). 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 28.42, 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 28.10.4).
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.

28.10.4 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 28.43 to 28.51.

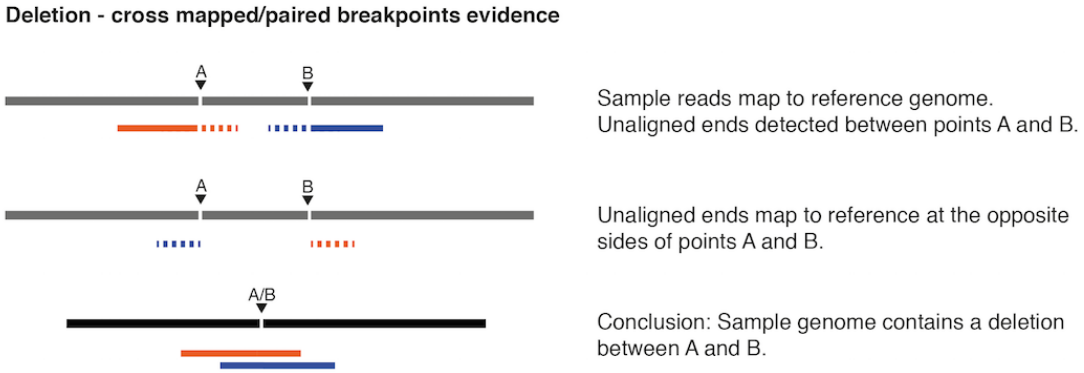


Figure 28.43: *A deletion with cross-mapping breakpoint evidence.*

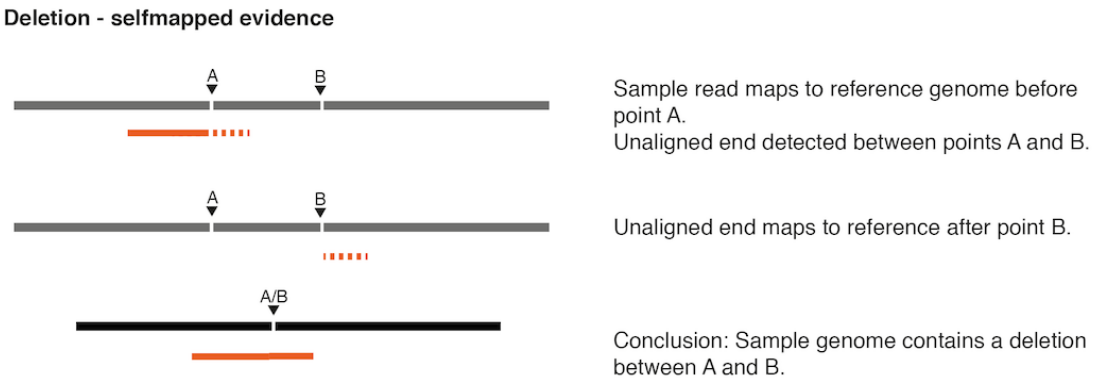


Figure 28.44: *A deletion with selfmapping breakpoint evidence.*

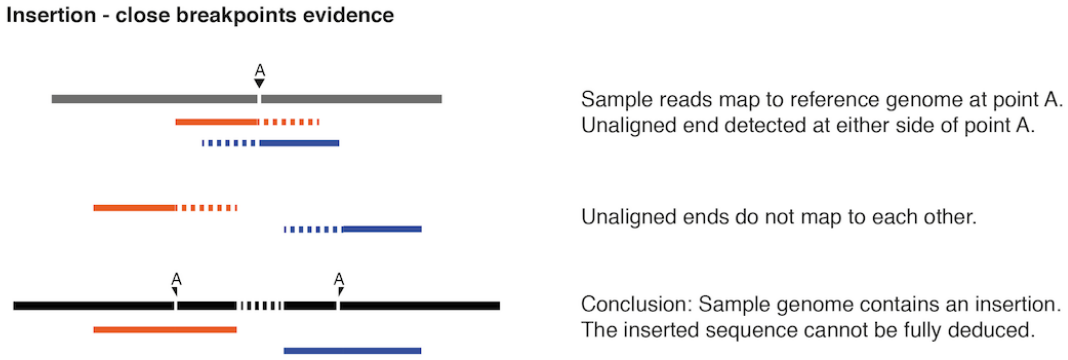


Figure 28.45: *An insertion with close breakpoint evidence.*

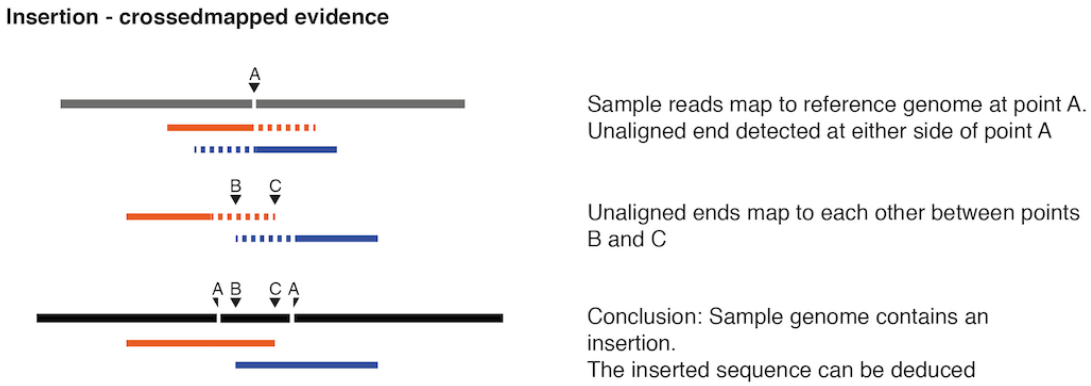


Figure 28.46: *An insertion with cross-mapped breakpoints evidence.*

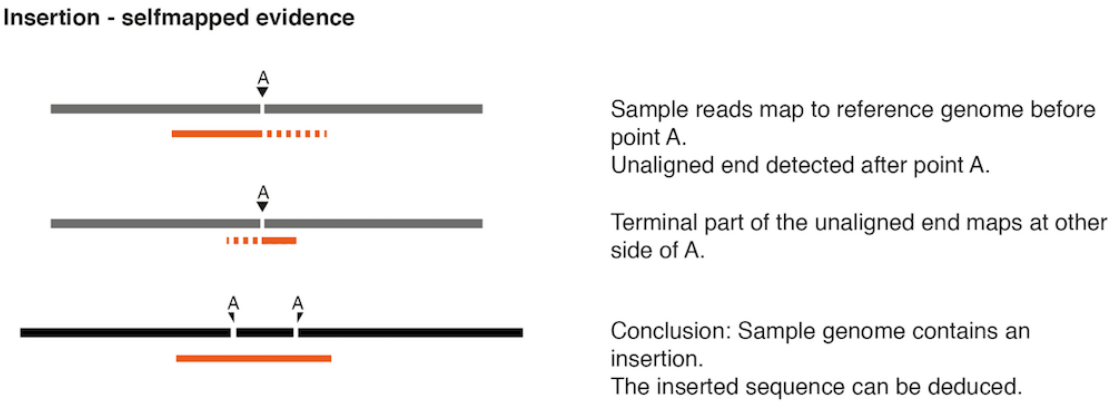


Figure 28.47: *An insertion with selfmapped breakpoint evidence.*

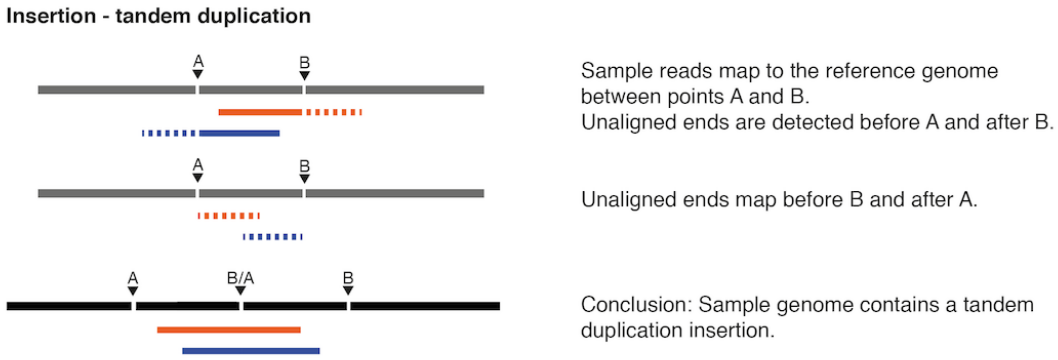


Figure 28.48: *An insertion with breakpoint mapping evidence corresponding to a 'Tandem duplication'.*

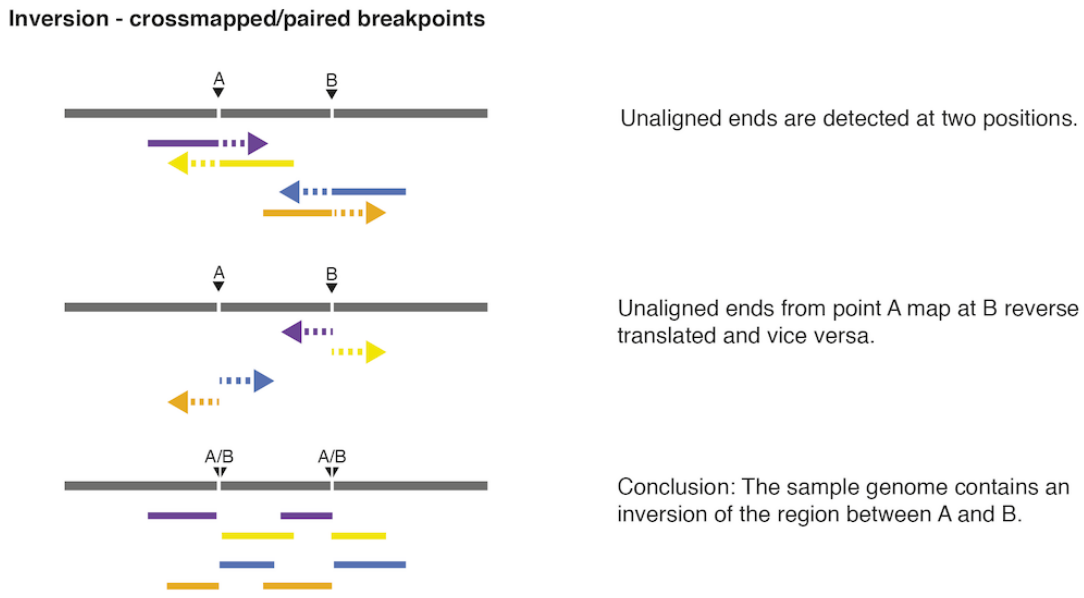


Figure 28.49: *The unaligned end mapping pattern of an inversion.*

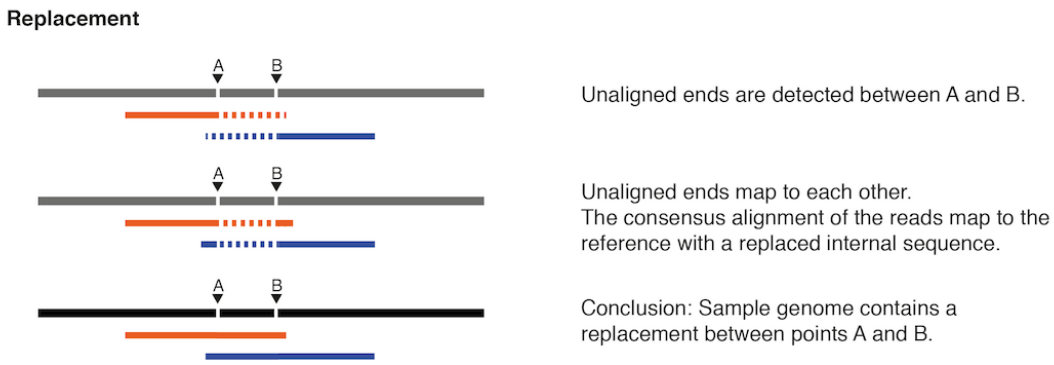


Figure 28.50: *The unaligned end mapping pattern of a replacement.*

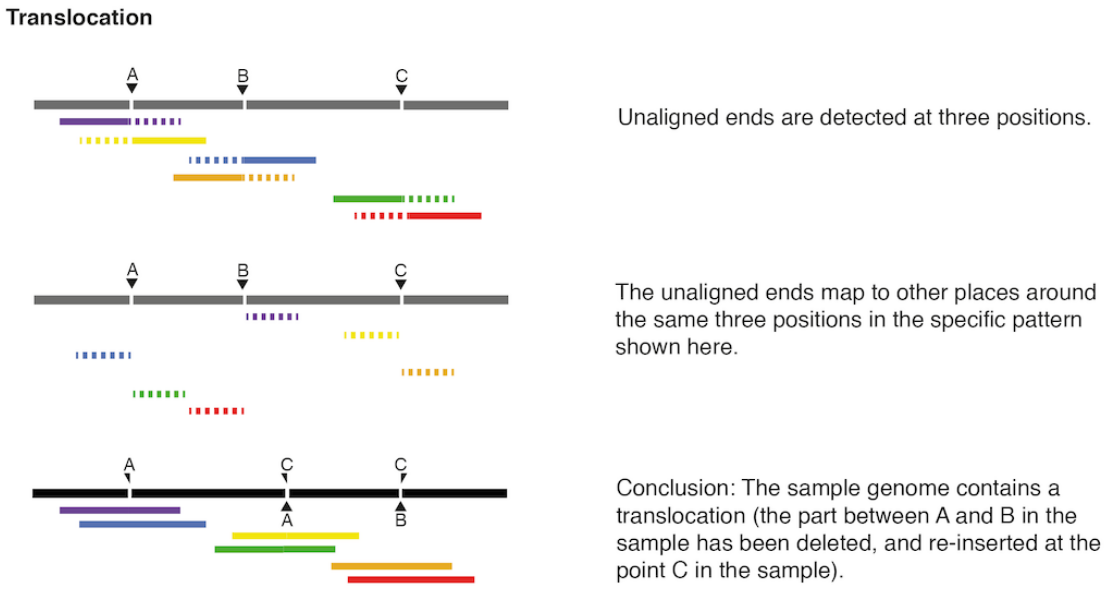


Figure 28.51: *The unaligned end mapping pattern of a translocation.*

28.10.5 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 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 dinucleotides (4*4*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 quatronucleotides, so with 256 possibilities, the quatronucleotides 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

Chapter 29

Resequencing

Contents

29.1 Variant filtering . . . 768
29.1.1 Filter against Known Variants . . . 768
29.1.2 Remove Marginal Variants . . . 770
29.1.3 Remove Homozygous Reference Variants . . . 771
29.1.4 Remove Variants Present in Control Reads . . . 771
29.2 Variant annotation . . . 772
29.2.1 Annotate from Known Variants . . . 772
29.2.2 Remove Information from Variants . . . 772
29.2.3 Annotate with Effect Scores . . . 773
29.2.4 Annotate with Conservation Score . . . 774
29.2.5 Annotate with Exon Numbers . . . 775
29.2.6 Annotate with Flanking Sequence . . . 775
29.2.7 Annotate with Repeat and Homopolymer Information . . . 776
29.3 Variants comparison . . . 777
29.3.1 Identify Shared Variants . . . 777
29.3.2 Identify Enriched Variants in Case vs Control Samples . . . 778
29.3.3 Trio Analysis . . . 779
29.4 Variant quality control . . . 782
29.4.1 Create Variant Track Statistics Report . . . 782
29.5 Functional consequences . . . 783
29.5.1 Amino Acid Changes . . . 783
29.5.2 Predict Splice Site Effect . . . 789
29.5.3 GO Enrichment Analysis . . . 790
29.5.4 Download 3D Protein Structure Database . . . 792
29.5.5 Link Variants to 3D Protein Structure . . . 793

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 27.

29.1 Variant filtering

In addition to the general filter for track tables, including the ability to create a new track from a selection (see section 24.2.2), there are a number of tools for general filtering of variants.

For functional filtering, see section 29.5.

29.1.1 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 6.2 and section 8.1).

To get started, go to:

> **Toolbox** | **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 29.1

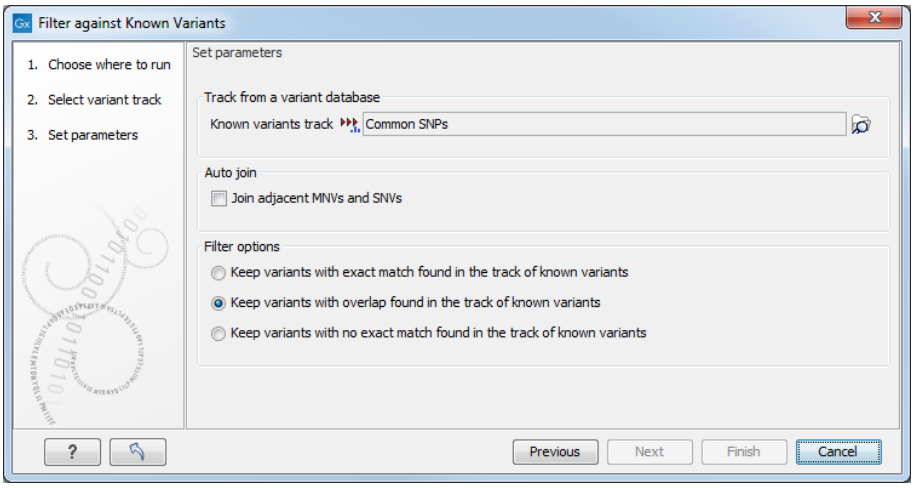


Figure 29.1: *Specifying a variant track to filter against.*

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.

29.1.2 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:

> **Toolbox** | **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 29.2.

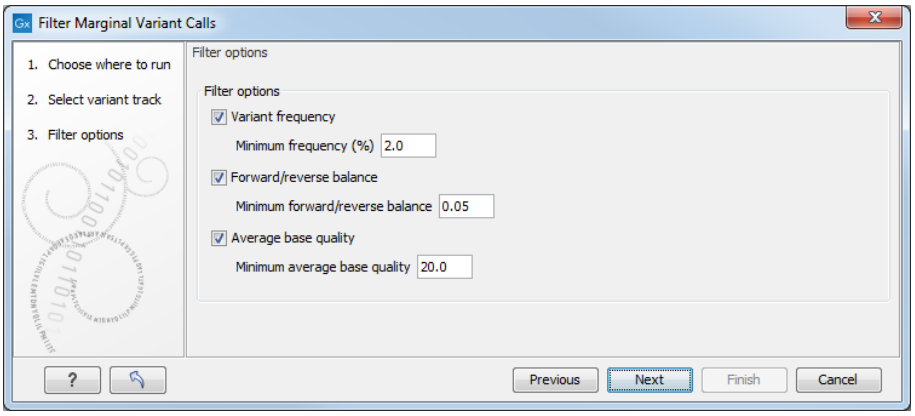


Figure 29.2: *One or more thresholds can be configured, defining the basis for variant removal.*

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 28.6.1.

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.

29.1.3 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:

> **Toolbox** | **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.

29.1.4 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.

> **Toolbox** | **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 29.3). 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.

The filter option can be used to set a threshold for which variants should be kept. In the dialog shown in figure 29.3 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.

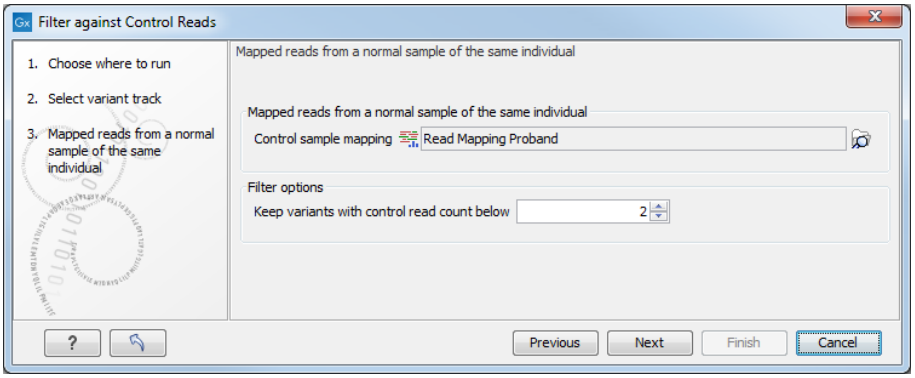


Figure 29.3: *The control reads data set.*

29.2 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 29.5.

29.2.1 Annotate from Known Variants

To run the Annotate from known variants tool, go to:

Toolbox | **Resequencing Analysis** ( ) | **Variant Annotation** ( ) | **Annotate from Known Variants** ( )

This tool will create a new track with all the experimental variants including added information about overlapping variants found in the track of known variants. The annotations 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.

For exact matches, all the information about the variant from the known variants track is transferred to the annotated variant. For partial matches and overlaps, the information from the known variants are not transferred.

29.2.2 Remove Information from Variants

The tool removes selected annotations from a variant track.

To start the tool, go to:

Toolbox | Resequencing Analysis ( ) | Variant Annotation ( ) | Remove Information from Variants ( )

In the first dialog, select a variant track (figure 29.4). To process several variant tracks at once, check the batch box. In this case, the next dialog will be the batch overview.

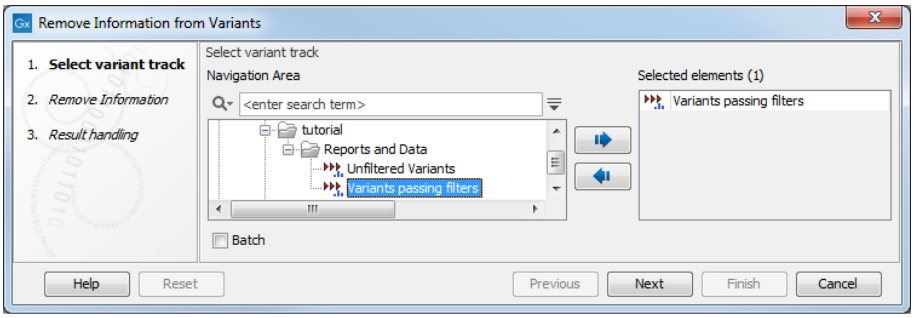


Figure 29.4: *Select a variant track.*

In the next dialog, click on the button **Load Annotations** to fill the Annotations field underneath as seen in figure 29.5. The content of the Annotations list depends on the annotations present in the track selected as input (and when batching, only in the first track selected). Choose with the radio button whether you want to remove or keep annotations, and select the annotations you wish to remove/keep - depending on what is easiest for your purpose. by clicking on the button Simple View, you can get the list of selected annotations only for verification before clicking **Next**.

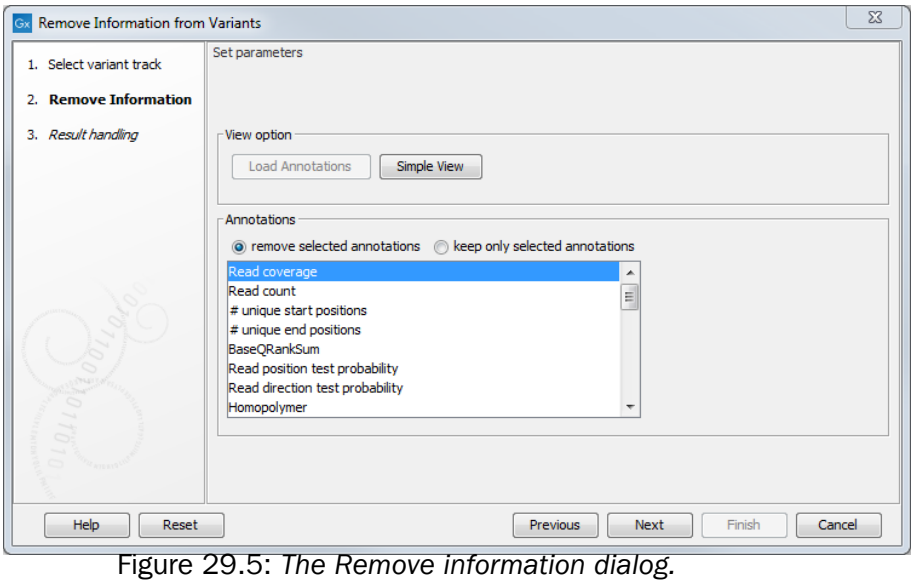


Figure 29.5: *The Remove information dialog.*

The result of the tool is, for each input file, a similar track containing only the annotations that were chosen to be kept.

29.2.3 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:

> **Toolbox** | **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 29.6). 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.

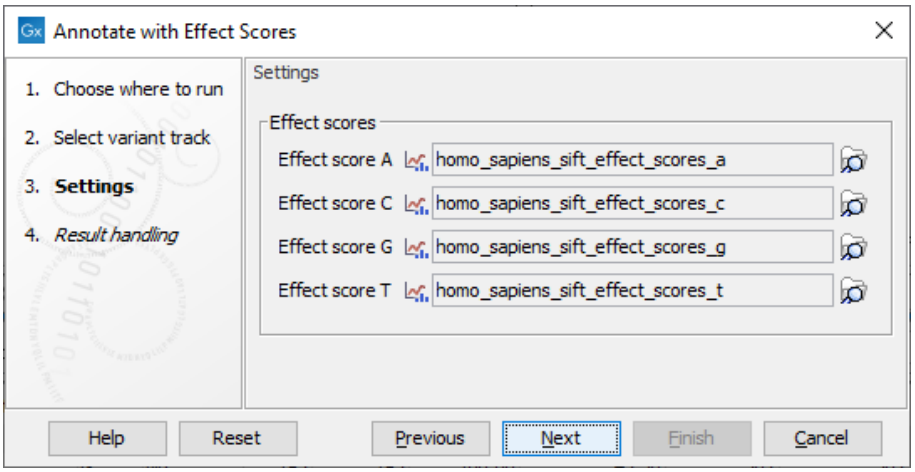


Figure 29.6: *Select Effect Score tracks.*

The tool outputs a variant track with an **Effect score** annotation added.

29.2.4 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 6.2.

> **Toolbox** | **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 29.7).

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 24.2.2).

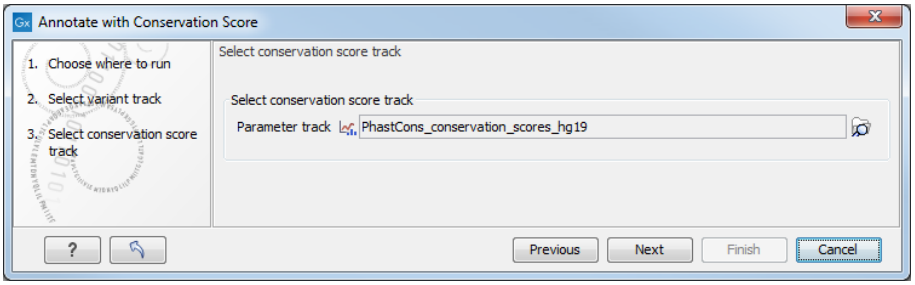


Figure 29.7: *The conservation score track.*

29.2.5 Annotate with Exon Numbers

To run the Annotate with Exon Numbers tool, go to:

> **Toolbox** | **Resequencing Analysis** ( ) | **Variant Annotation** ( ) | **Annotate with Exon Numbers** ( )

Given a track with mRNA annotations, a new track will be created in which variants are annotated with the numbering of the corresponding exon with numbered exons based on the transcript annotations in the input track (see an example of a result in figure 29.8).

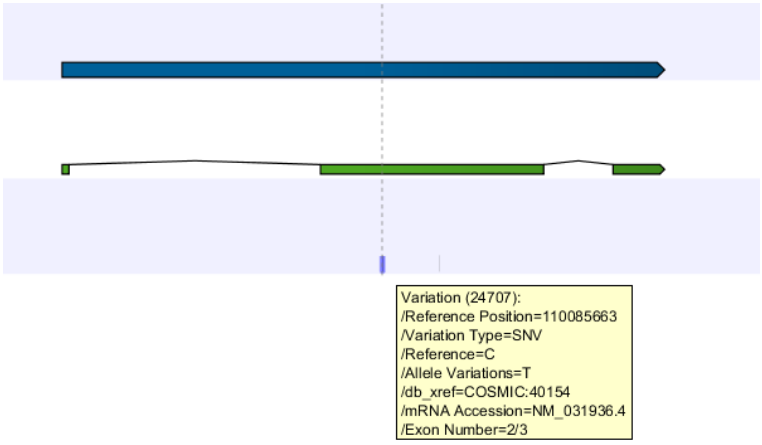


Figure 29.8: *A variant found in the second exon out of three in total.*

When there are multiple isoforms, a comma-separated list of the exon numbers is given.

29.2.6 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:

> **Toolbox** | **Resequencing Analysis** ( ) | **Variant Annotation** ( ) | **Annotate with Flanking Sequence** ( )

This opens a dialog where you can select a variant track ( ) to be annotated.

Clicking **Next** will display the dialog shown in figure 29.9

Select a sequence track that should be used for adding the flanking sequence, and specify how large the flanking region should be.

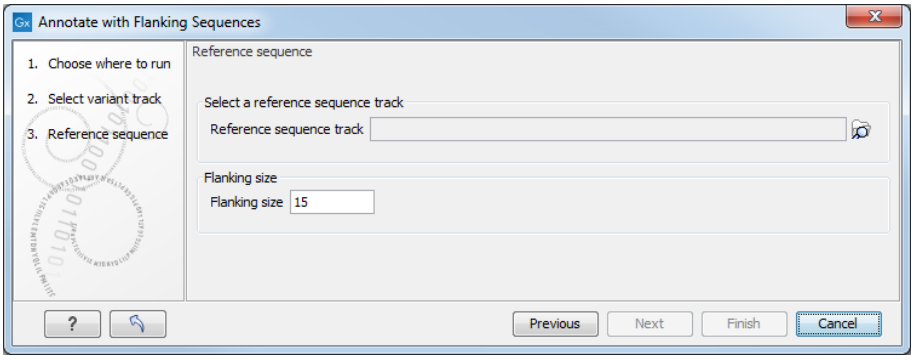


Figure 29.9: *Specifying a reference sequence and the amount of flanking bases to include.*

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.

29.2.7 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 copies of the variant's base type at that location on the reference, 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 of that variant 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 of that variant at that location on the reference, or for deletions, next to where the deletion occurred.

To determine if there is a homopolymer or repeat in a given reference region, a hidden Markov model (HMM) is used. The HMM will allow for some degree of mismatch between repeated elements on the reference if it determines that the sequence is still most likely to be a homopolymer or repeat. However, even where mismatches between repeated elements on the reference have been allowed, an insertion/replacement will not be marked as being part of a homopolymer or repeat region if there are any mismatches between it and the repeats next to it.

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 the Annotate with Repeat and Homopolymer Information tool, go to:

> **Toolbox | Resequencing Analysis ( ) | Variant Annotation ( ) | Annotate with Repeat and Homopolymer Information ( )**

The tool takes variant tracks ( ) as input.

In the next dialog, the reference sequence the variant track is based on should be selected.

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 an insertion or deletion in a homopolymer region, or "No" if it is not.
- **Repeat region** The value is "Yes" if the variant is an insertion or deletion in a repeat region, or "No" if it is not.

29.3 Variants comparison

In the toolbox, the folder **Variants Comparison** 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.

29.3.1 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).

> **Toolbox** | **Resequencing Analysis** ( ) | **Variants 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 29.10.

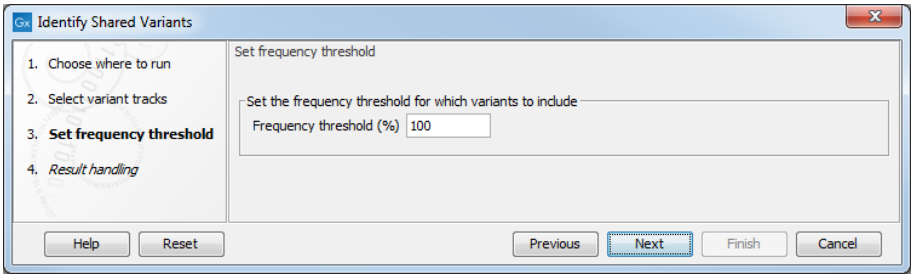


Figure 29.10: *Frequency treshold.*

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 29.10).
- **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.

29.3.2 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.

Toolbox | **Resequencing Analysis** () | **Variants Comparison** () | **Identify Enriched Variants in Case vs Control Samples** ()

In the first step of the dialog, you select the case variant tracks (figure 29.11).

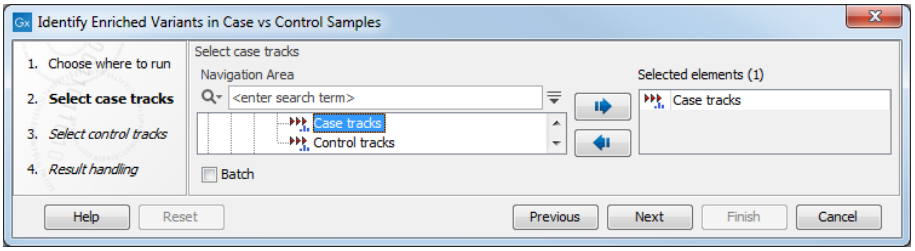


Figure 29.11: *Select the case variant track.*

Clicking **Next** shows the dialog in figure 29.12.

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 29.13), reporting only those variants with p-values lower than the threshold. All corrected and uncorrected p-values are shown here, so alternatively,

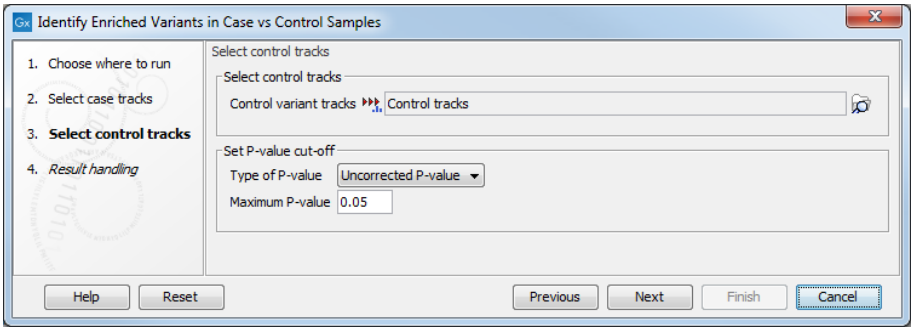


Figure 29.12: *In this dialog you can select the control tracks, a p-value correction method, and specify the p-value threshold for the fisher exact test.*

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.

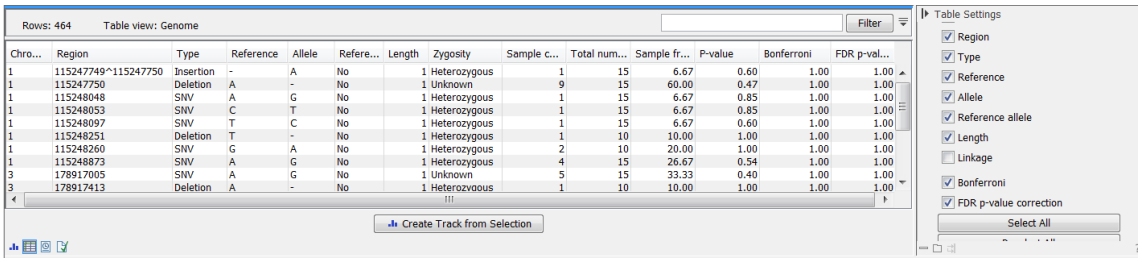


Figure 29.13: *In the output table, you can view information about all significant variants, select which columns to view, and filter manually on certain criteria.*

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.

29.3.3 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:

Toolbox | **Resequencing Analysis** ( ) | **Variants Comparison** ( ) | **Trio Analysis** ( )

In the first step of the dialog, select the variant track of the child. Clicking **Next** shows the dialog in figure 29.14.

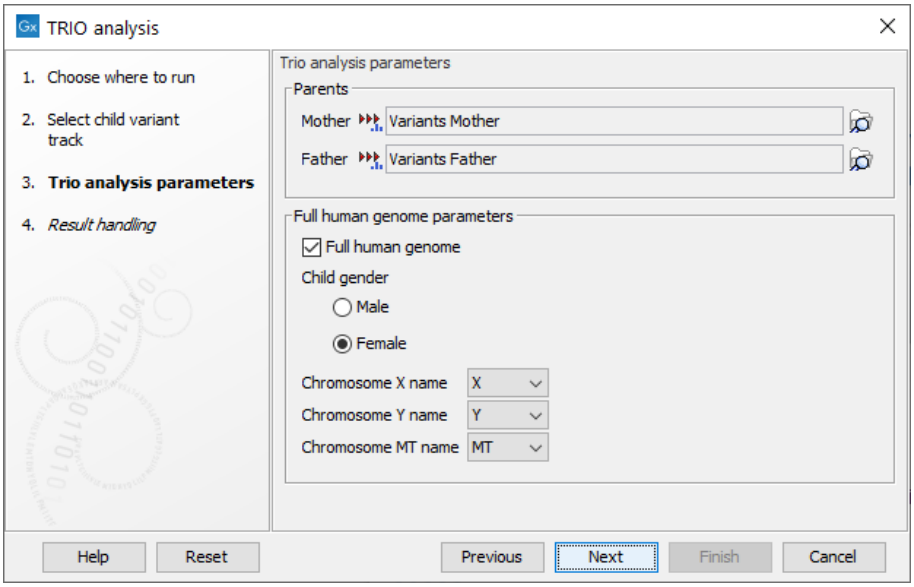


Figure 29.14: *Selecting variant tracks of the parents.*

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 29.15).

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 parents'.

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 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.

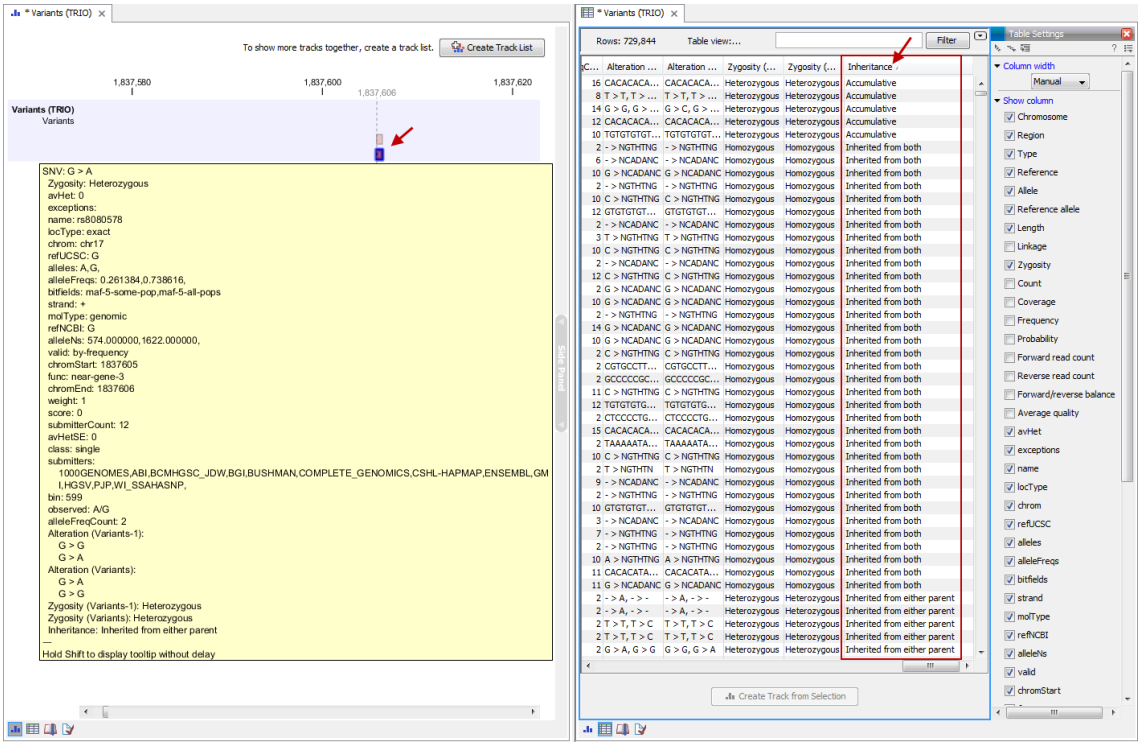


Figure 29.15: *Output from Trio Analysis showing the variants found in the child in track and table format.*

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.

Her 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)'. 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.

29.4 Variant quality control

29.4.1 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:

> **Toolbox** | **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 29.16 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

SNV	874,913
MNV	25,568
Insertion	79,281
Deletion	122,334
Replacement	5,365
Total (non-reference)	1,107,461

1 Variant types

	Before filtering	After filtering
SNV	874,913	226,417
MNV	25,568	5,604
Insertion	79,281	30,638
Deletion	122,334	37,759
Replacement	5,365	839
Total (non-reference)	1,107,461	301,257

Figure 29.16: *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 29.5.1). Figure 29.17 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

	Before filtering	After filtering
Non-exonic	956,961	263,268
Synonymous	39,809	14,891
Non-synonymous	110,691	23,098
Total (non-reference)	1,107,461	301,257

2 Amino acid changes

	Before filtering	After filtering
Non-exonic	956,961	N/A
Synonymous	39,809	N/A
Non-synonymous	110,691	N/A
Total (non-reference)	1,107,461	N/A

Figure 29.17: *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 29.5.2). Figure 29.18 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

	Before filtering	After filtering
Possible splice site disruption	9,051	1,572
No splice site disruption	1,098,410	299,685
Total (non-reference)	1,107,461	301,257

3 Splice site effect

	Before filtering	After filtering
Possible splice site disruption	N/A	1,572
No splice site disruption	N/A	299,685
Total (non-reference)	N/A	301,257

Figure 29.18: *Splice site effect sections with a filter track that contains annotations (left), and with an input track missing relevant annotations (right).*

29.5 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.

29.5.1 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:

Toolbox | **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 29.19).

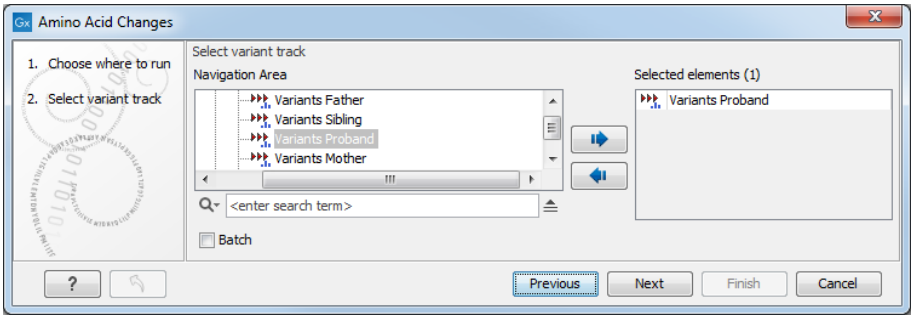


Figure 29.19: *The Amino Acid Changes annotation tool takes variant tracks as input.*

Click **Next** to go to the next wizard step (figure 29.20).

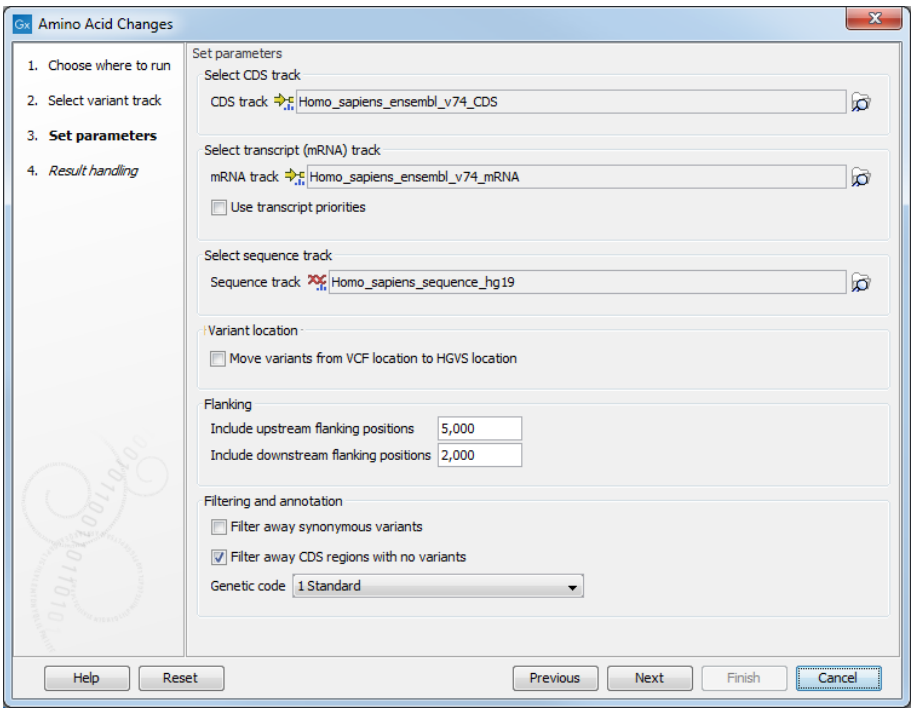


Figure 29.20: *Select CDS, mRNA, and sequence track and choose whether or not you would like to filter away synonymous variants.*

- **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". Transcripts without any priority are ignored; when adding c. and p. annotations, only the annotation from the highest priority transcript intersecting with the variant will be added. In cases where there are several "priority 1" transcripts, if two genes overlap a variant, only the highest priority transcript(s) will be reported, and this might be from just one of the genes. Note that tracks with priorities transcripts can be generated by modifying a gtf/gff file by adding 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, and in particular when working with downstream software such as Ingenuity Variant Analysis and QCI Interpret. This option does not affect the HGVS annotations added by the tool. Note that enabling this location 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.

- **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**.

  - The "Filter synonymous variants" option allows you to filter away synonymous variants that does not cause any change to the encoded amino acid.
  - By default, the tool will filter out CDS regions that have no variants. You can choose to include them in your amino acid annotation track by deselecting the "Filter CDs regions with no variants" option.
  - The genetic code is the code that is used for amino acid translation (see http://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 you 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 http://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 29.21.

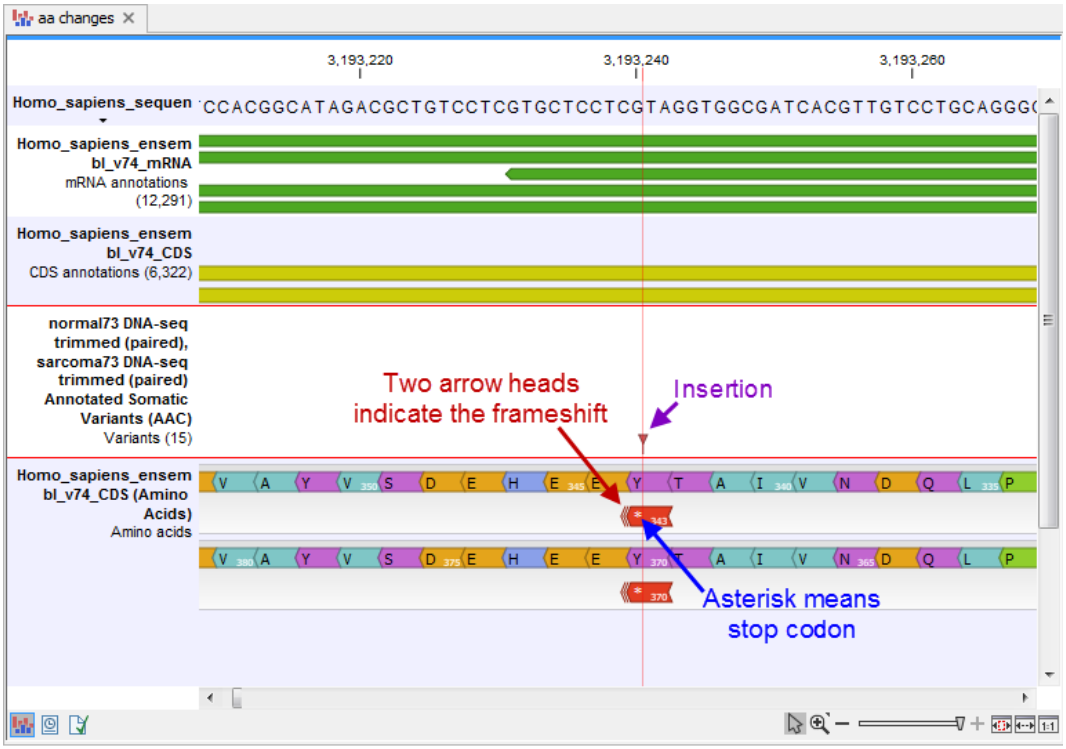


Figure 29.21: *The variant track and the amino acid track is here presented together with the reference sequence and the mRNA and CDS tracks. An insertion (purple arrow) has caused a frameshift (red arrow) that has changed an alanine to a stop codon (blue arrow).*

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 http://varnomen.hgvs.org/. 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 http://varnomen.hgvs.org/recommendations/protein/.

- **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.

An example of the output is given in figure 29.22.

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 29.23.

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.

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.

The amino acid track



Figure 29.22: *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.*

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 29.24).

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:

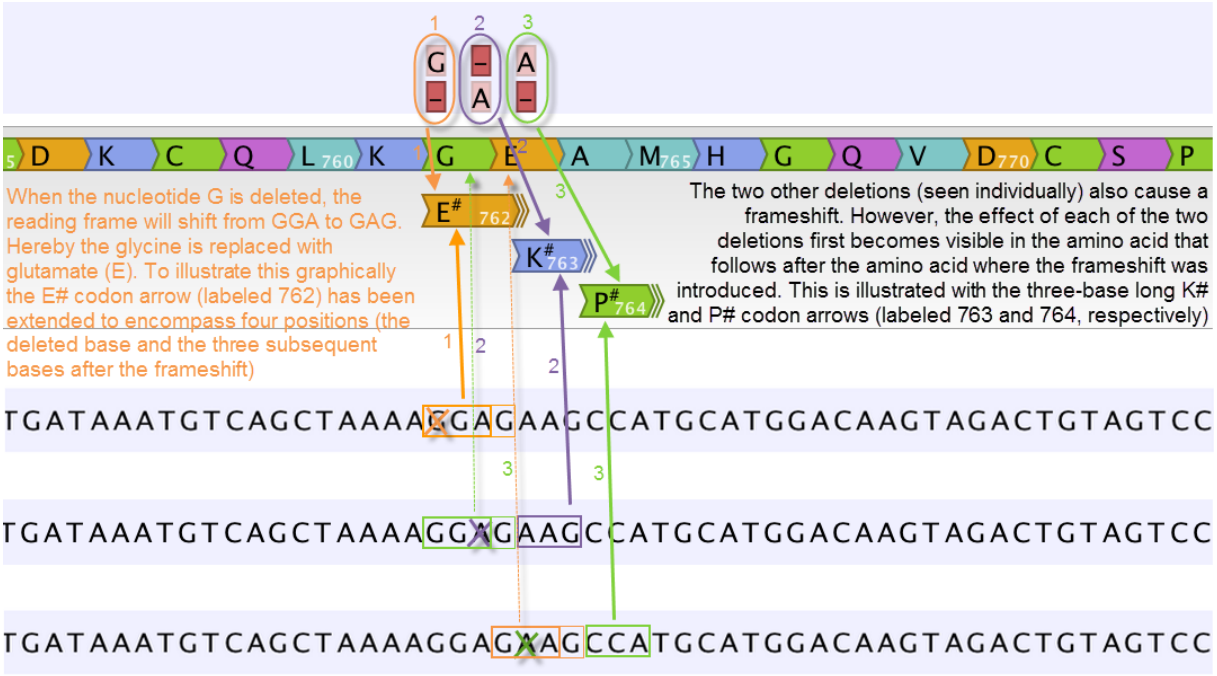


Figure 29.23: *Amino acids encoded from codons that potentially could have been affected by more than one variant are marked with a hash symbol (#) as the graphically presented amino acid changes always only include a single variant (a SNV, MNV, insertion, or deletion). Shown here are three different 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.*

  - **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`).

29.5.2 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 29.25) followed by a transcript track (see figure 29.26). As part of the dialog you can choose to exclude all variants that do not fall within a splice site.

If a variant falls within two base pairs of an intron-exon boundary, it will be annotated as a possible splice site disruption (see figure 29.27).

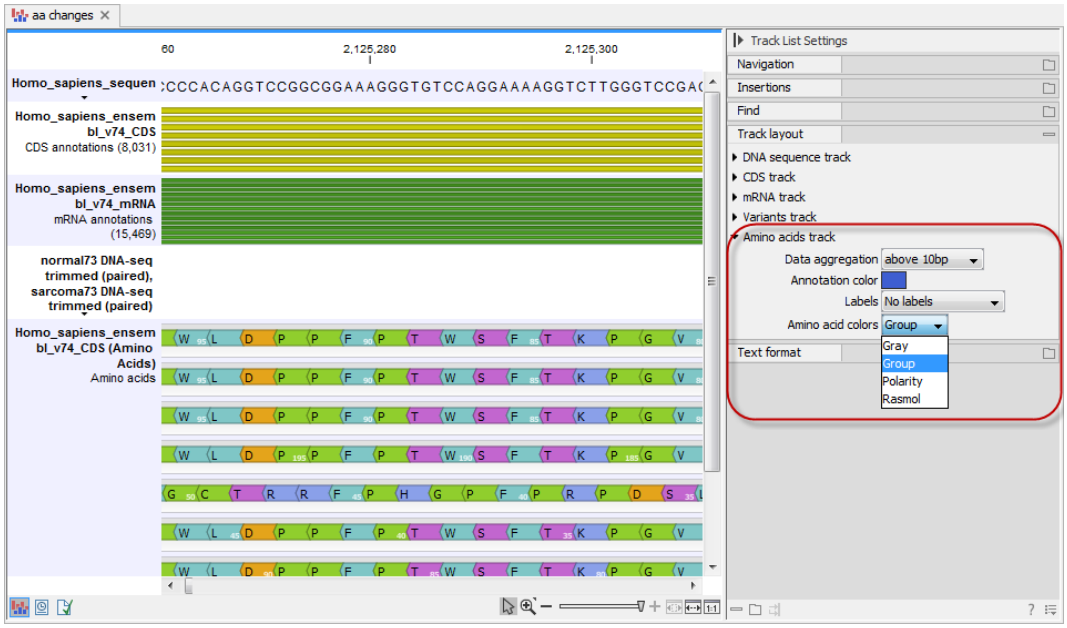


Figure 29.24: *The colors of the amino acids can be changed in the Side Panel under "Amino acids track".*

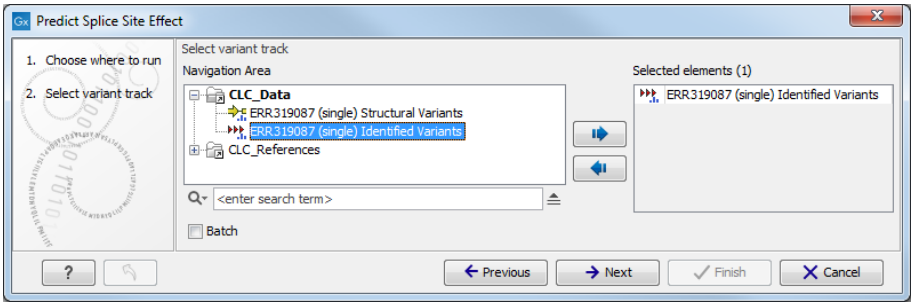


Figure 29.25: *Variant track selection.*

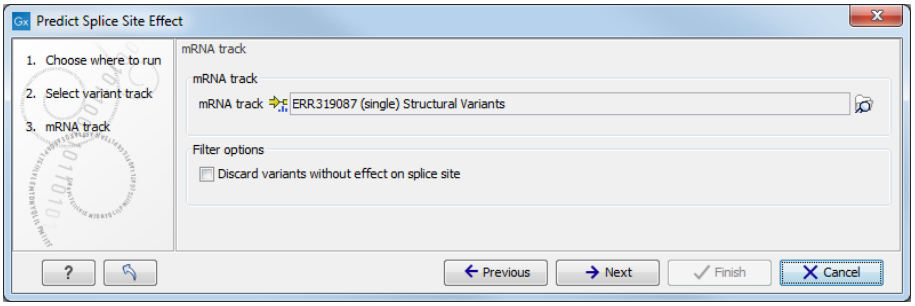


Figure 29.26: *Transcript track selection.*

29.5.3 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 associ-

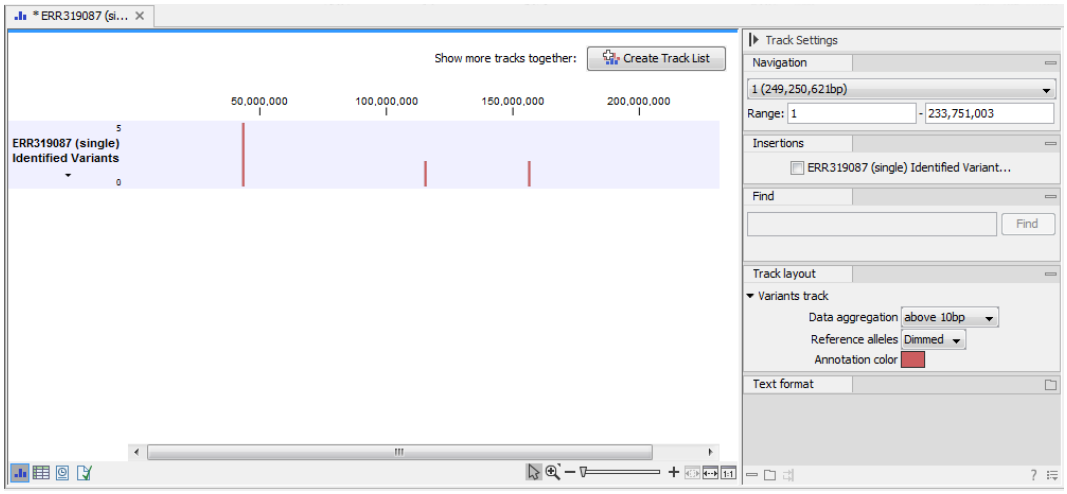


Figure 29.27: *Results of the Predict Splice Site Effect tool.*

ated Gene Ontology terms and can be downloaded that from the Gene Ontology web site for various species (http://www.geneontology.org/GO.downloads.annotations.shtml). Download the GO Annotations of the relavant species by clicking on the *.gz link in the table (see figure 29.28). Import the downloaded annotations into the workbench using **Import | Standard Import**.

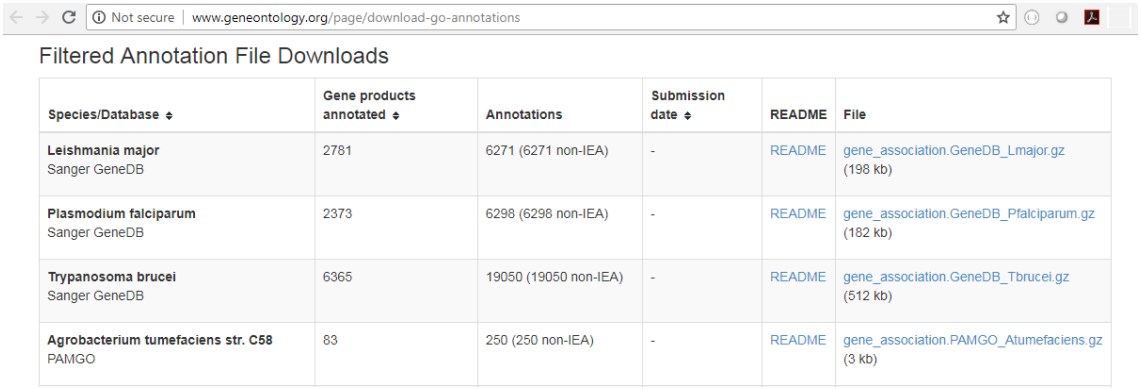


Figure 29.28: *Download the GO Annotations by clicking on the *.gz link in the table.*

You will also need a Gene track for the relevant species (learn more about gene tracks in section 24.1).

To run the GO Enrichment analysis, go to the toolbox:

> **Toolbox | 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 29.29).

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.

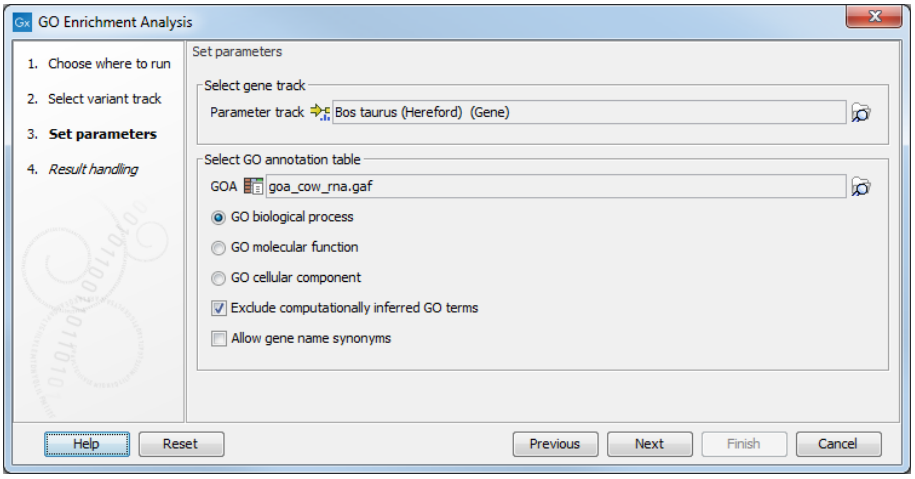


Figure 29.29: *The GO enrichment settings.*

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 29.30). 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.

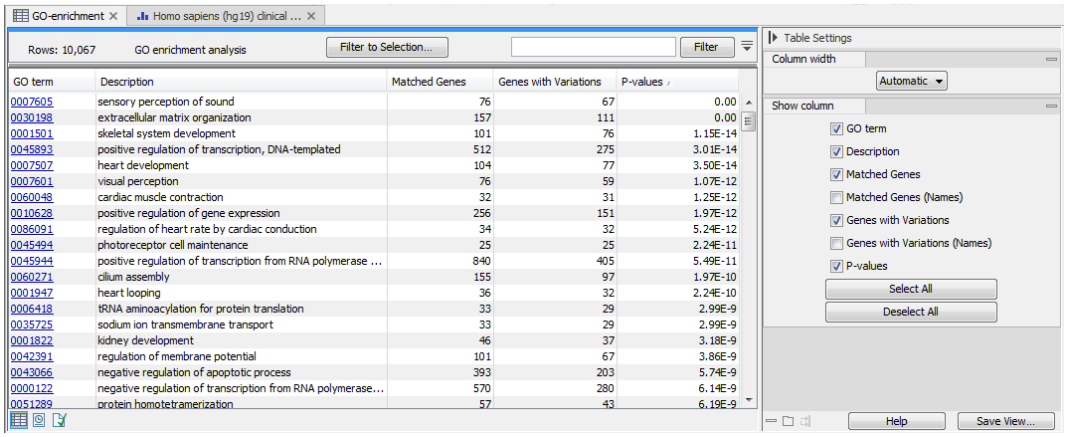


Figure 29.30: *The results of the analysis.*

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.

29.5.4 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 (http://www.wwpdb.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:

Toolbox | **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 29.31).

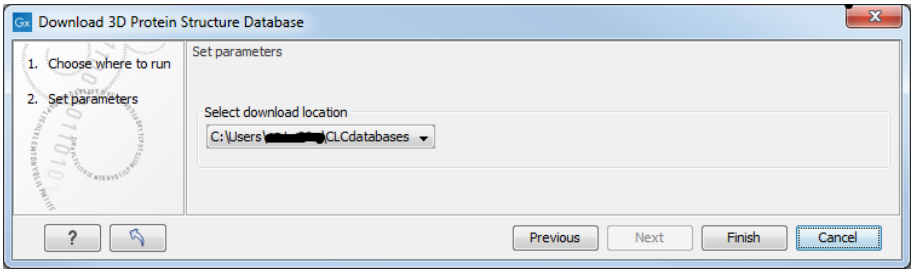


Figure 29.31: *Select the download location.*

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** ( )

29.5.5 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 29.5.5 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 29.5.5 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 section 29.5.4).

To run the tool, select:

> **Toolbox** | **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 29.32).

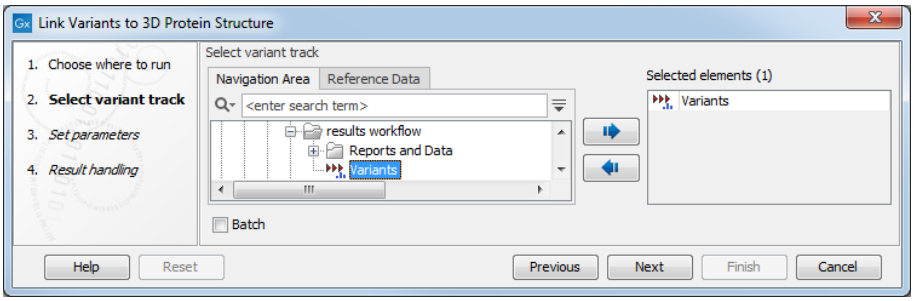


Figure 29.32: *Select the variant track holding the variants that you would like to visualize on 3D protein structures.*

Click **Next**. In the next wizard step, you must provide a CDS track and the reference sequence track (figure 29.33).

If you have not already downloaded a CDS and a reference sequence track, this can be done using the Reference Data Manager (see section 8.1).

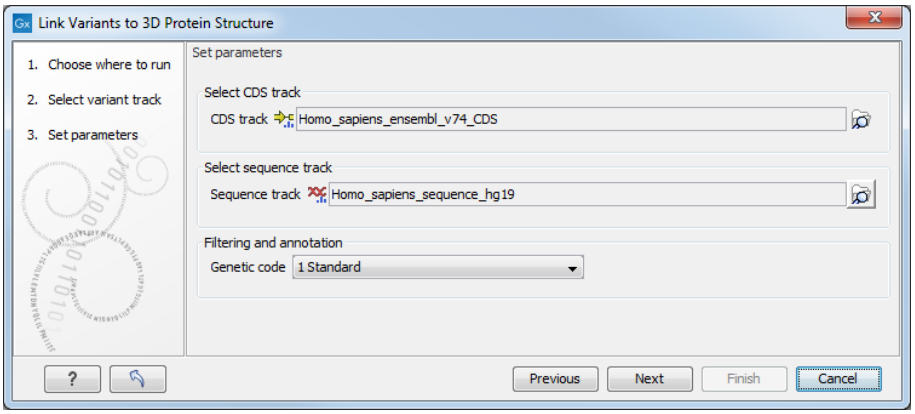
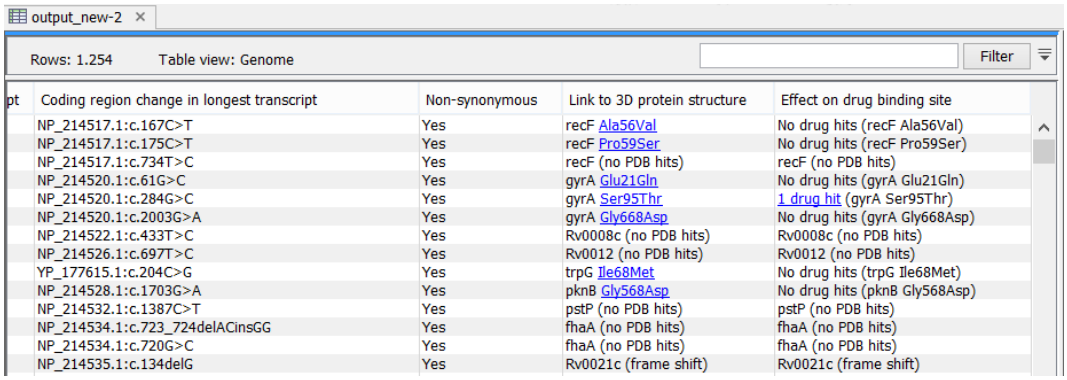


Figure 29.33: *Select CDS and reference sequence.*

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 29.34). 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 29.34: *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 29.5.4) 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 17.6.2).

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 29.5.5). 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 section 29.5.5.

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 29.34).
- '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 29.35).

  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.

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 - section 29.5.5).
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).

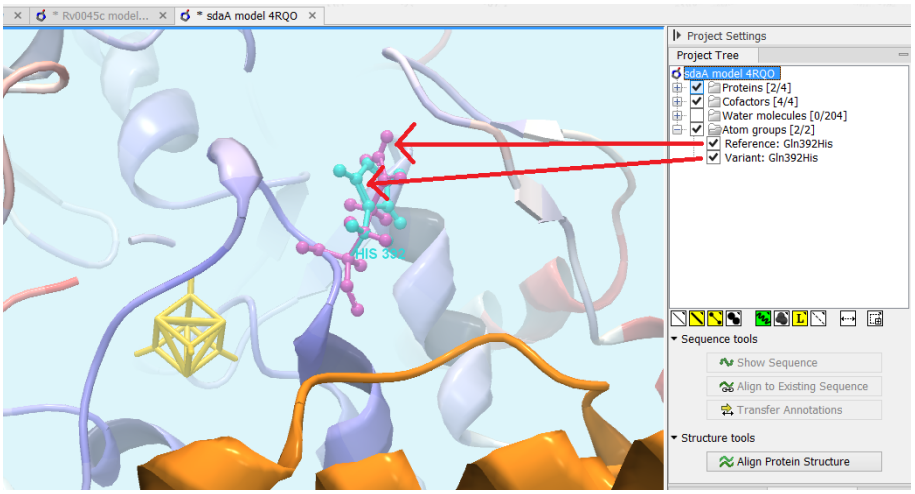


Figure 29.35: *Generated 3D view of variant. The reference amino acid is seen in purple and the variant in cyan on top of each other. Only the backbone of protein structures are visualized by default. The modeled protein structure is colored to indicate local model uncertainty - red for flexible and 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.*

4. **Create a model structure for the reference** by mapping it onto the template structure based on the sequence alignment (see section 17.6.2).

5. **Create a model structure with variant(s)** by mapping the protein sequence with the variant consequences onto the reference structure model (see section 17.6.2).

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 29.35), to indicate local model uncertainty (see section 17.6.2). 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 29.35).

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 29.35). 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 14.3.

Infobox: Biomolecules in *CLC Genomics Workbench*

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 29.36).

Rows: 7 Summary of hits found Filter

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
4KPF			STREPTOCOCCUS PNEUMONIAE	NOVEL FLUOROQUINOLONES IN COMPLEX WITH TOPOISOMERASE IV FROM S. PNEUMONIAE AN
3FOE	MOXIFLOXACIN (AVELOX)	DB00218	STREPTOCOCCUS PNEUMONIAE	STRUCTURAL INSIGHT INTO THE QUINOLONE-DNA CLEAVAGE COMPLEX OF TYPE IIA TOPOISON
3FOE	CLINAFLOXACIN		STREPTOCOCCUS PNEUMONIAE	STRUCTURAL INSIGHT INTO THE QUINOLONE-DNA CLEAVAGE COMPLEX OF TYPE IIA TOPOISON
3QX3	ETOPOSIDE (EPOSIN)	DB00773	HOMO SAPIENS	HUMAN TOPOISOMERASE IIBETA IN COMPLEX WITH DNA AND ETOPOSIDE
4G0V	MITOXANTRONE (MITOX)	DB01204	HOMO SAPIENS	HUMAN TOPOISOMERASE IIBETA IN COMPLEX WITH DNA AND MITOXANTRONE
4G0W	AMETANTRONE		HOMO SAPIENS	HUMAN TOPOISOMERASE IIBETA IN COMPLEX WITH DNA AND AMETANTRONE

Figure 29.36: *An example of a drug hit table with information about the drug and with links to 3D visualizations of variant-drug interaction.*

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 29.5.5, 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 29.37).
- **PDB drug name.** (Hidden by default) The identifier used by PDB for the ligand or drug.
- **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 PBD hit.

To learn more about model structure, see section 17.6.2.

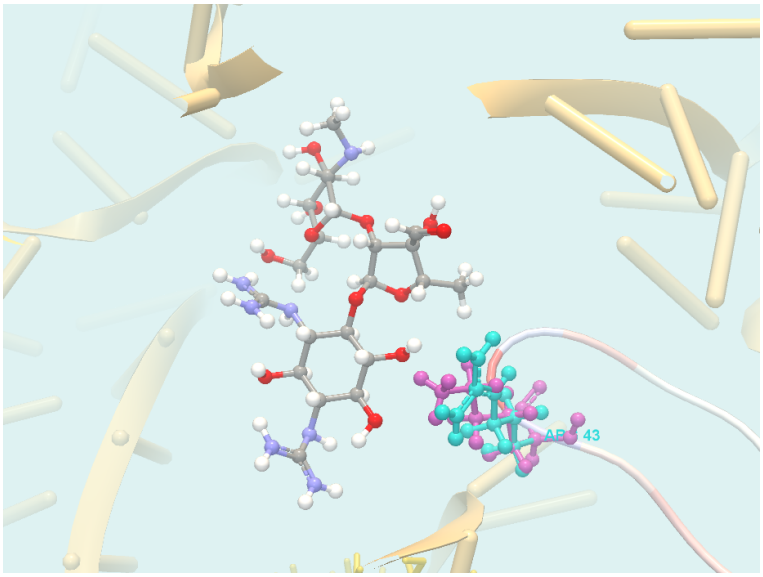


Figure 29.37: *3D visualization of variant-drug interaction. The drug (streptomycin) is in center of the view, and colored according to it's atom types. The variant is visualized as in figure 29.35. In this picture, the foreground has been cut away using the clipping plane functionality found in the Visualization tab in the Side Panel, to better see the variant-drug interaction.*

Chapter 30

RNA-seq and Small RNA analysis

Contents

30.1 RNA-seq normalization . . . 802
30.2 RNA-Seq Analysis . . . 802
30.2.1 Reads and reference settings . . . 803
30.2.2 Mapping settings . . . 805
30.2.3 The EM estimation algorithm . . . 807
30.2.4 Expression settings . . . 809
30.2.5 Output settings . . . 812
30.2.6 RNA-Seq result handling . . . 813
30.2.7 Expression tracks . . . 814
30.2.8 RNA-seq reads track . . . 818
30.2.9 RNA-Seq report . . . 819
30.2.10 Gene fusion reporting . . . 825
30.3 PCA for RNA-Seq . . . 825
30.3.1 Principal component analysis plot (2D) . . . 826
30.3.2 Principal component analysis plot (3D) . . . 828
30.4 Differential Expression . . . 829
30.4.1 The GLM model . . . 830
30.4.2 Differential Expression in Two Groups . . . 832
30.4.3 Differential Expression for RNA-Seq . . . 833
30.4.4 Filtering on average expression . . . 836
30.4.5 Output of the Differential Expression tools . . . 837
30.5 Create Heat Map for RNA-Seq . . . 841
30.5.1 Clustering of features and samples . . . 841
30.5.2 The heat map view . . . 844
30.6 Create Expression Browser . . . 845
30.6.1 The expression browser . . . 847
30.7 Create Venn Diagram for RNA-Seq . . . 849
30.7.1 Venn diagram table view . . . 850
30.8 Gene Set Test . . . 851
30.8.1 Tool output and GAF file comparison . . . 854

30.9 miRNA analysis 855
30.9.1 Quantify miRNA 855
30.9.2 Quantify miRNA outputs 858
30.9.3 Naming isomiRs 862
30.9.4 Annotate with RNAcentral Accession Numbers 863
30.9.5 Create Combined miRNA Report 863
30.9.6 Extract IsomiR Counts 864

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 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 11.5.

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 the Create Heat Map and PCA for RNA-Seq tools.

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 10.

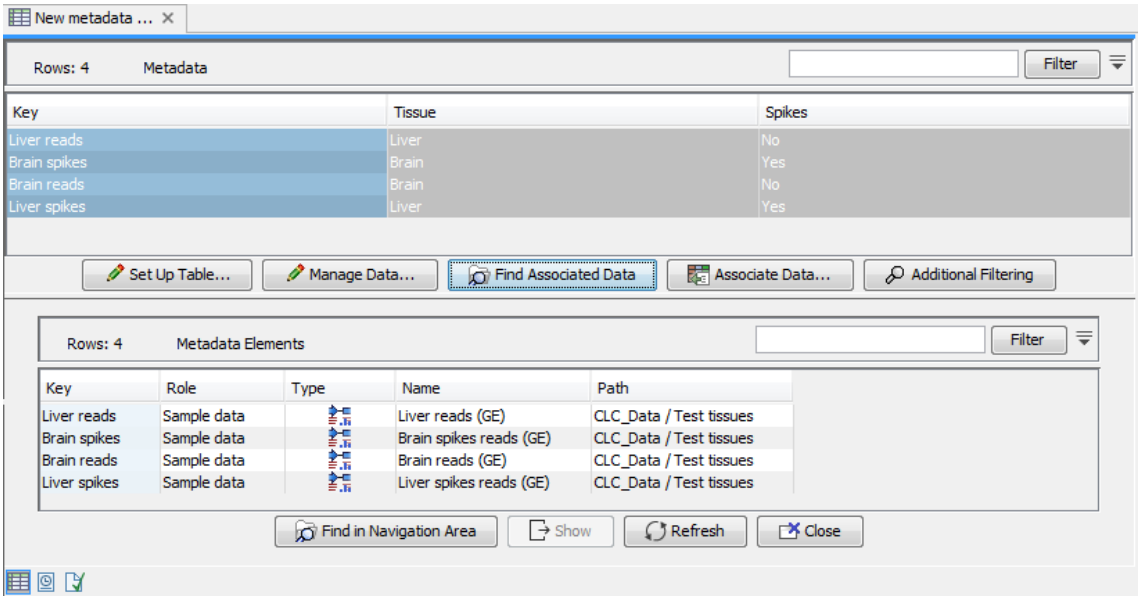


Figure 30.1: *A metadata table with expression samples associated with it.*

30.1 RNA-seq normalization

TMM Normalization Since the sequencing depth might differ between samples, a per-sample library size normalization must be performed before samples can be compared. In the case of the tools included in the RNA-Seq folder, this normalization is automatically applied by the tools.

For the RNA-Seq tools that compare samples (PCA for RNA-Seq, Create Heat Map for RNA-Seq, Differential Expression for RNA-Seq and Create Expression Browser), library size normalization is automatically performed using the TMM (trimmed mean of M values) method of Robinson and Oshlack, 2010 (see https://genomebiology.biomedcentral.com/articles/10.1186/gb-2010-11-3-r25). Libraries sizes are then used as part of the per-sample normalization. TMM normalization is the normalization used in edgeR [Robinson et al., 2010].

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.

logCPM After TMM factors are calculate for each sample, we calculate the TMM-adjusted log CPM counts (similar to the EdgeR approach):

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 Finally, we perform cross-sample normalization. For each row (gene/-transcript), a Gaussian normalization (Z-Score normalization) is applied: data is shifted so that mean is zero, and the standard deviation one.

30.2 RNA-Seq Analysis

The following describes the overall process of the RNA-Seq analysis when using an annotated eukaryote genome (see section 30.2.1 for more information on other types of reference data).

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.

An example is shown in figure 30.2.

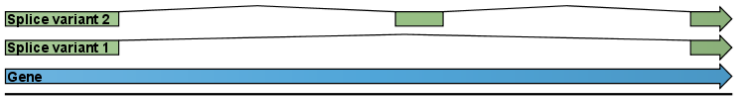


Figure 30.2: *A simple gene with three exons and two splice variants.*

This is a simple gene with three exons and two splice variants. The transcripts are extracted as shown in figure 30.3.

Next, the reads are mapped against all the transcripts, and to the whole genome. For more information about the read mapper, see section 27.1.

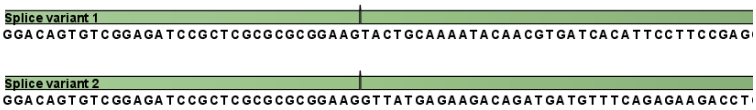


Figure 30.3: *All the exon-exon junctions are joined in the extracted transcript.*

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.

30.2.1 Reads and reference settings

To start the RNA-Seq analysis, go to:

Toolbox | **RNA-Seq Analysis (**  **)** | **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 6.3.

If you have several samples that you wish to analyze independently and compare afterwards, you can run the analysis in batch mode (see section 9.3).

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 30.4.

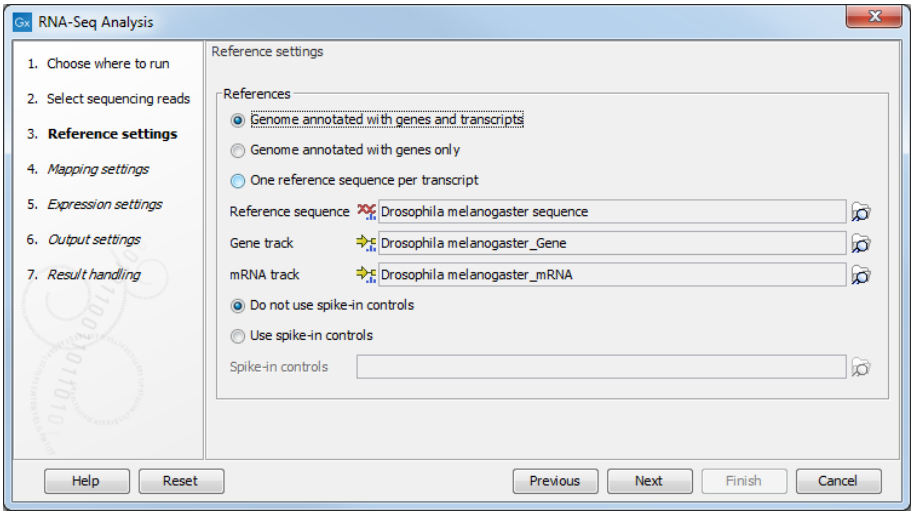


Figure 30.4: *Defining a reference genome for RNA-Seq.*

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 30.2). 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 be can obtained in various ways:

- Directly downloaded as tracks using the Reference Data Manager (see section 8.1).
- Imported as tracks from fasta and gff/gtf files (see section 6.2)
- Imported from Genbank or EMBL files and converted to tracks (see section 24.7).
- Downloaded from Genbank (see section 7.1) and converted to tracks (see section 24.7).

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 30.2.1). 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 34.2).
- In cases where the input reads are paired-end, choose the option "Count paired reads as two" in the Expression level options 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 6.4.

30.2.2 Mapping settings

When the reference has been defined, click **Next** and you are presented with the dialog shown in figure 30.5.

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 27.1.3.

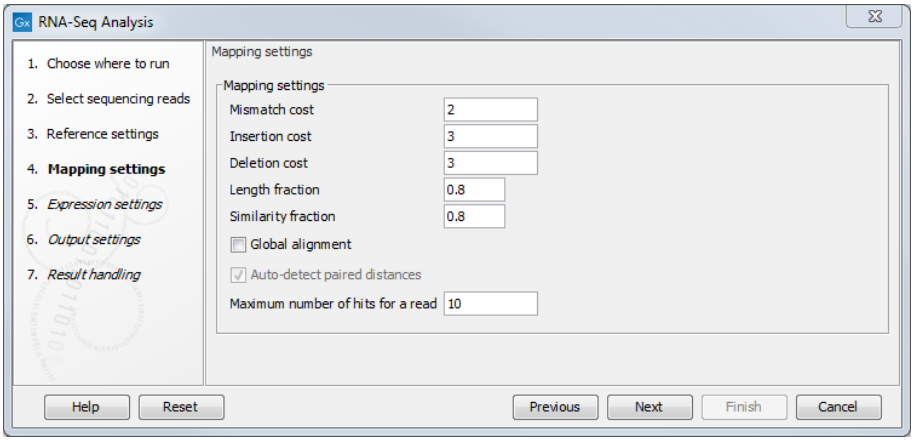


Figure 30.5: *Defining mapping parameters for RNA-Seq.*

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 30.2.3). 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.

30.2.3 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 30.6). 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.

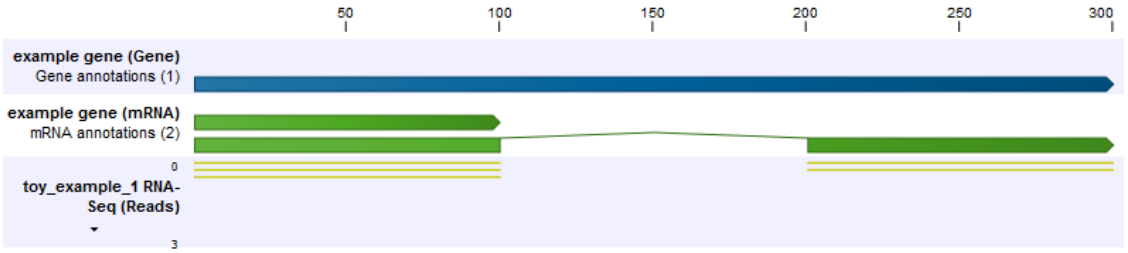


Figure 30.6: *The longer transcript has twice the abundance, but four times the number of reads as the shorter transcript.*

In the second example, the set up is the same, but now the shorter transcript is twice as abundant as the longer transcript (figure 30.7). Because the longer transcript is twice as long, there are equal numbers of reads from each transcript.

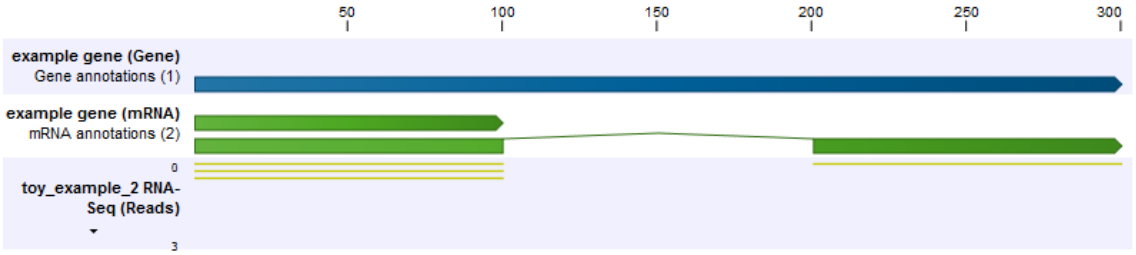


Figure 30.7: *The longer transcript has half the abundance, but the same number of reads as the shorter transcript.*

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.

30.2.4 Expression settings

When the reference has been defined, click **Next** and you are presented with the dialog shown in figure 30.8.

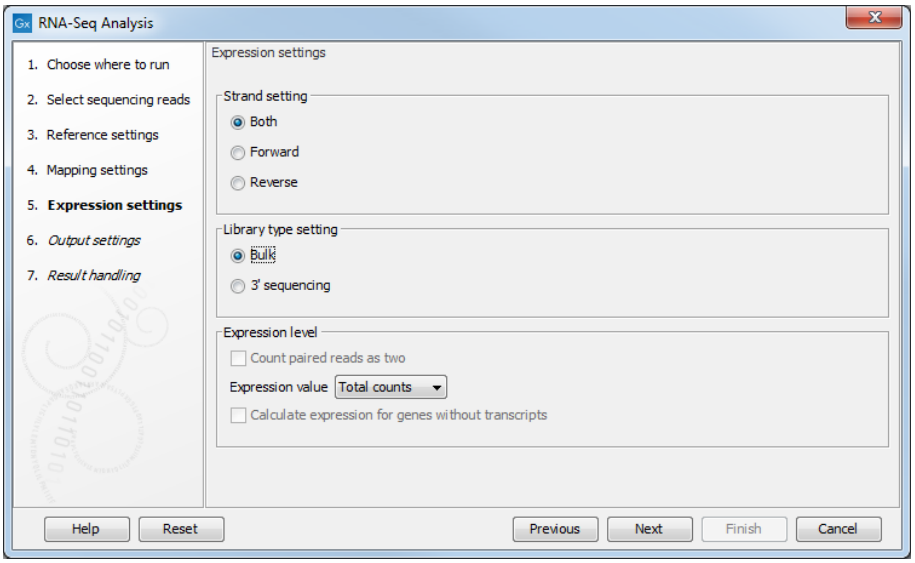


Figure 30.8: *Defining how expression values should be calculated.*

These parameters determine the way expression values are counted.

Strand setting When this option is checked, the user can specify whether the reads should be mapped in the same orientation as the transcript from which they originate (forward) or in the reverse direction (reverse). This will typically be appropriate when a strand specific protocol for read generation has been used. It 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 running RNA seq with the strand specific option turned on you can only make use of pairs in forward-reverse orientation, meaning that mate pairs are not supported.

Library type setting

- **Bulk**. Use for reads expected to be uniformly distributed across the full length of transcripts. This is the default.
- **3' sequencing**. Use for reads expected to be biased towards the 3' end of transcripts. When this option is selected:
  - 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 (reads in gene) / (total 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 UMIs by tools of the Biomedical Genomics Analysis plugin:
  - Expression values in the GE track are based on the number of distinct UMIs for each gene, rather than the number of reads.
  - Values reported in the "Distribution of biotypes" section of RNA-seq reports are calculated based on the number of distinct UMIs for each gene. Other values in the report are as described in section 30.2.9.

Count paired reads as two 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 three 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]
- It is possible to detect **Gene fusions** when one read in a pair maps in one gene and the other part maps in another gene. Several reads exhibiting the same pattern supports the presence of a fusion gene.

You can read more about how paired data are imported and handled in section 6.3.7.

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. The default behavior of the *CLC Genomics Workbench* is to count fragments (FPKM) rather than individual reads when two reads map as an intact pair. That is, an intact pair is given a count of one. Reads from a pair are considered part of 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. Neither member of a broken pair is counted when the default counting scheme is used. The reasoning is that when reads map as a broken pair, it is an indication that something is not right. For example, perhaps the

[1] Note that the *CLC Genomics Workbench* only calculates the expression of the transcripts already annotated on the reference.

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 as is done using the default counting scheme. 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 the **Count paired reads as two** option, you choose to count mapped 'reads' (RPKM) rather than mapped 'fragments' (FPKM). That means that, 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 given a count of one. Single mapped reads are also given a count of one. Note that this approach does not represent the abundance of fragments being sequenced correctly, since the two reads of a pair derive from the same fragment, whereas a fragment sequenced with single reads only give rise to one read.

Note that whether you choose to calculate RPKM or FPKM, the value will be given in a column called "RPKM" for all subsequent analysis.

Expression value 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 30.4). The content of the result files is described in section 30.2.6.

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 30.2.2).
- **TPM**. (Transcripts per million). This is computed as $\frac{\text{RPKM} \cdot 10^6}{\sum \text{RPKM}}$, 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 30.2.4).

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]:

$$RPKM = \frac{total\ exon\ reads}{mapped\ reads(millions) \times exon\ length\ (KB)}.$$

For prokaryotic genes and other non-exon based regions, the calculation is performed in this way:

$$RPKM = \frac{total\ gene\ reads}{mapped\ reads(millions) \times gene\ length\ (KB)}.$$

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 30.2.4.

30.2.5 Output settings

Click **Next** and you are presented with the dialog shown in figure 30.9. The parameter is enabled when using paired data.

The **Minimum read count fusion gene table** parameter ensures that only combinations of genes supported by at least this number of read pairs are included. The default value is 5, which means that at least 5 pairs need to connect two genes in order to report it in the result (see section 30.2.10).

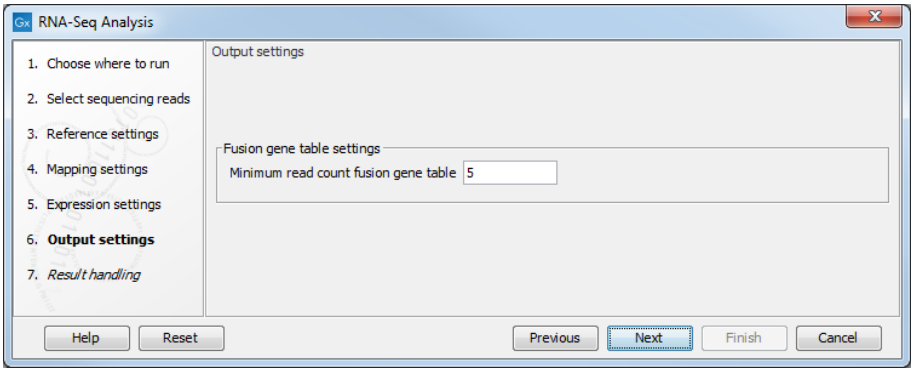


Figure 30.9: *Fusion genetable settings.*

30.2.6 RNA-Seq result handling

Clicking **Next** will allow you to specify the output options as shown in figure 30.10.

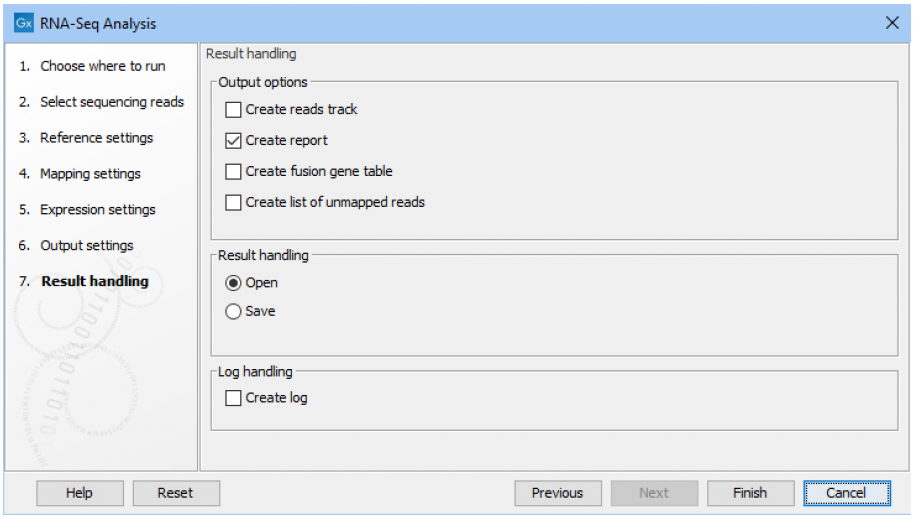


Figure 30.10: *Selecting the output of the RNA-Seq analysis.*

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 30.2.9 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 fusion gene table**. This option is enabled when using paired data, and will create a table that lists potential fusion genes (see section 30.2.10).
- **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 30.2.2). If you started with paired reads, then more than one list of unmapped 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).

30.2.7 Expression tracks

Both tracks can be shown in a **Table** ( ) and a **Graphical** ( ) view.

The expression track table view has the following options (figure 30.11).

- 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.

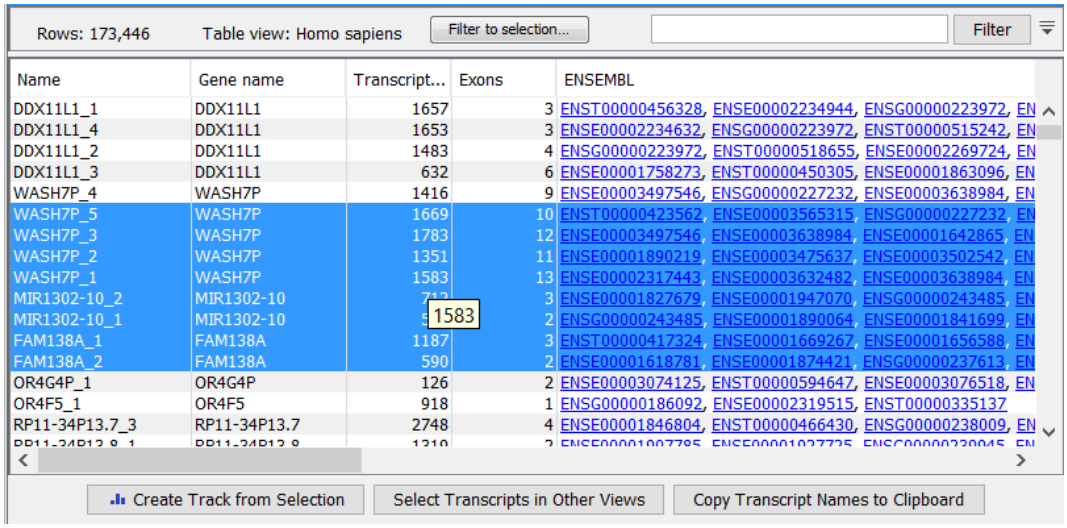


Figure 30.11: *RNA-Seq results shown in a table view.*

By creating a **Track list**, the graphical view can be shown together with the read mapping track and tracks from other samples:

File | **New** | **Track List ( )**

Select the mapping and expression tracks of the samples you wish to visualize together and select the annotation tracks used as reference for the RNA-Seq and click **Finish**.

Once the track list is shown, double-click the label of the expression track to show it in a table view.

Clicking a row in the table makes the track list view jump to that location, allowing for quick inspection of interesting parts of the RNA-Seq read mapping (see an example in figure 30.12).

Reads spanning two exons are shown with a dashed line between each end as shown in figure 30.12, and the thin solid line represents the connection between two reads in a pair.

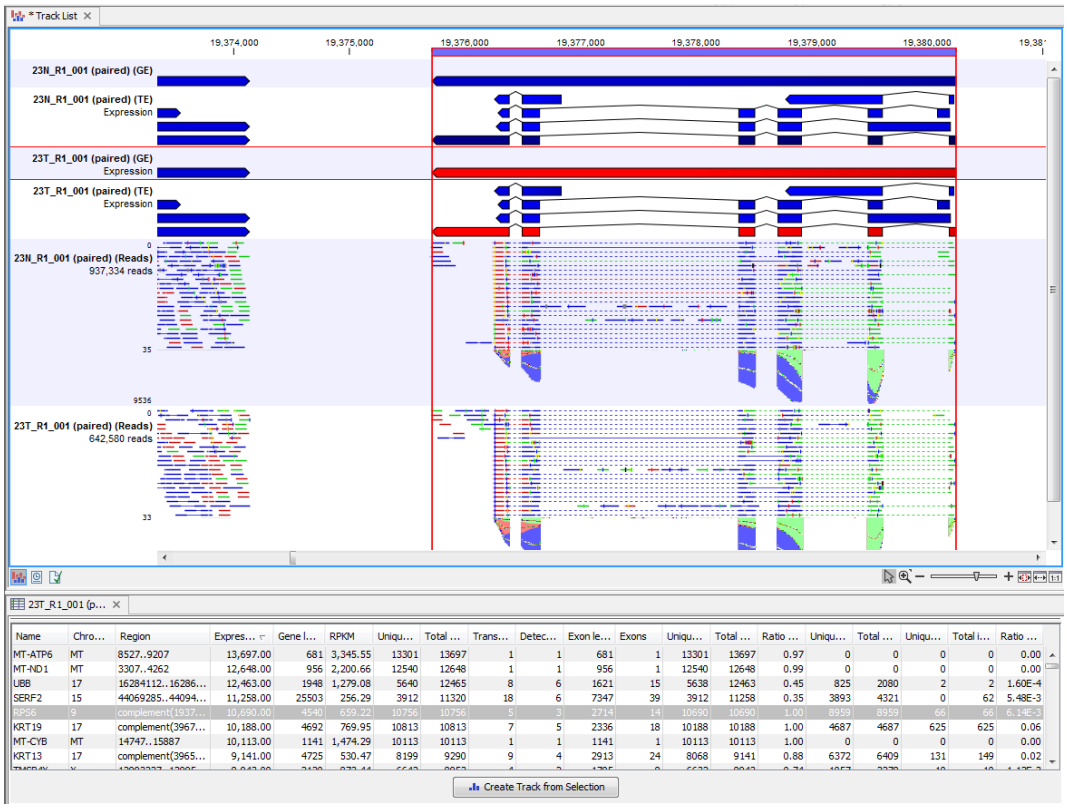


Figure 30.12: *RNA-Seq results shown in a split view with an expression track at the bottom and a track list with read mappings of two samples at the top.*

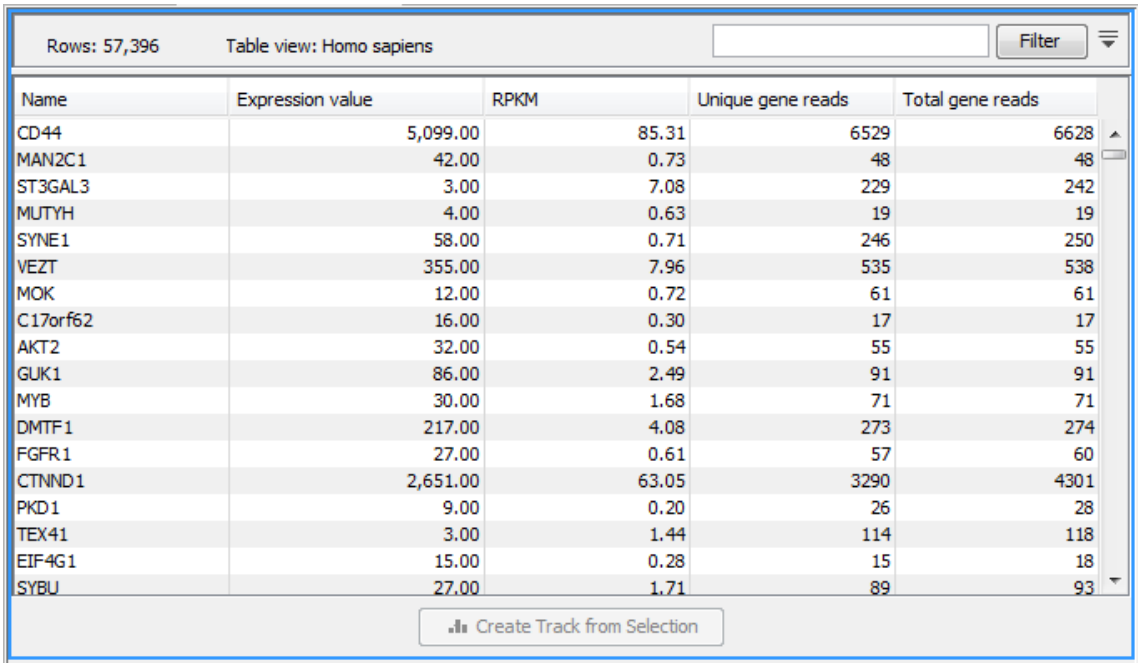
When doing comparative analysis, double click on one of the Expression or Statistical Comparison tracks in a track list to get its table view. Then click on the "Select genes in other views" button in any other table or expression browser will cause the track list to zoom to the selected gene.

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 24.8.2.

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 30.13 for an example subset of an expression track).

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.



Rows: 57,396 Table view: Homo sapiens Filter

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

Create Track from Selection

Figure 30.13: *A subset of a result of an RNA-Seq analysis on the gene level. Not all columns are shown in this figure*

- **Expression value**. This is based on the expression measure chosen as described in section 30.2.4.
- **Gene length** The length of the gene as annotated.
- **TPM (Transcripts per million)**. This is computed as $\frac{\text{RPKM} \cdot 10^6}{\sum \text{RPKM}}$, where the sum is over the RPKM values of all genes/transcripts (see http://bioinformatics.oxfordjournals.org/content/26/4/493.long).
- **RPKM**. This is the expression value measured in RPKM [Mortazavi et al., 2008]: $RPKM = \frac{\textit{total exon reads}}{\textit{mapped reads(millions)} \times \textit{exon length (KB)}}$. See section 30.2.4 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 30.12). 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 30.12). 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 30.4, 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 30.2.4.
- **TPM (Transcripts per million)**. This is computed as $\frac{\text{RPKM} \cdot 10^6}{\sum \text{RPKM}}$, where the sum is over the RPKM values of all genes/transcripts (see http://bioinformatics.oxfordjournals.org/content/26/4/493.long).
- **RPKM**. This is the expression value measured in RPKM [Mortazavi et al., 2008]: $RPKM = \frac{\textit{total exon reads}}{\textit{mapped reads(millions)} \times \textit{exon length (KB)}}$. See section 30.2.4 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 30.2.3).
- **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.

30.2.8 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 24 and section 27.2.2.

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 30.14 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 **Disconnect paired reads** box, within the Read Mapping Settings bar on the right-hand side of the track.

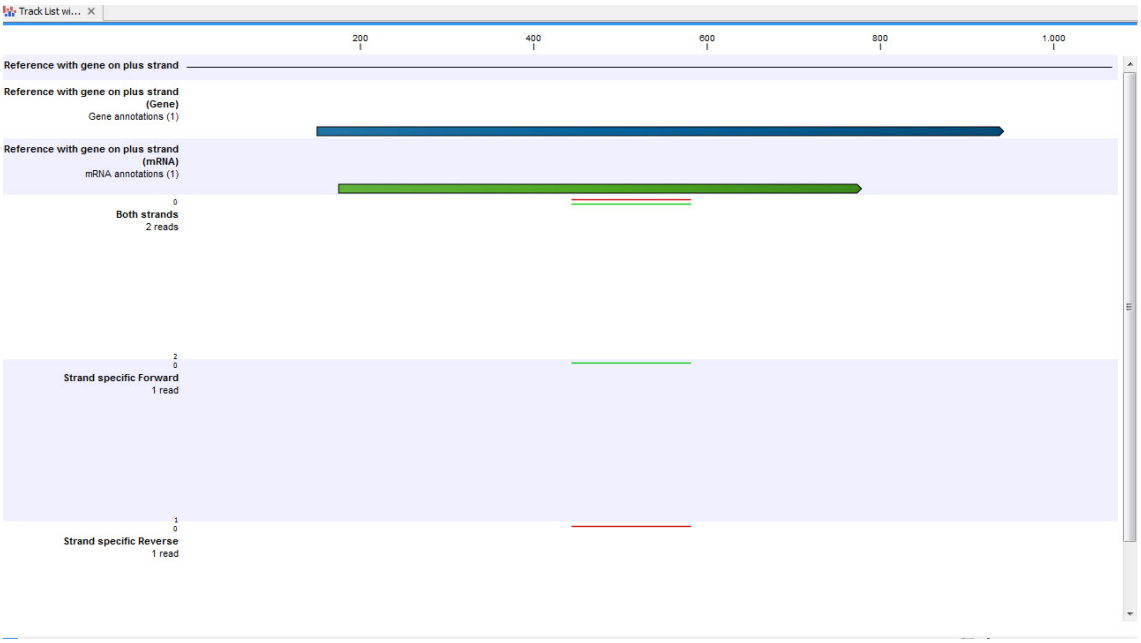


Figure 30.14: *A track list showing a gene and transcript on the plus strand, and various mapping results. The first reads track shows a mapping of two reads (one 'forward' and one 'reverse') using strand specific 'both' option. Both reads map successfully; the forward read colored green (because it 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.*

30.2.9 RNA-Seq report

An example of an RNA-seq report generated if you choose the **Create report** option is shown in

figure 30.15.

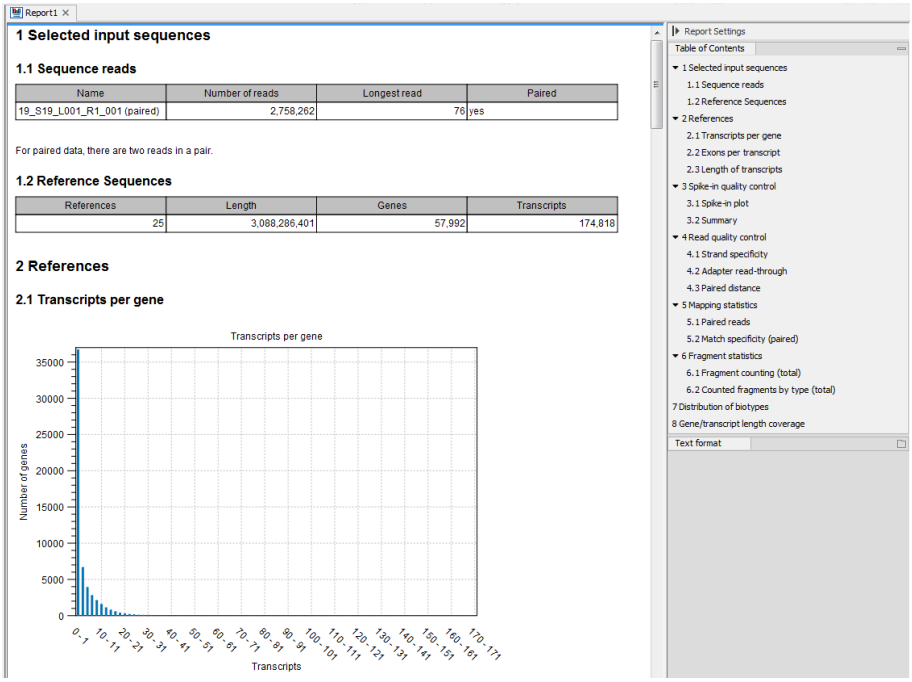


Figure 30.15: *Report of an RNA-Seq run.*

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.

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 30.16 to see an optimal spike-in plot).
- **Summary table**. A table provides more details on the spike-in detection. Figure 30.17 shows a failed spike-in control, with a table where results that require attention are highlighted in pink.

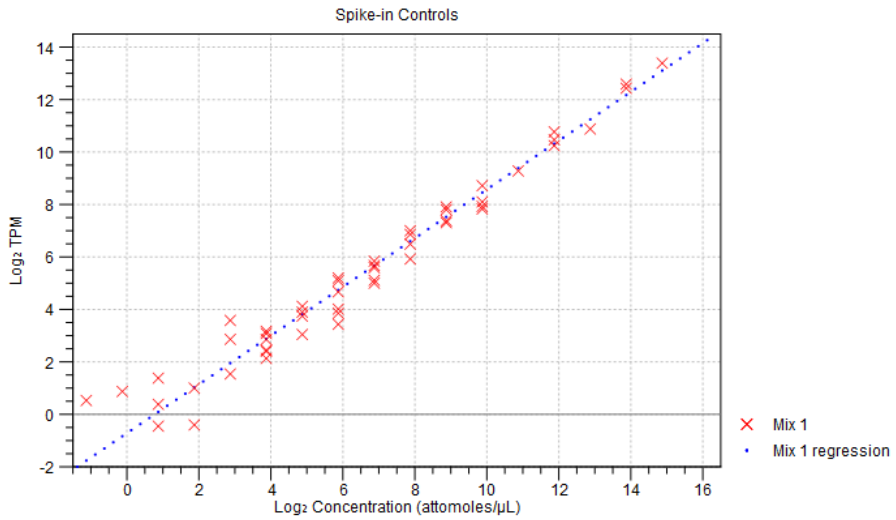
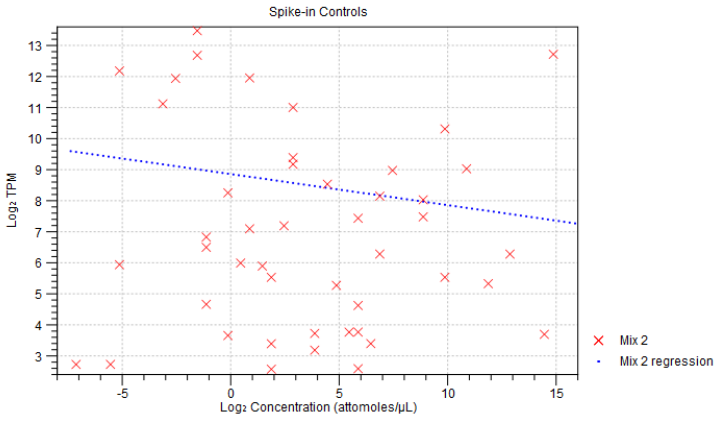


Figure 30.16: *Spike-in plot showing how the points fall close to the regression line at high concentration.*



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 30.17: *Summary table where less than optimal results are highlighted.*

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 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. 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 also be trimmed. In future experiments, consider selecting fragments that are longer than the read size.

- 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, 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 30.5. 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).
- **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 junctionas specified in figure 30.12 .
  - **Intergenic**. All reads that map partly or entirely between genes.
  - **Total**. Total amount of reads for a particular type.

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 30.18).

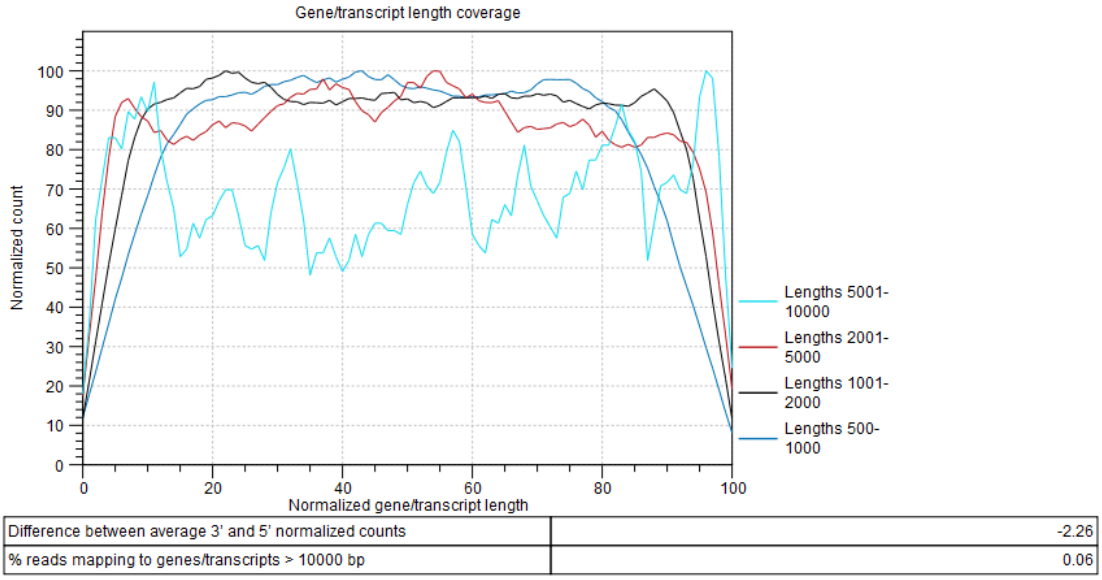


Figure 30.18: *Gene/transcript length coverage plot.*

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.

30.2.10 Gene fusion reporting

When using paired data, there is also an option to create an annotation track summarizing the evidence for gene fusions. An example is shown in figure 30.19.

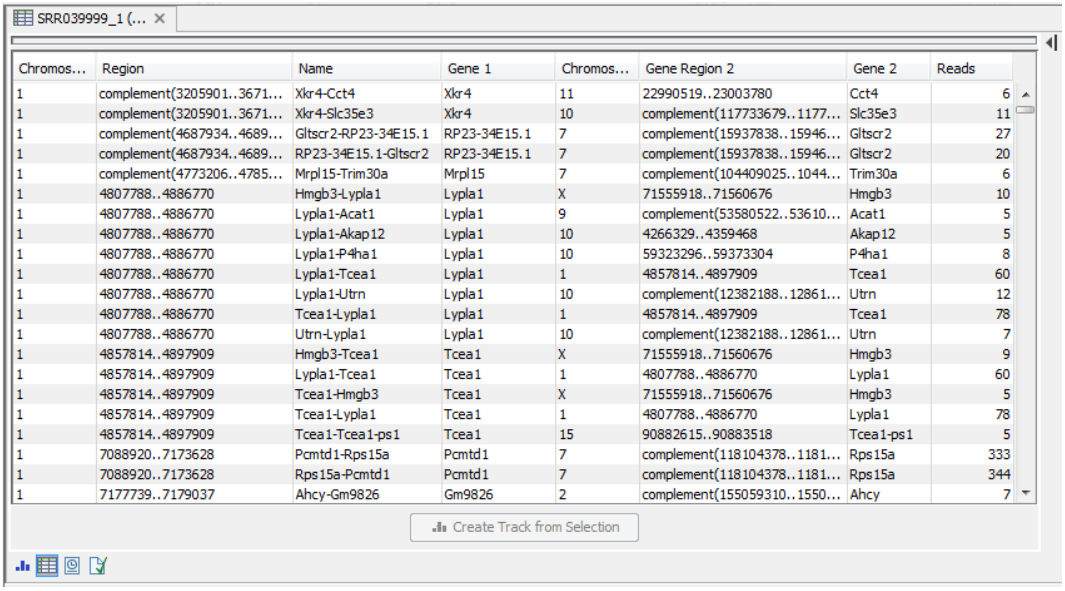


Figure 30.19: *An example of a gene fusion table.*

Each row represents one gene where read pairs suggest it could be fused with another gene. This means that each fusion is represented by two rows.

The **Minimum read count** option in figure 30.9 is used to make sure that only combinations of genes supported by at least this number of read pairs are included. The default value is 5, which means that at least 5 pairs need to connect two genes in order to report it in the result.

The result table shows the following information for each row:

- **Name**. The name of the fusion (the two gene names combined).
- **Information per gene**. Gene name, chromosome and position are included for both genes.
- **Reads**. How many reads that are mapped across the two genes.

Note that the reporting of gene fusions is very simple and should be analyzed in much greater detail before any evidence of gene fusions can be verified. The table should be considered more of a pointer to genes to explore rather than evidence of gene fusions. Please note that you can include the fusion genes track in a track list together with the reads tracks to investigate the mapping patterns in greater detail:

File | **New** | **Track List (** **)**

30.3 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 30.1.

To start the analysis:

Toolbox | **RNA-Seq and Small RNA Analysis (**  **)**| **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. Note that principal components are available when you export the PCA plot to a tabular format (*.tsv, *.csv, *.xls). The export has a row for each sample (dot in the PCA plot), and columns for the coordinates of that point in PC1, PC2, PC3.

30.3.1 Principal component analysis plot (2D)

The default view is a two-dimensional principal component plot as shown in figure 30.20.

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 30.1.

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.

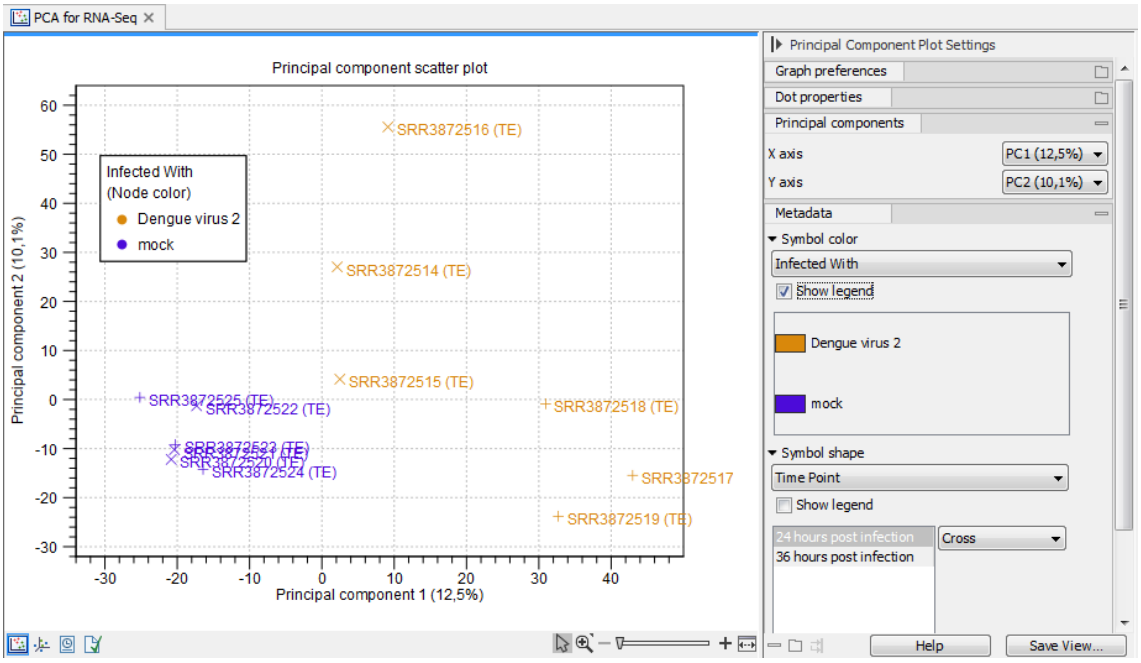


Figure 30.20: *A principal component plot.*

- **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).

30.3.2 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.

Notice that the 3D PCA rendering feature requires a graphics card capable of supporting OpenGL 2.0. Please make sure the latest driver for the graphics card is installed. Indirect rendering (such as x11 forwarding through ssh), remote desktop connection/VNC, and running in virtual machines is not supported.

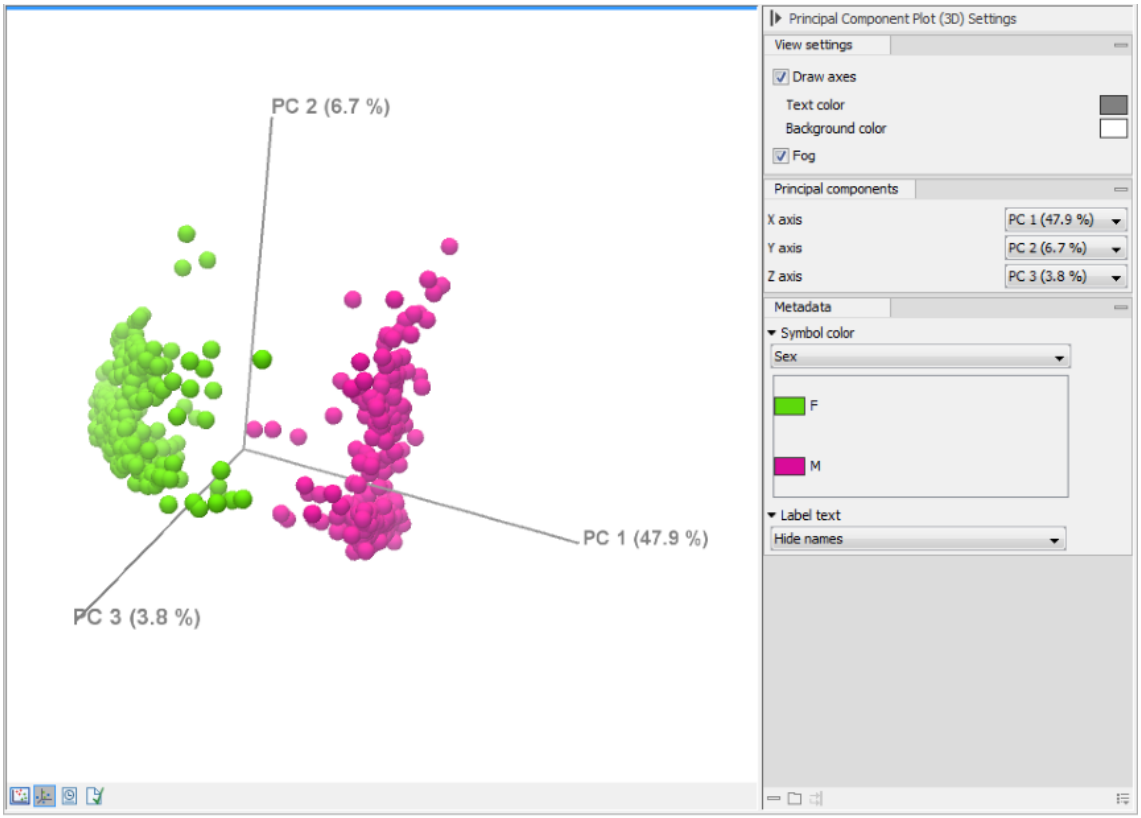


Figure 30.21: *A principal component 3D plot.*

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.

30.4 Differential Expression

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] illustrates 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 DESeq, 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 DEseq, EdgeR, read our benchmark results here: https://digitalinsights.qiagen.com/news/blog/discovery/lasting-expressions/.

30.4.1 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

$$\log y_i = (\text{placebo and Male}) + \text{drugA} + \text{drugB} + \text{Female} + \text{constant}_\text{i} \tag{30.1}$$

where y_i is the expression level for the gene in sample i; the combined term $(\text{placebo and Male})$ 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

$$\log y_i = (\text{placebo and Male}) + \text{constant}_\text{i}.$$

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

$$\log y_i = (\text{placebo and Male}) + \text{drugB} + \text{constant}_\text{i}.$$

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 {3,10,4} 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]. [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.

For dispersion, we used the following strategy:

1. We sort the genes from lowest to highest averageLogCPM (CPM is defined also by the edgeR authors) For each gene, we calculate its loglikelihood at a grid of known dispersions. The known dispersions are $0.2 * 2^i$, where $i = -6, -4.8, -3.6...6$ such that there are 11 values of i in total. You can imagine the results of this as being stored in an array with one column for each gene and one row for each dispersion, with neighboring columns having similar averageLogCPM (because of the sorting in the previous step).

2. We now calculate a weighted loglikelihood for each gene at these same known dispersions. This is the original loglikelihood for that gene at that dispersion plus a "weight factor" multiplied by the average loglikelihood for genes in a window of similar averageLogCPM. 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 / (numberOfSamples - 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 loglikelihoods. Our implementation of the FMM spline is translated from the Fortran code found here `http://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 loglikelihood. This is the dispersion we use for that gene.

Statistical testing

[2] 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 $\frac{1}{n}\sum(x_i - \overline{x})^2$, but this is the *sample* variance and often too low. A commonly used correction for the *population* variance is: $\frac{1}{n-1}\sum(x_i - \overline{x})^2$.

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. 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 in equation 30.1 with those in the reduced GLM $\log y_i = (\mathrm{Male}) + \mathrm{Female} + \mathrm{constant_i}$.

30.4.2 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 30.4.1. 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 the Differential Expression in Two Groups analysis:

> **Toolbox** | **RNA-Seq and Small RNA Analysis** ( )| **Differential Expression in Two Groups** ( )

In the first dialog (figure 30.22), 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.

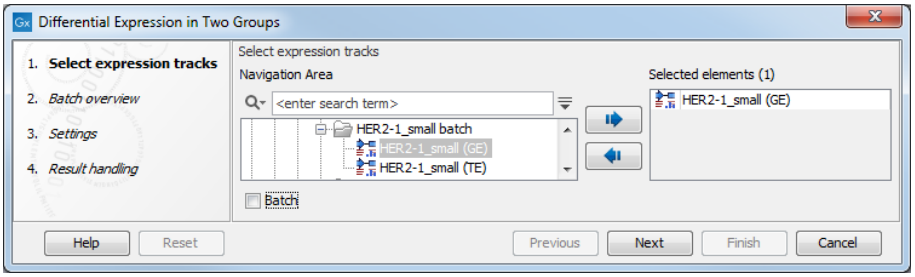


Figure 30.22: *Select expression tracks for analysis.*

In the **Settings** dialog, select a number of control Expression tracks ( ) (GE or TE). A warning message (as seen in figure 30.23) 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.

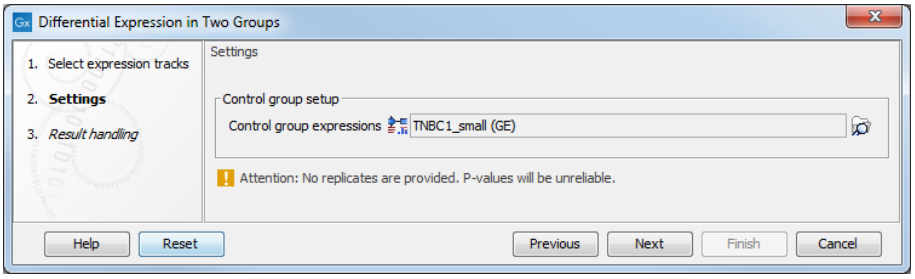


Figure 30.23: *Select enough control expression tracks to ensure that replicates are provided.*

In the final dialog, choose whether or not to filter on average expression prior to FDR correction. 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 30.4.4.

The output of the tool is a comparison table study vs. control that can be visualized as a Statistical comparison track (section 30.4.5) and a Volcano plot (section 30.4.5).

30.4.3 Differential Expression for RNA-Seq

The **Differential Expression for RNA-Seq** tool 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 30.4.1.

To run the Differential Expression for RNA-Seq analysis:

Toolbox | **RNA-Seq and Small RNA Analysis (**)| **Differential Expression for RNA-Seq**
( )

Select a number of Expression tracks ( ) and click **Next** figure 30.24.

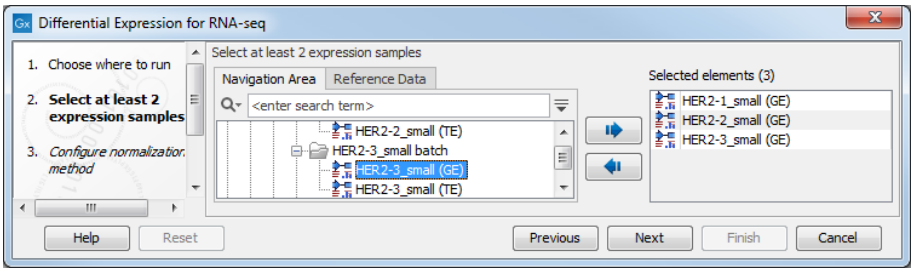


Figure 30.24: *Select a number of Expression tracks.*

For 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" is used. For Eukaryotes the values are "Total Exon Reads", whereas for Prokaryotes the values are "Total Gene Reads".

Note that the order of comparisons can be controlled by changing the order of Expression track inputs.

The normalization options available can be seen in figure 30.25.

First, choose the application that was used to generate the expression tracks selected at the

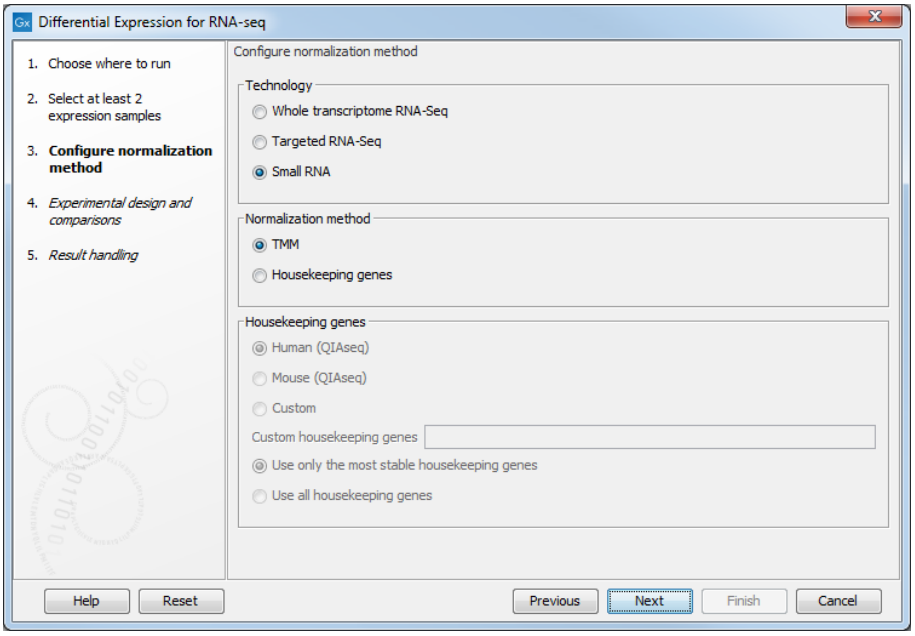


Figure 30.25: *Normalization methods.*

previous step: Whole transcriptome RNA-Seq, Targeted RNA-Seq and 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.

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 for 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** panel (figure 30.26), a Metadata table must be selected that describes the factors and groups for all the samples.

- **Metadata table** The metadata table describing the factors for the selected Expression tracks.

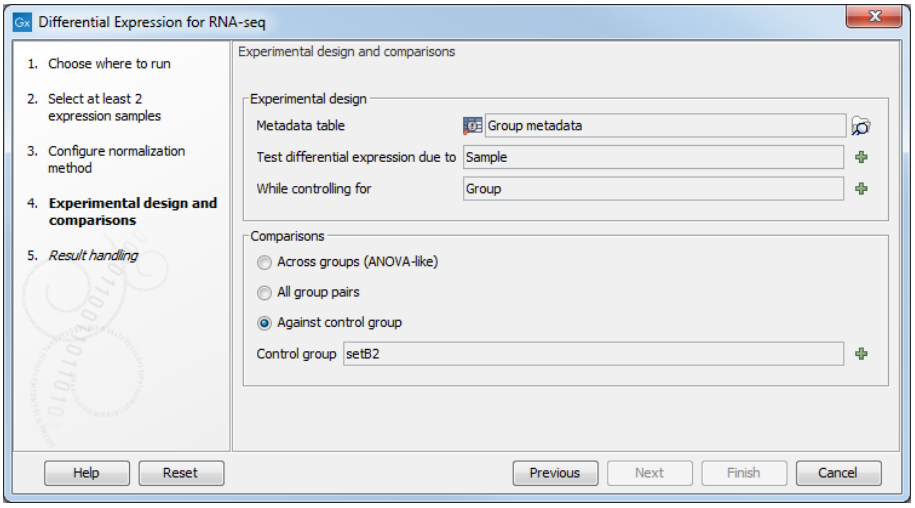


Figure 30.26: *Setting up the experimental design and comparisons.*

- **Test differential expression due to** Specify the one factor differential expression is tested for.
- **While controlling for** Specify confounding factors, i.e., factors that are not of primary interest, but may affect gene expression.

The **Comparisons** panel determines the number and type of statistical comparison tracks output by the tool (see section 30.4.5 for more details).

The Differential Expression for RNA-Seq tool produces different numbers and types of statistical comparison tracks depending on the settings of the **Comparisons** panel. Depending on the choice either a Wald test or a Likelihood Ratio test is 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 RPKM 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 RPKM 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 RPKM values between the two named tissue types.

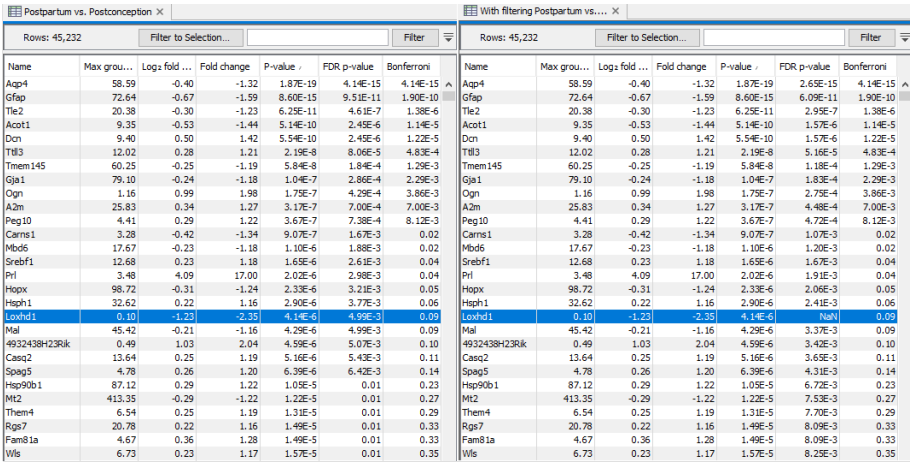
Note: Fold changes are calculated from the GLM, which corrects for differences in library size between the samples and the effects of confounding factors. It is therefore not possible to derive these fold changes from the original counts by simple algebraic calculations.

In the final dialog, choose whether or not to filter on average expression prior to FDR correction. 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 30.4.4.

30.4.4 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 30.27. 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.



Name	Max grou...	Log2 fold ...	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
Ttll3	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
Carns1	3.28	-0.42	-1.34	9.07E-7	1.67E-3	0.02
Mbd6	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
Prl	3.48	4.09	17.00	2.02E-6	2.98E-3	0.04
Hopx	98.72	-0.31	-1.24	2.33E-6	3.21E-3	0.05
Hsph1	32.62	0.22	1.16	2.90E-6	3.77E-3	0.06
Loxhd1	0.10	-1.23	-2.35	4.14E-6	4.99E-3	0.09
Mal	45.42	-0.21	-1.16	4.29E-6	4.99E-3	0.09
4932438H23Rik	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

Name	Max grou...	Log2 fold ...	Fold change	P-value	FDR p-value	Bonferroni
Aqp4	58.59	-0.40	-1.32	1.87E-19	2.65E-15	4.14E-15
Gfap	72.64	-0.67	-1.59	8.60E-15	6.09E-11	1.90E-10
Tle2	20.38	-0.30	-1.23	6.25E-11	2.95E-7	1.38E-6
Acot1	9.35	-0.53	-1.44	5.14E-10	1.57E-6	1.14E-5
Dcn	9.40	0.50	1.42	5.54E-10	1.57E-6	1.22E-5
Ttll3	12.02	0.28	1.21	2.19E-8	5.16E-5	4.83E-4
Tmem145	60.25	-0.25	-1.19	5.84E-8	1.18E-4	1.29E-3
Gja1	79.10	-0.24	-1.18	1.04E-7	1.83E-4	2.29E-3
Ogn	1.16	0.99	1.98	1.75E-7	2.75E-4	3.86E-3
A2m	25.83	0.34	1.27	3.17E-7	4.48E-4	7.00E-3
Peg10	4.41	0.29	1.22	3.67E-7	4.72E-4	8.12E-3
Carns1	3.28	-0.42	-1.34	9.07E-7	1.07E-3	0.02
Mbd6	17.67	-0.23	-1.18	1.10E-6	1.20E-3	0.02
Srebf1	12.68	0.23	1.18	1.65E-6	1.67E-3	0.04
Prl	3.48	4.09	17.00	2.02E-6	1.91E-3	0.04
Hopx	98.72	-0.31	-1.24	2.33E-6	2.06E-3	0.05
Hsph1	32.62	0.22	1.16	2.90E-6	2.41E-3	0.06
Loxhd1	0.10	-1.23	-2.35	4.14E-6	NaN	0.09
Mal	45.42	-0.21	-1.16	4.29E-6	3.37E-3	0.09
4932438H23Rik	0.49	1.03	2.04	4.59E-6	3.42E-3	0.10
Casq2	13.64	0.25	1.19	5.16E-6	3.65E-3	0.11
Spag5	4.78	0.26	1.20	6.39E-6	4.31E-3	0.14
Hsp90b1	87.12	0.29	1.22	1.05E-5	6.72E-3	0.23
Mt2	413.35	-0.29	-1.22	1.22E-5	7.53E-3	0.27
Them4	6.54	0.25	1.19	1.31E-5	7.70E-3	0.29
Rgs7	20.78	0.22	1.16	1.49E-5	8.09E-3	0.33
Fam81a	4.67	0.36	1.28	1.49E-5	8.09E-3	0.33
Wls	6.73	0.23	1.17	1.57E-5	8.25E-3	0.35

Figure 30.27: *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.*

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.
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.

30.4.5 Output of the Differential Expression tools

Statistical comparison tracks

The Differential Expression for RNA-Seq tool will output one or more statistical comparison tracks or tables. The statistical comparison table offers the same functionality than the track, except for the track view.

An example of a statistical comparison track is shown in figure 30.28. Statistical comparison tracks make it possible to show differential expression data alongside other kinds of tracks in a genomic context.

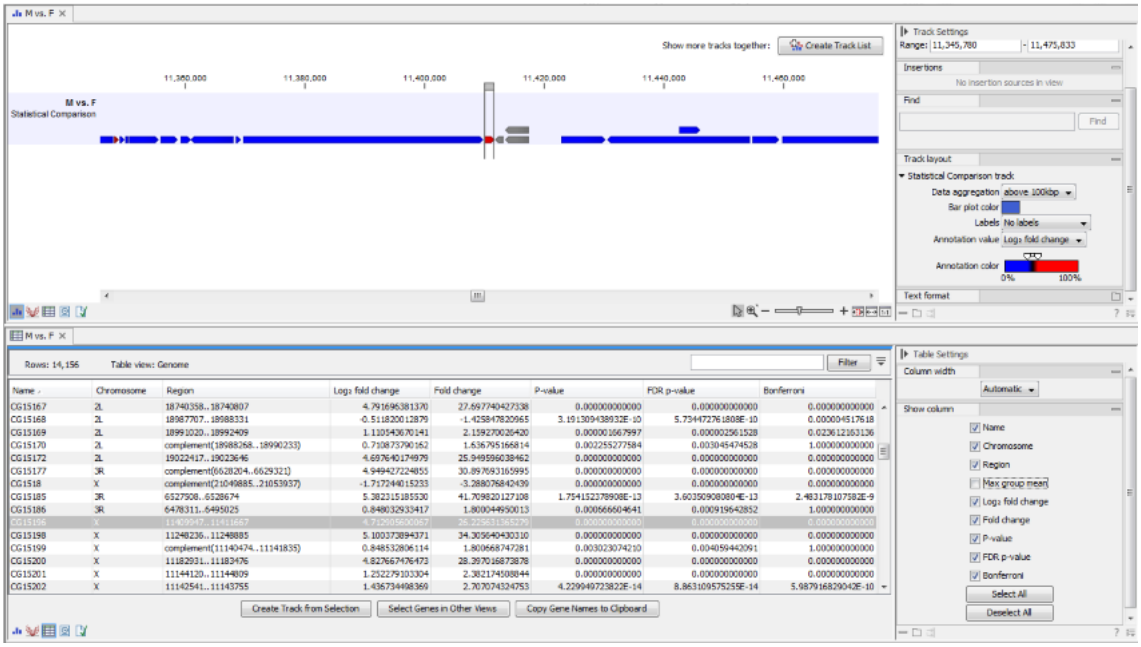


Figure 30.28: *Statistical comparison track view.*

In particular, the Fold Change value will tell you how expression levels in group 2 are relative to that in group 1.

- If expression values in group 2 are twice as large as in group 1, the fold change will be +2.
- If expression values in group 1 are twice as large as in group 2, the fold change will be -2.

Note that it is not possible to derive these fold changes from the CPM values by simple algebraic calculations as the Differential Expression for RNA-Seq tool works by fitting a statistical model (which accounts for differences in sequencing-depth) to raw counts.

The track layout of the statistical comparison track can be customized as follows:

- **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.
- **Bar plot color** Selects the color of aggregated data.
- **Labels** Determines where the gene name should be shown.
- **Annotation value** The value that is graphically shown in detail view:
  - **Max group means** For each group in the statistical comparison, the average RPKM is calculated. This value is the maximum of the average RPKM's.
  - **$\mathbf{Log_2}$ fold change** The logarithmic fold change.
  - **Fold change** The (signed) fold change. Genes/transcripts that are not observed in any sample have undefined fold changes and are reported as NaN (not a number).
  - **P-value** Standard p-value. Genes/transcripts 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.
  - **Bonferroni** The Bonferroni corrected p-value.
- **Annotation color** Determines how the annotation value is mapped onto a color.

The expression track table view has the following options:

- 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.

Volcano plots

Statistical comparisons also offer a volcano plot view.

An example of a volcano plot is shown in figure 30.29.

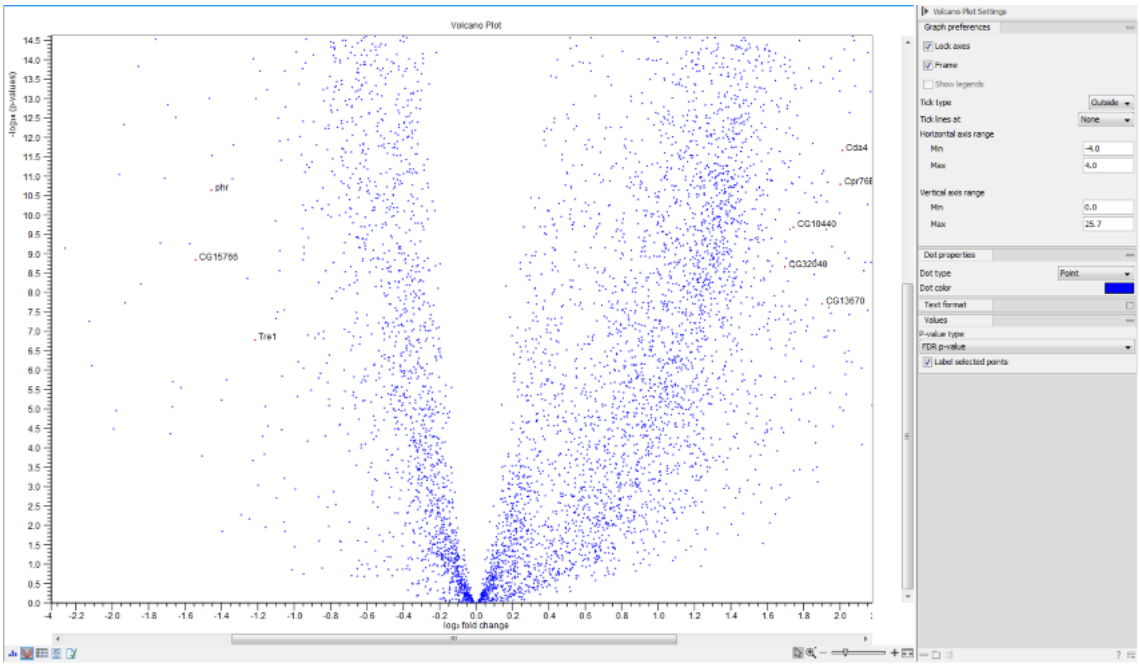


Figure 30.29: *Volcano plot.*

The volcano plot shows the relationship between the p-values of a statistical test and the fold changes among the samples. The $\log_2$ fold changes are plotted on the x-axis, and the $-\log_{10}$ p-values are plotted on the y-axis. Features of interest are typically those in the upper left and right hand corners of the volcano plot, as these have large fold changes (lie far from $x = 0$) and are statistically significant (have large y-values).

Sometimes, the volcano plot will show unexpected pattern looking like "wings", such as the ones highlighted with red arrows in figure 30.30.

These patterns reflect the mathematical relationship between fold change and p-value, which often becomes exposed when there are few replicates and when 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) might be in another "wing".

When working with several samples, it can be useful to make an Expression Browser with all the samples and to open this alongside the Volcano plot. Click a point in the Volcano plot to select it and then right-click to **Select Genes in Other Views**. This will select the appropriate row in the expression browser.

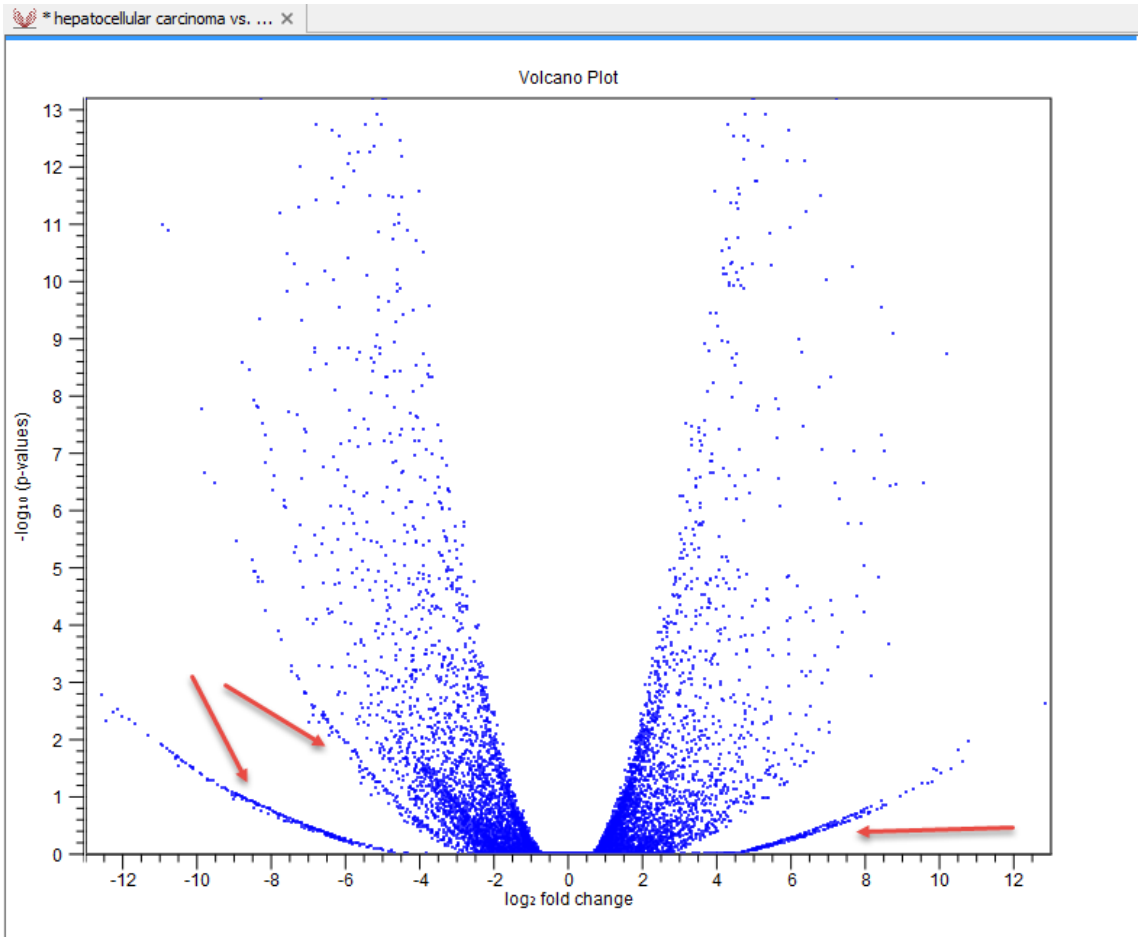


Figure 30.30: *Volcano plot displaying unexpected "wing" patterns.*

Volcano plot side panel It is possible to change the type of p-value from the side panel (see below).

The view settings can be adjusted using the **Side Panel**. 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** and **Text format**, where you can adjust the coloring and appearance of the dots and text.

At the bottom are options for choosing which values to display:

- **P-value type** Selects which type of p-value to use.
- **Label selected points** Chooses whether selected points should be labeled.
- **Lower limit on p-values** Round all p-values smaller than this number to the chosen value (for example, using the default setting, a value of zero will become 1E-16) so even small values can be visualized on a logarithmic scale volcano plot.

Note that if you wish to use the same settings next time you open a volcano plot, you need to save the settings of the **Side Panel**.

30.5 Create Heat Map for RNA-Seq

The **Create Heat Map** tool simultaneously clusters samples and features, showing a two dimensional heat map of expression values. Each column corresponds to one sample, and each row corresponds to a feature (a gene or a transcript). The samples and features are both hierarchically clustered. 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 30.1.

30.5.1 Clustering of features and samples

The hierarchical clustering clusters features by the similarity of their expression profiles over the set of samples. It clusters samples by the similarity of expression patterns over their features.

Each clustering has a tree structure that is generated by

1. Letting each feature or 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 contains all the features or samples).

The tree is drawn so that the distances between clusters are reflected by the lengths of the branches in the tree.

To create a heat map:

> **Toolbox** | **RNA-Seq and Small RNA Analysis** ( )| **Create Heat Map for RNA-Seq** ( )

Select at least two expression tracks ( ) and click **Next**.

This will display the wizard shown in figure 30.31. The hierarchical clustering algorithm requires that you specify a distance measure and a cluster linkage. The distance measure is used to specify how distances between two features or samples should be calculated. The cluster linkage specifies how the distance between two clusters, each consisting of a number of features or samples, should be calculated.

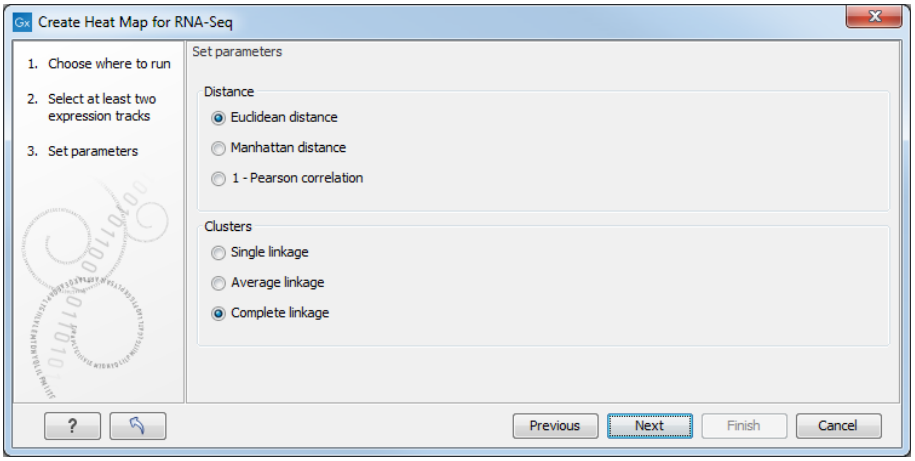


Figure 30.31: *Parameters for Create Heat Map.*

There are three kinds of **Distance measures**:

- **Euclidean distance**. The ordinary distance between two points - the length of the segment connecting them. If $u = (u_1, u_2, \ldots, u_n)$ and $v = (v_1, v_2, \ldots, v_n)$, then the Euclidean distance between u and v is

$$|u - v| = \sqrt{\sum_{i=1}^{n}(u_i - v_i)^2}.$$

- **1 - Pearson correlation**. The Pearson correlation coefficient between two elements $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$ is defined as

$$r = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i - \overline{x}}{s_x}\right) * \left(\frac{y_i - \overline{y}}{s_y}\right)$$

  where $\overline{x}/\overline{y}$ is the average of values in x/y and s_x/s_y is the sample standard deviation of these values. It takes a value $\in [-1, 1]$. Highly correlated elements have a high absolute value of the Pearson correlation, and elements whose values are un-informative about each other have Pearson correlation 0. Using $1 - |Pearsoncorrelation|$ as distance measure means that elements that are highly correlated will have a short distance between them, and elements that have low correlation will be more distant from each other.

- **Manhattan distance**. The Manhattan distance between two points is the distance measured along axes at right angles. If $u = (u_1, u_2, \ldots, u_n)$ and $v = (v_1, v_2, \ldots, v_n)$, then the Manhattan distance between u and v is

$$|u - v| = \sum_{i=1}^{n} |u_i - v_i|.$$

The possible cluster linkages are:

- **Single linkage**. The distance between two clusters is computed as the distance between the two closest elements in the two clusters.
- **Average linkage**. The distance between two clusters is computed as the average distance between objects from the first cluster and objects from the second cluster. The averaging is performed over all pairs (x, y), where x is an object from the first cluster and y is an object from the second cluster.
- **Complete linkage**. The distance between two clusters is computed as the maximal object-to-object distance $d(x_i, y_j)$, where x_i comes from the first cluster, and y_j comes from the second cluster. In other words, the distance between two clusters is computed as the distance between the two farthest objects in the two clusters.

After having selected the distance measure, click **Next** to set up the feature filtering options as shown in figure 30.32.

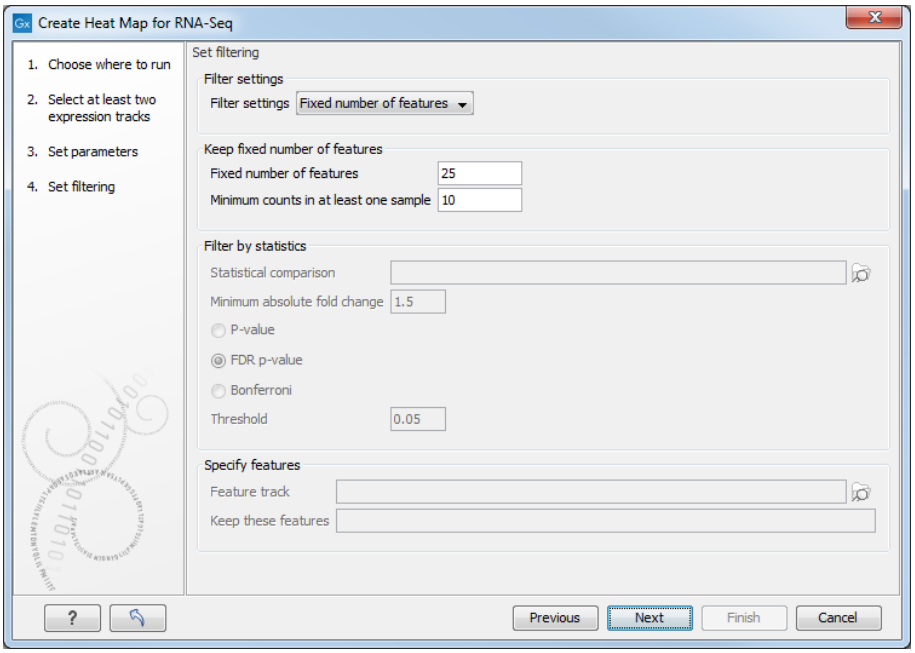


Figure 30.32: *Feature filtering for Create Heat Map.*

Genomes usually contain too many features to allow for a meaningful visualization of all genes or transcripts. Clustering hundreds of thousands of features is also very time consuming. Therefore we recommend reducing the number of features before clustering and visualization.

There are several different **Filter settings** to filter genes or transcripts:

- **No filtering** Keeps all features.
- **Keep fixed number of features**
  - **Fixed number of features** The given number of features with the highest index of dispersion (the ratio of the variance to the mean) are kept. Raw count values (not normalized) are used for calculating the index of dispersion.
  - **Minimum counts in at least one sample** Only features with more than this number of counts in at least one sample will be taken into account. Raw count values (not normalized) are used.
- **Filter by statistics** Keeps features that are differentially expressed according to the specified cut-offs.
  - **Statistical comparison** A single statistical comparison track output by the Differential Expression for RNA-Seq tool.
  - **Minimum absolute fold change** Only features with a higher absolute fold change are kept.
  - **Threshold** Only features with a lower p-value are kept. It is possible to select which type of p-value to use.
- **Specify features** Keeps a set of features, as specified by either a feature track or by plain text.
  - **Feature track** Any genes or transcripts defined in the feature track will be kept.
  - **Keep these features** A plain text list of feature names. Any white-space characters, and ",", and ";" are accepted as separators.

30.5.2 The heat map view

After the tool completes, a heat map like the one shown in (figure 30.33) is produced. In the heat map each row corresponds to a feature and each column to a sample. The 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 expression values used are normalized log CPM values, see section 30.1.

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** group (see figure 30.33).

- **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 normally 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 are 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.

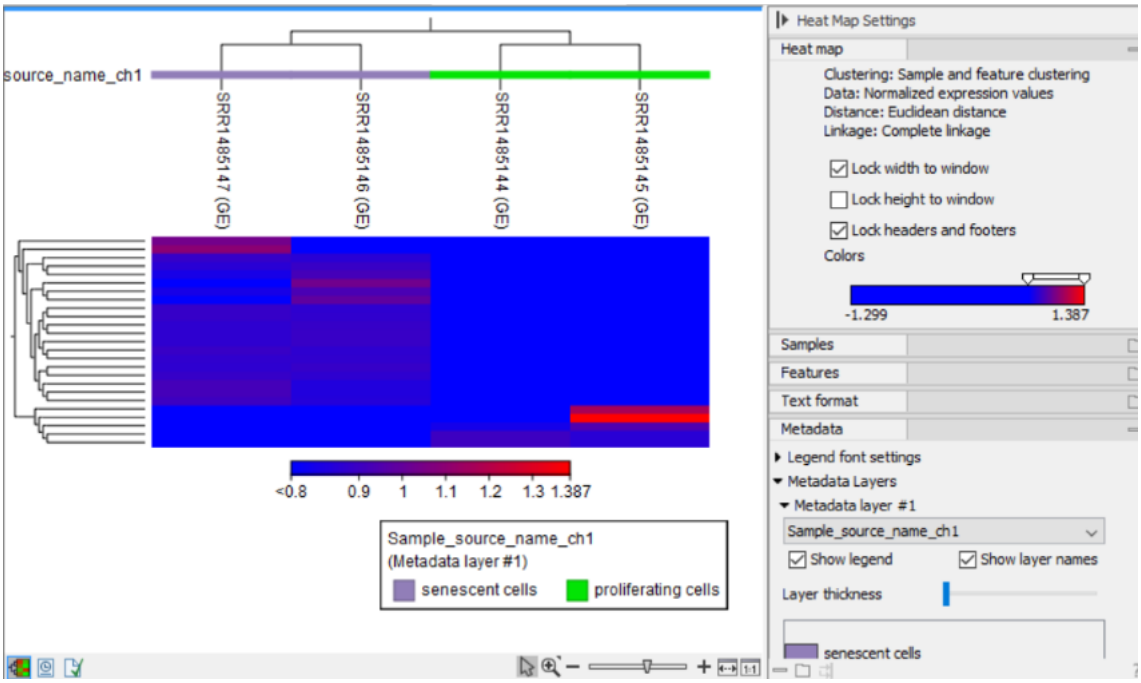


Figure 30.33: *The 2D heat map.*

- **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, color legends, and trees above or below the heat map. The tree options also control the **Tree size**, including the option of showing the full tree, no matter how much space it will use.

The **Metadata** group makes it possible to visualize metadata associated with the Expression tracks:

- **Legend font settings** adjusts the label settings.
- **Metadata layers** Adds a color bar to the hierarchical sample tree, colored according to the value of a chosen metadata table column.

30.6 Create Expression Browser

The **Create Expression Browser** tool makes it possible to inspect gene and transcript expression level counts, annotations and statistics for many samples at the same time.

To run the tool, go to:

Toolbox | **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 30.34).

Click on the **Next** button.

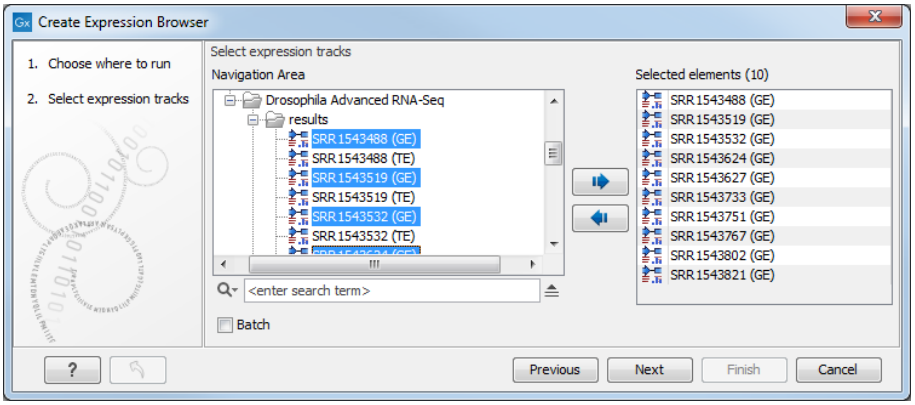


Figure 30.34: *Select expression tracks, either GE or TE.*

In the second wizard dialog, statistical comparisons and an annotation resource can be selected (see figure 30.35). Information from the selected elements is included in the expression browser.

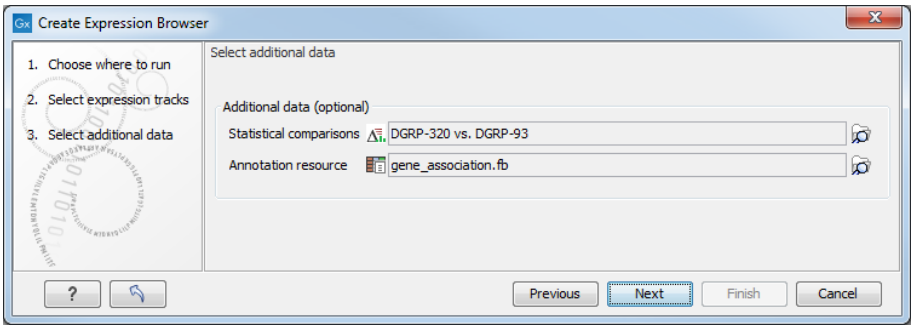


Figure 30.35: *Information from statistical comparisons and an annotation resource can optionally be included in the expression browser being created.*

Statistical comparisons are generated by differential expression tools, described in section 30.4. 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 `http://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 30.1, spelled exactly as shown, 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/`).

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 30.1: Column headers for columns containing identifiers.

- Annotations for human, mouse and rat bundled with Reference Data Sets.

30.6.1 The expression browser

An Expression Browser is shown in figure 30.36.

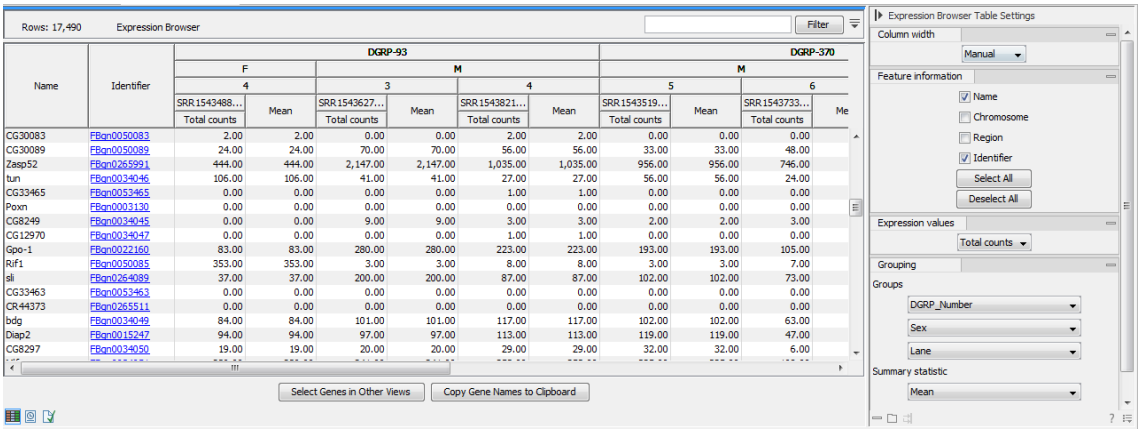


Figure 30.36: *Expression browser table when no statistical comparison or annotations resources were provided.*

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 30.1) 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 30.36.

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 30.36 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 30.37). The columns correspond to the different statistical values generated by the Differential Expression for RNA-Seq tool as detailed in section 30.4.5.

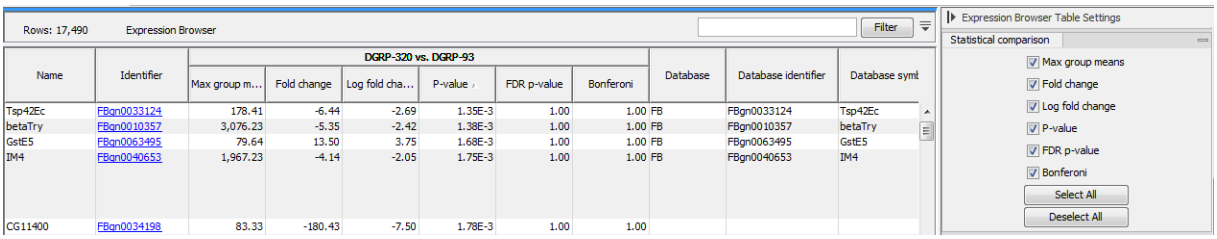


Figure 30.37: *Expression browser table when a statistical comparison is present.*

If an annotation database is provided, extra columns can be displayed in the table using the "Annotation" section of the Settings panel (figure 30.38). 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.

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.

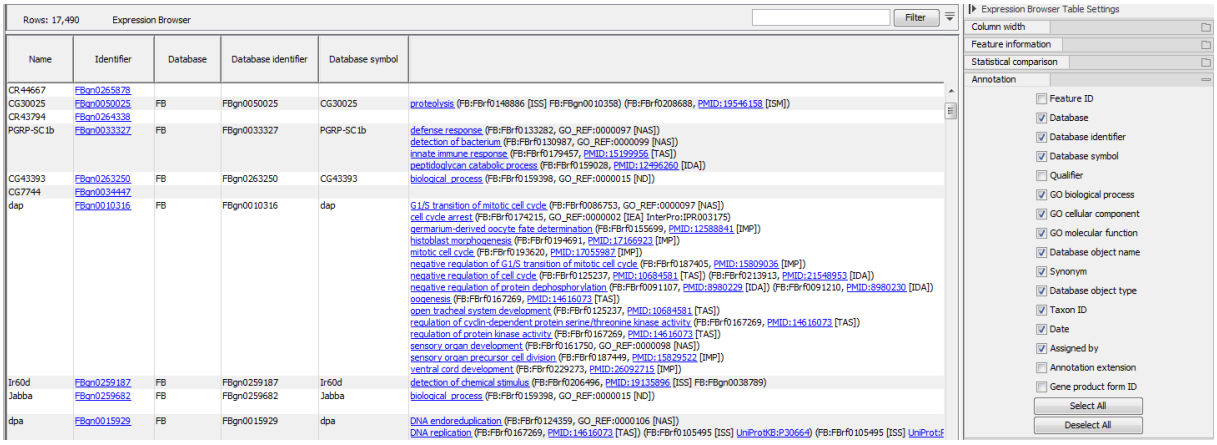


Figure 30.38: *Expression browser table when a GO annotation file is present.*

30.7 Create Venn Diagram for RNA-Seq

The **Create Venn Diagram** tool makes it possible to compare the overlap of differentially expressed genes or transcripts in two or more statistical comparison tracks. The genes considered to be differentially expressed can be controlled by setting appropriate p-value and fold change thresholds.

To create the Venn diagram:

Toolbox | **RNA-Seq and Small RNA Analysis (**  **)**| **Create Venn Diagram** ()

Select a number of statistical comparison tracks () and click **Next** (see figure 30.39).

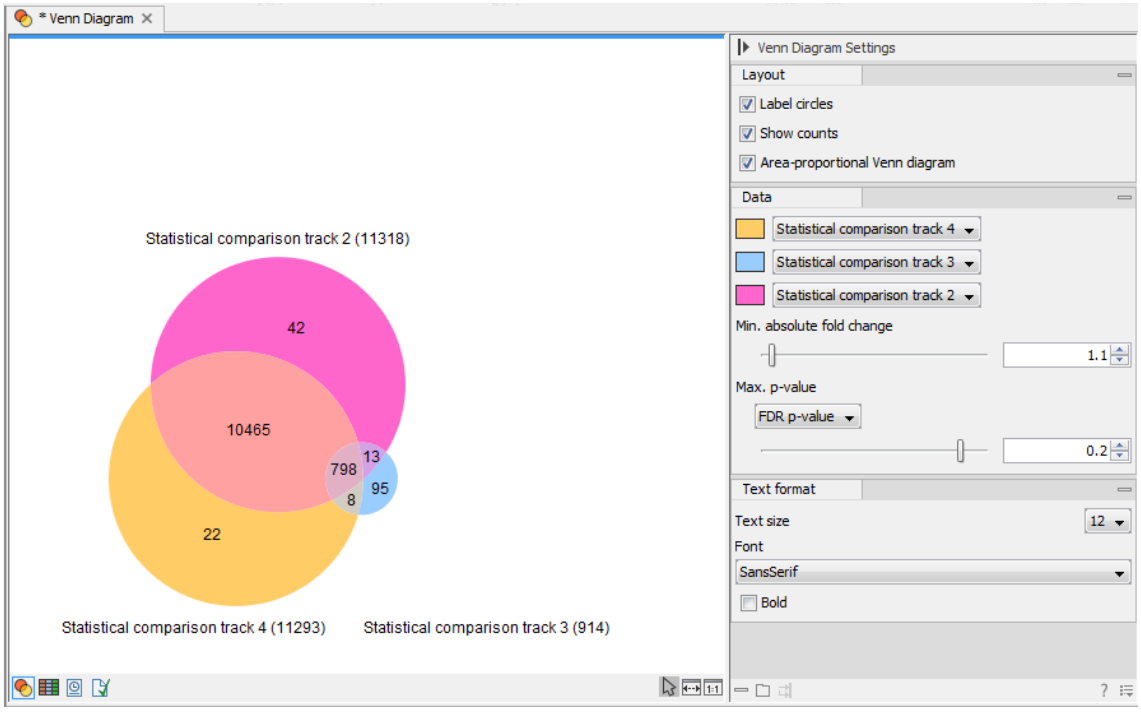


Figure 30.39: *The resulting Venn diagram.*

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 comparison tracks.
- **Show counts** Toggles the display of gene or transcript counts.
- **Area-proportional Venn Diagram** When drawn as a Standard Venn Diagram, circles are drawn with fixed positions and identical size. When drawn in the default Venn Diagram mode, sizes and positions of the circles are adjusted in proportion to the number of overlapping features.

The **Data** side panel group makes it possible to choose the differentially expressed genes or features of interest. The set of statistical comparisons to be compared can be selected using the drop down combo boxes at the top of the group. It is possible to customize the color of a given statistical comparison using the color picker next to the drop down combo box.

- **Minimum absolute fold change** Only genes or transcripts with an absolute fold change higher than the specified threshold are taken into account.
- **Maximum P-value** Only genes or transcripts with a p-value less then 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.

30.7.1 Venn diagram table view

It is possible to inspect the p-values and fold changes for each gene or transcript individually in the Venn diagram table view (see figure 30.40).

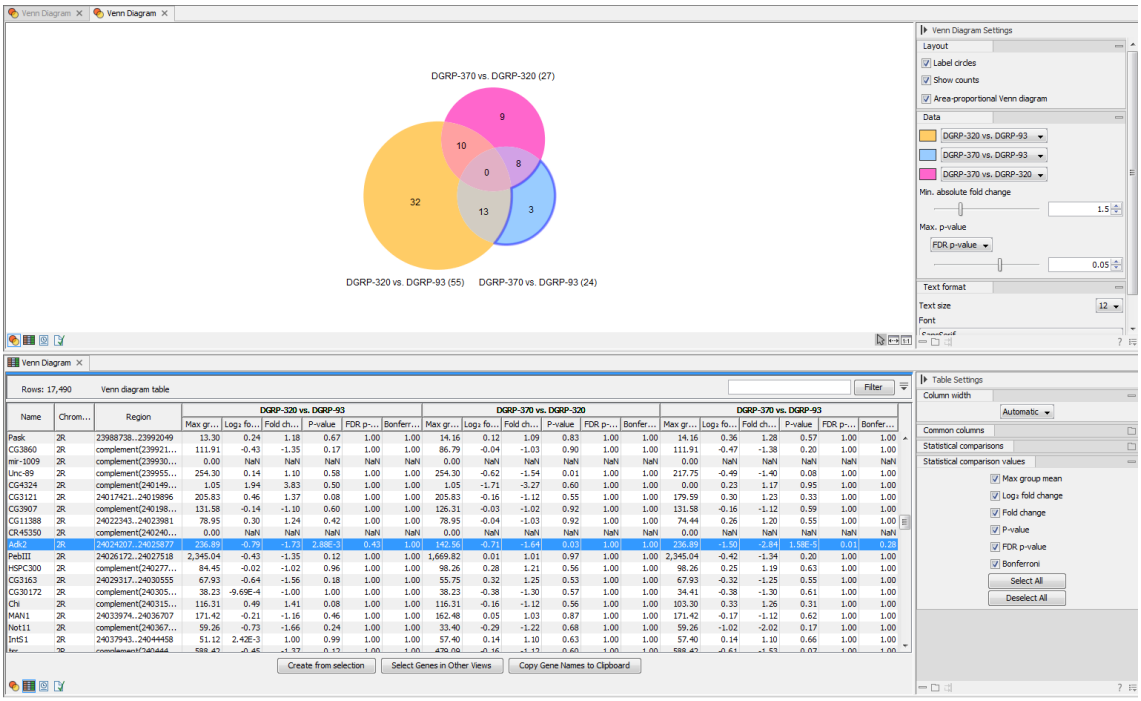


Figure 30.40: *The Venn diagram table view.*

Clicking a circle segment in the Venn Diagram plot will select the genes or transcript 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 list of genes or transcripts using the **Create from selection**.

In the **Side Panel** to the right it is possible to adjust the Table settings. It is possible to adjust the column layout, and select which columns should be included in the table.

30.8 Gene Set Test

The **Gene Set Test** tool tests whether GO terms are over-represented in a set of differentially expressed genes (input as a statistical comparison track) using a hypergeometric test. The tool will require a GO annotation file that must be previously saved in the Navigation Area of the workbench.

GO annotation files are available from several sources (Blast2Go, GO ontology database). Before import, check that the table does have a GO column, and if not, edit the table to change the relevant column header to GO before import in the workbench using the Standard Import function. 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.

RefSeq files are available via the Reference Data Manager, and are saved in the "CLC_Reference" folder in the Navigation Area if you have already downloaded a Reference Data Set.

It is also possible to format a text file of custom annotations into a format the Gene Set Test tool can use (see section K.5 and section K.4).

- The file type should be *.csv.
- The values should be comma-separated and in quotation marks.
- The first column should be "Probe Set ID" or one of the other recognized values mentioned in the manual, and the values in the first column must match the feature names in the data exactly.
- The actual annotations should be found in one of the "Gene Ontology"-type columns: Gene Ontology Biological Process, Gene Ontology Cellular Component, Gene Ontology Molecular Function.
- The separator // is used to separate the name of an annotation from its description, and the separator /// is used to separate different annotations from each other. Each annotation should then look like: "AnnotationA_name // AnnotationA description /// AnnotationB_name // AnnotationB description".

This custom annotation file can be imported using the Standard Import functionality.

To start the tool:

Toolbox | **RNA-Seq and Small RNA Analysis** ( )| **Gene Set Test** ( )

Select a statistical comparison track ( ) and click **Next** (see figure 30.41). To run several statistical comparisons at once, use the batch function.

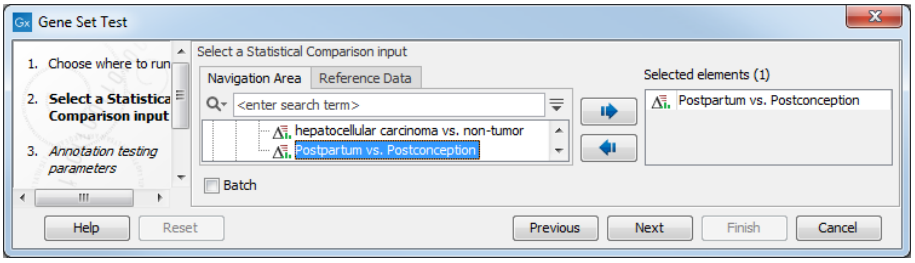


Figure 30.41: *Select one statistical comparison.*

In the "Annotation testing parameters" dialog, you need to specify a GO annotation file and have several annotation testing options(see figure 30.42).

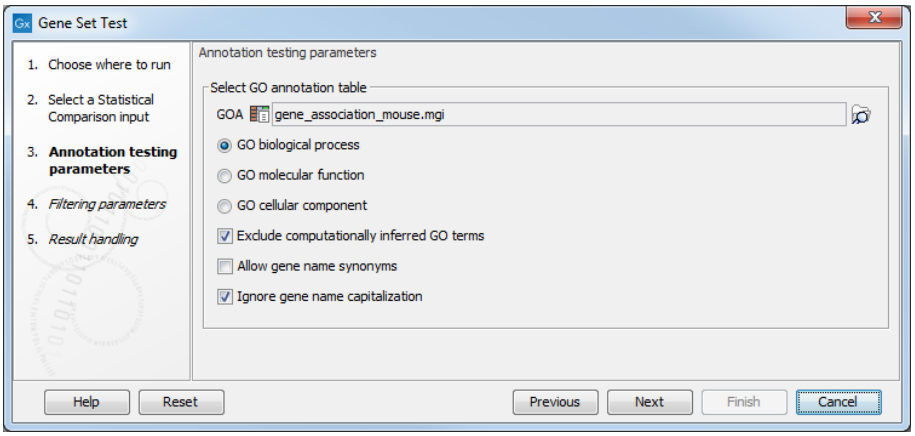


Figure 30.42: *Select annotation tetsing parameters.*

- **GOA**: Specify a GO annotation file (such as described in the introduction of this section) using the Browse button to the right of the field.
- **GO biological process** Tests for enriched GO biological processes, i.e., a series of events or molecular functions such as "lipid storage" or "chemical homeostasis".
- **GO molecular function** Tests for enriched GO molecular functions. The GO molecular functions of a gene can be such as "retinoic acid receptor activity" or "transcription regulator activity".
- **GO cellular component** Tests for enriched GO cellular component. A GO cellular component ontology describes locations, such as "nuclear inner membrane" or "ubiquitine ligase complex".
- **Exclude computationally inferred GO terms** excludes uncurated GO terms with evidence code IEA, i.e., terms that were automatically curated but not reviewed by a curator.
- **Allow gene name synonyms** allows matching of the gene name with database identifiers and synonyms.
- **Ignore gene name capitalization** ignores capitalization in feature names: a gene called "Dat" in the statistical comparison track will be matched with the one called "dat" in the annotation file when this option is checked. If "Dat" and "dat" are believe to be different genes, the option should be unchecked.

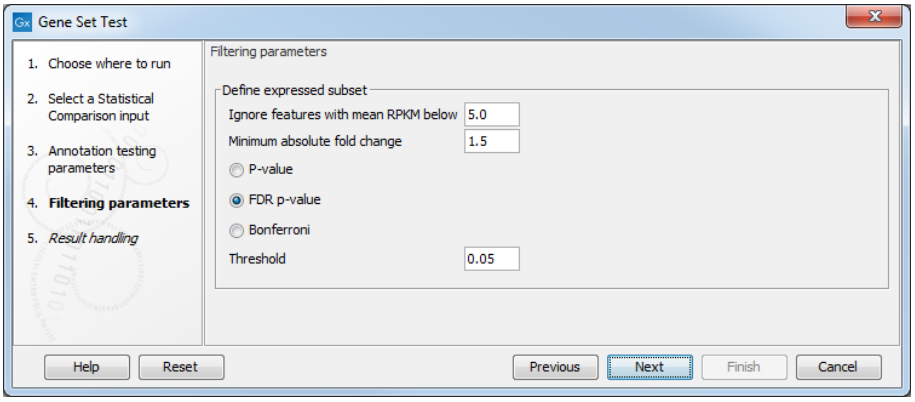


Figure 30.43: *Specify filtering parameters.*

Click Next to access the "Filtering parameters" dialog (see figure 30.43).

Instead of annotating all genes present in the statistical comparison track, it is possible to focus on the subset of genes that are differentially expressed. The filtering parameters allow you to define this subset:

- **Ignore features with mean RPKM below**. Only features where the highest group mean RPKM exceeds this limit will be included in the analysis.
- **Minimum absolute fold change**. Define the minimum absolute fold change value for a feature, and specify whether this fold change should calculated as **p-value**, **FDR p-value** or **Bonferroni** (for a detailed definition of these, see section 31.5.4).
- **Threshold**. Maximum p-value for a feature.

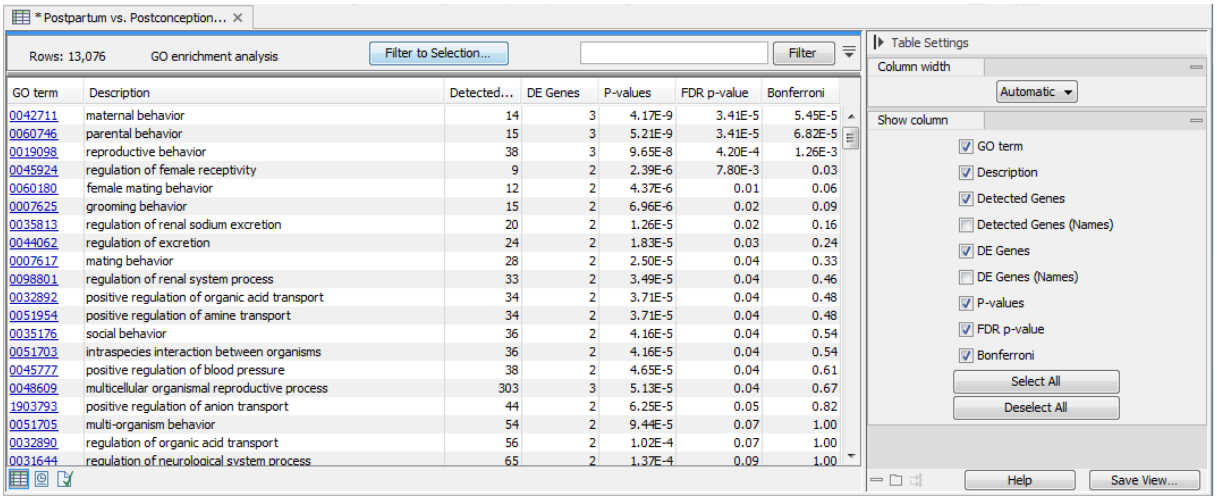
Click **Finish** to open or save the file in a specified location of the Navigation Area.

During analysis, a black banner in the left hand side of the workbench warns if duplicate features were found while processing the file. If you get this warning message, consider unchecking the "Ignore gene name capitalization" option.

The output is a table called "GO enrichment analysis" (see figure 30.44). The table is sorted in order of ascending p-values but it can easily be sorted differently, as well as filtered to highlight only the GO terms that are over-represented.

The table lists for each GO term the number and names of **Detected Genes**, i.e., the total number of genes in the annotation for a given GO term which is being considered for the analysis, and of DE (Differentially Expressed) Genes. Genes that are not detected (i.e., genes that have Max group mean = 0, meaning they are not expressed in any sample) are not included in the analysis. By excluding undetected genes, we make the background of the test specific to the experiment (for example, if someone is comparing liver cells under two conditions, then the most appropriate background is the set of genes expressed in the liver).

The table also provides FDR and Bonferroni-corrected p-values. When testing for the significance of a particular GO term, we take into account that GO has a hierarchical structure. For example, when testing for the term "GO:0006259 DNA metabolic process", we include all genes that are annotated with more specific GO terms that are types of DNA metabolic process such as "GO:0016444 somatic cell DNA recombination". Also note that the p-values provided in the table



GO term	Description	Detected...	DE Genes	P-values	FDR p-value	Bonferroni
0042711	maternal behavior	14	3	4.17E-9	3.41E-5	5.45E-5
0060746	parental behavior	15	3	5.21E-9	3.41E-5	6.82E-5
0019098	reproductive behavior	38	3	9.65E-8	4.20E-4	1.26E-3
0045924	regulation of female receptivity	9	2	2.39E-6	7.80E-3	0.03
0060180	female mating behavior	12	2	4.37E-6	0.01	0.06
0007625	grooming behavior	15	2	6.96E-6	0.02	0.09
0035813	regulation of renal sodium excretion	20	2	1.26E-5	0.02	0.16
0044062	regulation of excretion	24	2	1.83E-5	0.03	0.24
0007617	mating behavior	28	2	2.50E-5	0.04	0.33
0098801	regulation of renal system process	33	2	3.49E-5	0.04	0.46
0032892	positive regulation of organic acid transport	34	2	3.71E-5	0.04	0.48
0051954	positive regulation of amine transport	34	2	3.71E-5	0.04	0.48
0035176	social behavior	36	2	4.16E-5	0.04	0.54
0051703	intraspecies interaction between organisms	36	2	4.16E-5	0.04	0.54
0045777	positive regulation of blood pressure	38	2	4.65E-5	0.04	0.61
0048609	multicellular organismal reproductive process	303	3	5.13E-5	0.04	0.67
1903793	positive regulation of anion transport	44	2	6.25E-5	0.05	0.82
0051705	multi-organism behavior	54	2	9.44E-5	0.07	1.00
0032890	regulation of organic acid transport	56	2	1.02E-4	0.07	1.00
0031644	regulation of neurological system process	65	2	1.37E-4	0.09	1.00

Figure 30.44: *The GO enrichment analysis table generated by the Gene Set Test tool.*

are meant as a guide, as GO annotations are not strictly independent of each other (for example, "reproduction" is a broad category that encompass a nested set of terms from other categories such as "pheromone biosynthetic process").

30.8.1 Tool output and GAF file comparison

It can happen that you will find some discrepancy in the number of genes in your Gene Set test results and the original GAF file. This is also the case for results of the Hypergeometric Tests on Annotations, and Gene Set Enrichment Analysis (GSEA) tools.

In some cases, **the result contains more genes than expected for a GO term**. When testing for the significance of a particular GO term, we take into account that GO has a hierarchical structure. For example, when testing for the term "GO:0006259 DNA metabolic process", we include all genes that are annotated with more specific GO terms that are types of DNA metabolic process. As can be seen on figure 30.45, these include genes that are annotated with the more specific term "GO:0033151 V(D)J recombination". This is because "GO:0033151 V(D)J recombination" is a subtype of "GO:0002562 somatic diversification of immune receptors via germline recombination within a single locus", which in turn is a subtype of "GO:0016444 somatic diversification of immune receptors", which is a subtype of "GO:0006310 DNA recombination", which is a subtype of the original search term "GO:0006259 DNA metabolic process". Websites like geneontology.org ([Ashburner et al., 2000] and [The Gene Ontology Consortium, 2019]) provide an overview of the hierarchical structure of GO annotations.

In other cases, **some annotations in the GAF file are missing from the Gene Set Test result**. If the option "Exclude computationally inferred GO terms" is selected, then annotations in the GAF file that are computationally inferred (their description includes the `[IEA]` tag as in figure 30.46) will be excluded from the result. Thus, if the GAF file shows that almost all annotations are computationally inferred, we recommend the tool be run without "Exclude computationally inferred GO terms".



Figure 30.45: *Hierarchical structure of GO annotations from geneontology.org.*

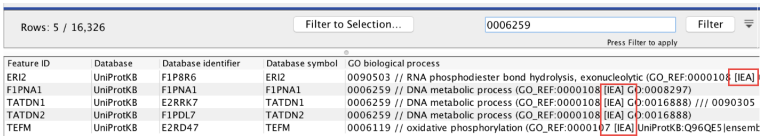


Figure 30.46: *The [IEA] tag describes annotations that are computationally inferred.*

30.9 miRNA analysis

30.9.1 Quantify miRNA

The Quantify miRNA tool counts and annotates miRNAs using miRBase and performs expression analysis of the results. The tool will output two expression tables. The "Grouped on mature" table has a row for each mature miRNA. The same mature miRNA may be produced from different precursor miRNAs. The "Grouped on seed" table has a row for each seed sequence. The same seed sequence may be found in different mature miRNAs.

The tool will take:

- Trimmed reads (both UMI or normal)
- a miRBase database. Note: we do not support custom databases that include isoforms of small RNA, such as isopiRNA databases.

- and optionally Spike-ins: A list of sequences that have been spiked-in. Mapping against this set of sequences will be preformed 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:

Toolbox | **RNA-Seq and Small RNA Analysis** ( )| **miRNA Analysis** ( ) | **Quantify miRNA** ( )

First select the trimmed reads as in figure 30.47.

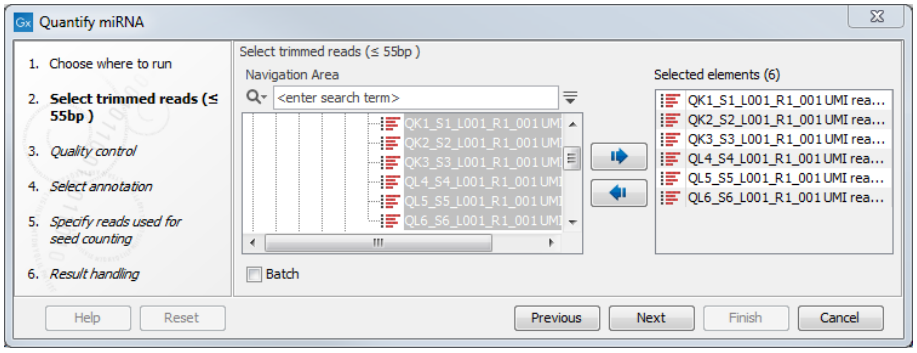


Figure 30.47: *Select the reads.*

If the sequencing was performed using **Spike-ins** controls, the option "Enable spike-ins" can be enabled in the Quality control dialog (figure 30.48), 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.

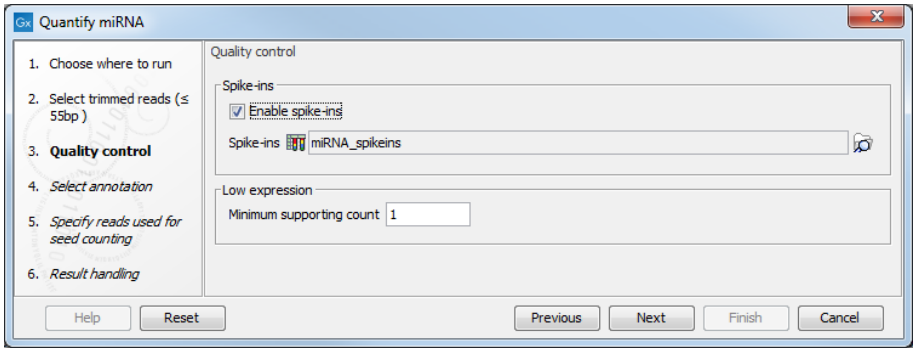


Figure 30.48: *Specifying spike-ins is optional, and you can change the threshold under which a small RNA will be considered expressed.*

In the annotation dialog, several configurations are available.

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 30.49).

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 I.1.7.

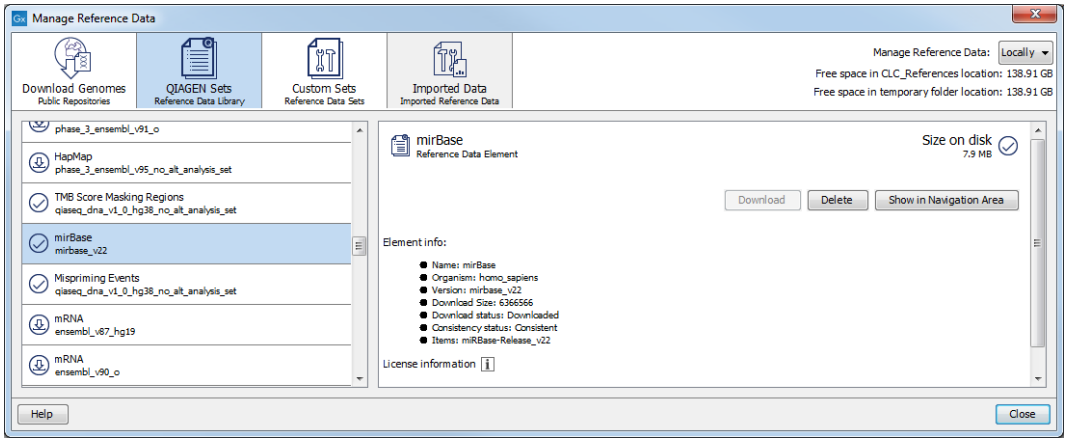


Figure 30.49: *Download the latest miRBase database in the Workbench.*

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 in the priority list (figure 30.50). 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.

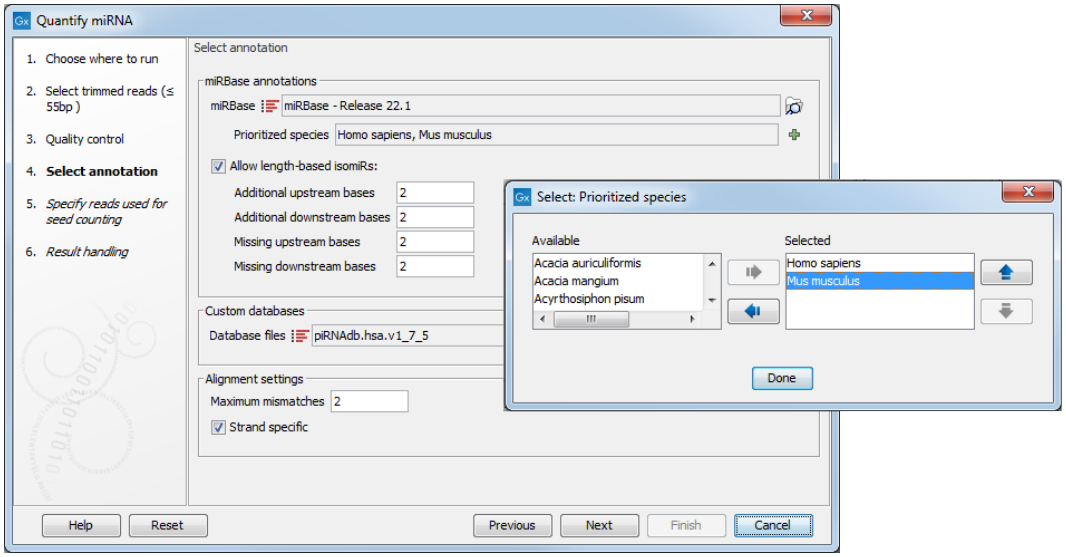


Figure 30.50: *Specify and prioritize species to use for annotation, and how stringent the annotation should be.*

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 analyses.

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. The option

"Strand specific" is checked by default, which means that only the plus strand of the reference will be searched.

In the next dialog (figure 30.51), 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.

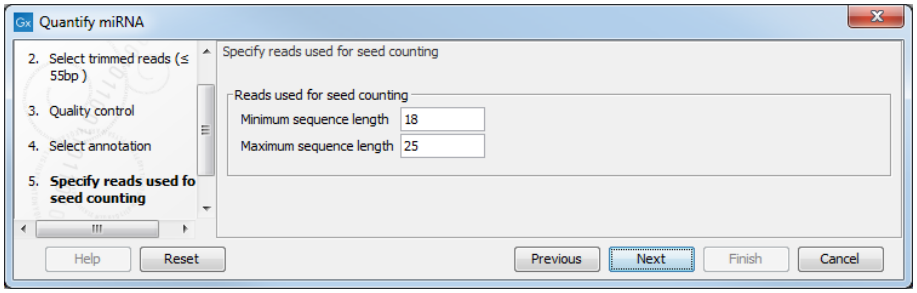


Figure 30.51: *This dialog defines the length of the reads that will be merged according to their seed.*

The tool will output expression tables: One is called "grouped on mature", with the miR of the mature product used as key. The other is "grouped on seed" and the sequence is used as key. In addition, and depending on the options selected in the last dialog, the tool can output a report, a sequence list of reads that could not be mapped to any references and an expression table "grouped on custom databases" (if any was provided) that can be used for subsequent expression analysis tools such as Differential Expression, PCA for RNA-Seq and Create Heat-Map for RNA-Seq. These outputs are detailed in the next section.

30.9.2 Quantify miRNA outputs

Grouped on seed ()

In this expression table, there is a row for each seed sequence (figure 30.52).

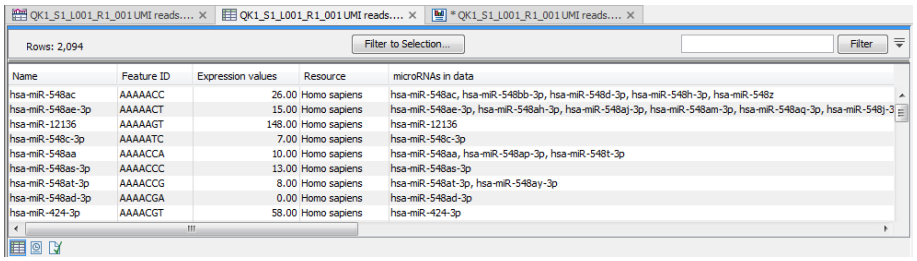


Figure 30.52: *Expression table grouped on seed.*

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 30.53). Double click on a row to open a unique reads alignment (seen at the bottom of figure 30.53). 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 30.9.3.

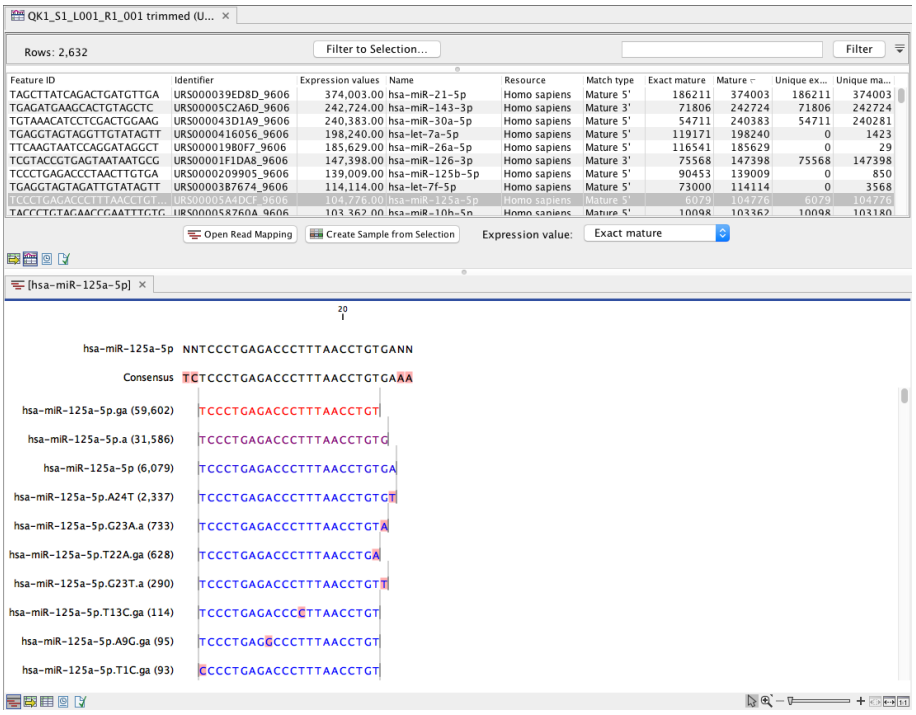


Figure 30.53: *Expression table grouped on mature, with a view of a unique reads alignment.*

This table contains the following information:

- **Feature ID** Sequence of the mature miRNA
- **Identifier** The RNAcentral Accession of the mature miRNA
- **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 30.54). Double click on a row to open a unique reads alignment (seen at the bottom of figure 30.54). 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 30.9.3.

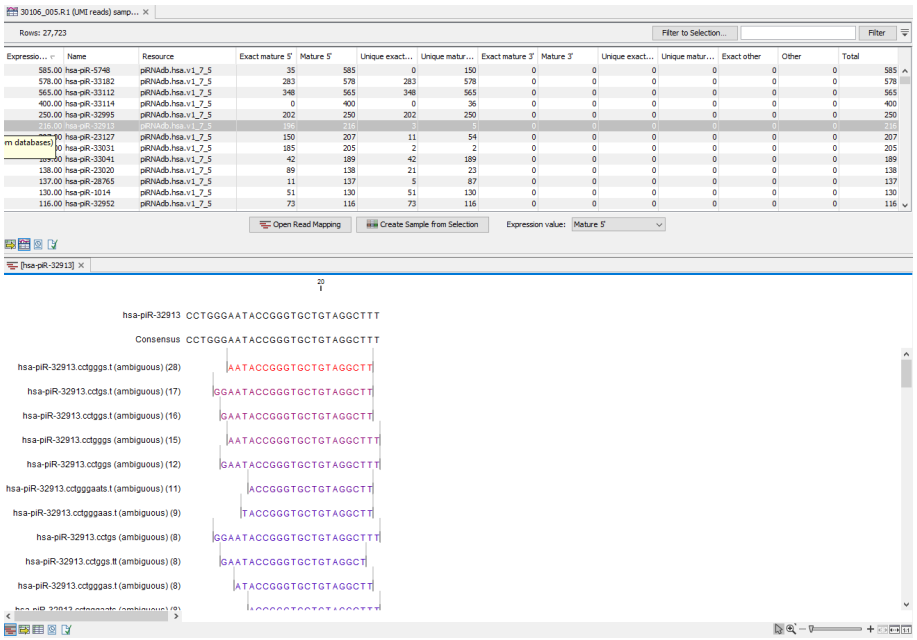


Figure 30.54: *Expression table grouped on custom database, with a view of a unique reads alignment.*

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**

Reports and discarded reads

The workflow also outputs reports:

- A trimming report, only relevant when working with Ion Torrent reads
- A UMI Report that indicates how many reads were ignored and the reason why they were not included in a UMI read. Discarded reads are also output for review.
- A Quantification 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 reports issued for one sample using the Create Combined miRNA Report tool.

30.9.3 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: http://github.com/miRTop/incubator/blob/master/isomirs/isomir_naming.md

Deletions are in lowercase and there is a suffix s for 5' deletions (figure 30.55):

```
ACGT (ref)
 CGT .as
ACG  .t
  GT .acs
```

Figure 30.55: *Naming of deletions.*

Insertions are in uppercase and there is a suffix s for 5' insertions (figure 30.56):

```
 ACGT  (ref)
 ACGTT .T
 ACGTG .G
AACGT  .As
 ACGTTT .TT
 ACGTGT .GT
```

Figure 30.56: *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 30.57):

```
 ACGT (ref)
 ATGT .C2T
 ATAT .C2T.G3A
AATGT .As.C2T (and not C3T)
  TGT .as.C2T (and not C1T)
```

Figure 30.57: *Naming of mutations.*

Deletions followed by insertions will be annotated as shown in figure 30.58:

```
ACGT (ref)
TCGT .A1T (and not .as.Tes)
ACGA .T4A (and not .t.Ae)
```

Figure 30.58: *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.

30.9.4 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:

Toolbox | **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 30.59), 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`

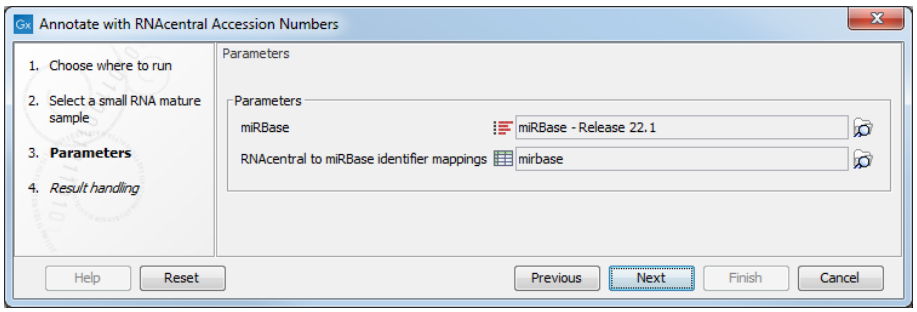


Figure 30.59: *Specify the miRBase database and RNACentral table.*

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 it can be passed on to the gene set test against gene ontology.

30.9.5 Create Combined miRNA Report

Create Combined miRNA Report combines all reports created from the QIAseq miRNA Quantification workflow across multiple samples into a single report in order to get 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.

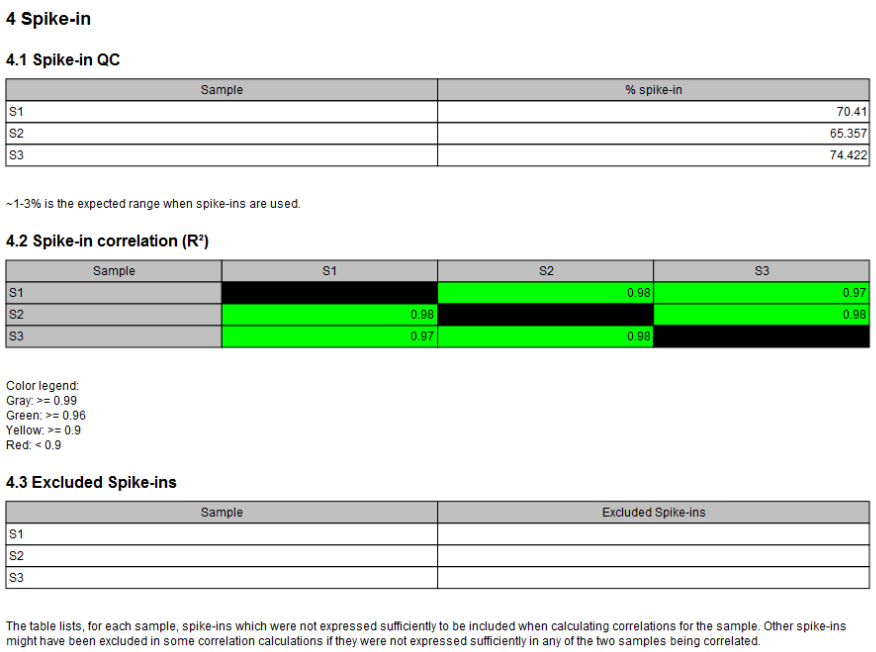
To start the tool, go to:

Toolbox | **RNA-Seq and Small RNA Analysis (**)| **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 30.60). 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 30.61), and another table for the Top non-miRNA sequences when one was provided.



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 (R^2)

Sample	S1	S2	S3
S1		0.98	0.97
S2	0.98		0.98
S3	0.97	0.98	

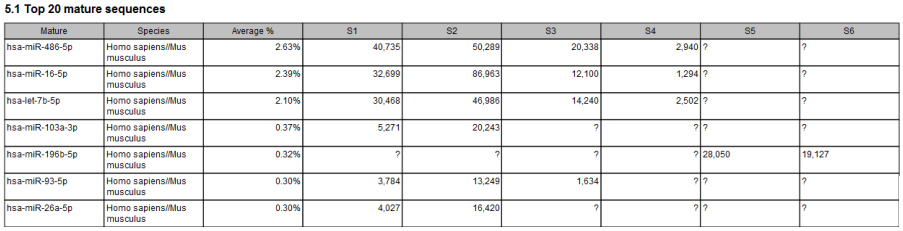
Color legend:
Gray: >= 0.99
Green: >= 0.96
Yellow: >= 0.9
Red: < 0.9

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 30.60: *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	86,963	12,100	1,294	?	?
hsa-let-7b-5p	Homo sapiens//Mus musculus	2.10%	30,468	46,986	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 30.61: *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.*

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 30.62).

30.9.6 Extract IsomiR Counts

The **Extract IsomiR Counts** takes as input "grouped on mature" expression table, output by the **Quantify miRNA** tool. It extracts IsomiR composition and count information from each underlying miRNA alignment and produces a table 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.

To run **Extract IsomiR Counts**, go to:

Toolbox | **RNA-Seq and Small RNA Analysis** ( )| **miRNA Analysis** ( ) | **Extract IsomiR Counts** ( )

Select a "grouped on mature" expression table as input. Information from all miRNAs in this table will be extracted.

Extract IsomiR Counts output

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	?	?
TTCGAAT	0.01%	86	137	79	23	?	?
TTCGATT	0.01%	76	110	71	25	?	?
TCGAATC	0.01%	76	106	84	20	?	?
TCGATTC	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
TTCGAGT	0.00%	45	72	36	?	?	?
GGTGTAG	0.00%	?	?	38	13	?	?
AAAGCTA	0.00%	?	?	?	17	?	?
TAATAGG	0.00%	?	?	?	16	?	?
GCTAAAC	0.00%	?	?	?	?	173	166
TTGACCT	0.00%	?	?	?	15	?	?
GTTAAGT	0.00%	?	?	?	15	?	?
GTTCGAT	0.00%	38	?	41	?	?	?

Figure 30.62: *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.*

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 30.9.3 for further details.

Chapter 31

Microarray analysis

Contents

31.1 Experimental design . . . 867
31.1.1 Setting up an experiment . . . 867
31.1.2 Organization of the experiment table . . . 869
31.1.3 Adding annotations to an experiment . . . 874
31.1.4 Scatter plot view of an experiment . . . 874
31.1.5 Cross-view selections . . . 876
31.2 Transformation and normalization . . . 876
31.2.1 Selecting transformed and normalized values for analysis . . . 877
31.2.2 Transformation . . . 877
31.2.3 Normalization . . . 878
31.3 Quality control . . . 880
31.3.1 Create Box Plot . . . 880
31.3.2 Hierarchical Clustering of Samples . . . 883
31.3.3 Principal Component Analysis . . . 886
31.4 Feature clustering . . . 889
31.4.1 Hierarchical clustering of features . . . 889
31.4.2 K-means/medoids clustering . . . 892
31.5 Statistical analysis - identifying differential expression . . . 895
31.5.1 Empirical analysis of DGE . . . 896
31.5.2 Tests on proportions . . . 898
31.5.3 Gaussian-based tests . . . 899
31.5.4 Corrected p-values . . . 900
31.5.5 Volcano plots - inspecting the result of the statistical analysis . . . 901
31.6 Annotation tests . . . 903
31.6.1 Hypergeometric Tests on Annotations . . . 903
31.6.2 Gene Set Enrichment Analysis . . . 905
31.7 General plots . . . 907
31.7.1 Histogram . . . 907
31.7.2 MA plot . . . 909
31.7.3 Scatter plot . . . 911

This section focuses on analysing expression data from sources such as microarrays using tools found under the **Microarray Analysis ( )** folder of the Toolbox. 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 Toolbox and are described in section 30 and section 30.9.

31.1 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.

31.1.1 Setting up an 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..

To set up an experiment:

Toolbox | **Microarray Analysis ( )**| **Set Up Experiment ( )**

Select the samples that you wish to use by double-clicking or selecting and pressing the **Add** ( ) button (see figure 31.1).

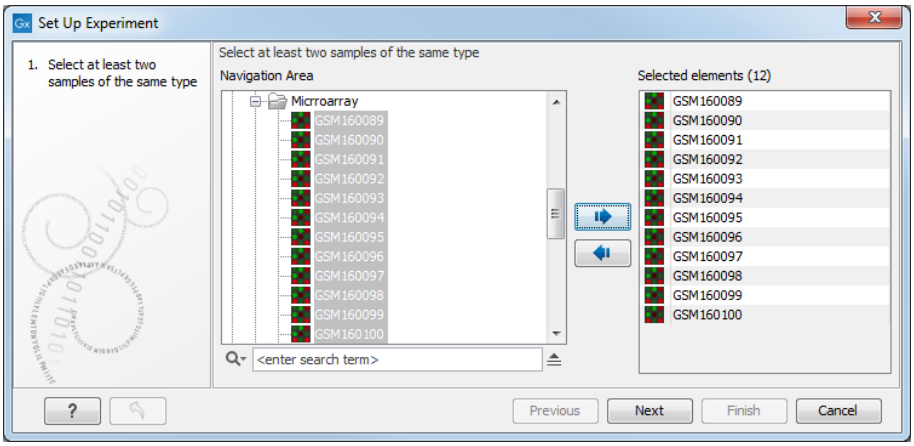


Figure 31.1: *Select the samples to use for setting up the experiment.*

Clicking **Next** shows the dialog in figure 31.2.

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 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.

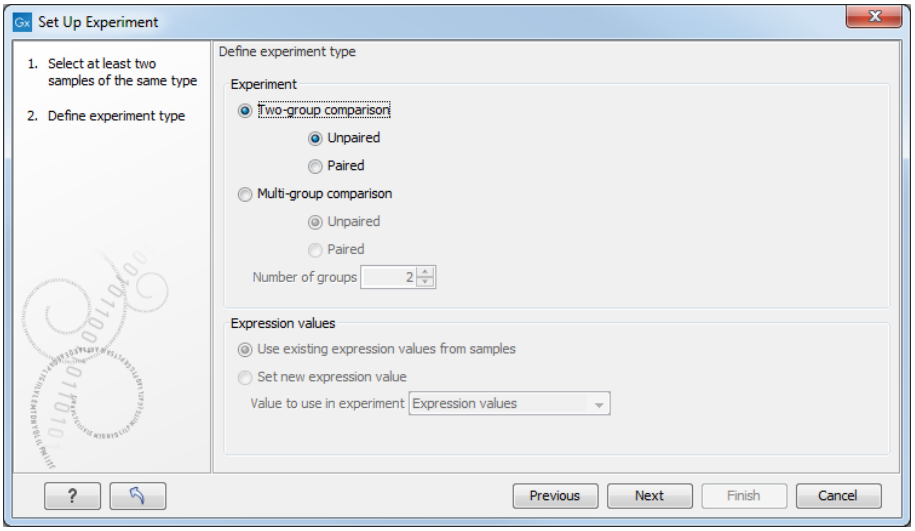


Figure 31.2: *Defining the number of groups and expression value type.*

- **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]: $RPKM = \frac{total\ exon\ reads}{mapped\ reads(millions) \times exon\ length\ (KB)}$. 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 30.2.4.
  - **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 31.3.

Depending on the number of groups selected in figure 31.2, 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** ( ) button. If you need more groups, simply click **Add New Group**.

Click **Next** when you have named the groups, and you will see figure 31.4.

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.

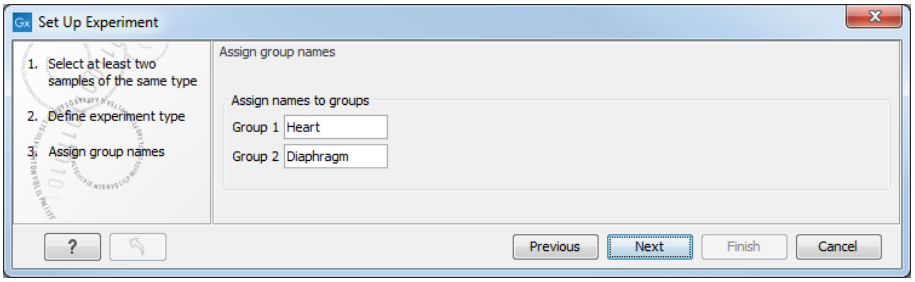


Figure 31.3: *Naming the groups.*

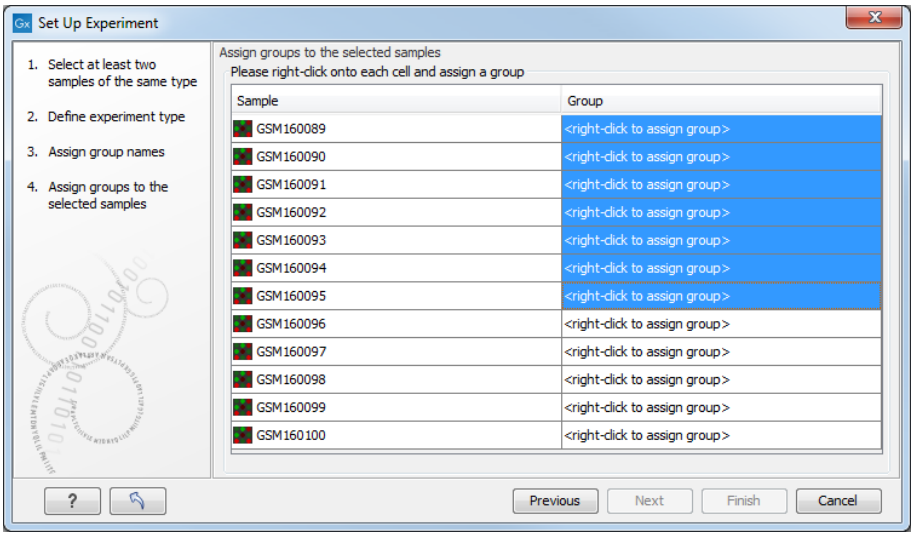


Figure 31.4: *Putting the samples into groups.*

If you have chosen **Paired** in figure 31.2, 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.

31.1.2 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 31.5).

For a general introduction to table features like sorting and filtering, see section 3.2.

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 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

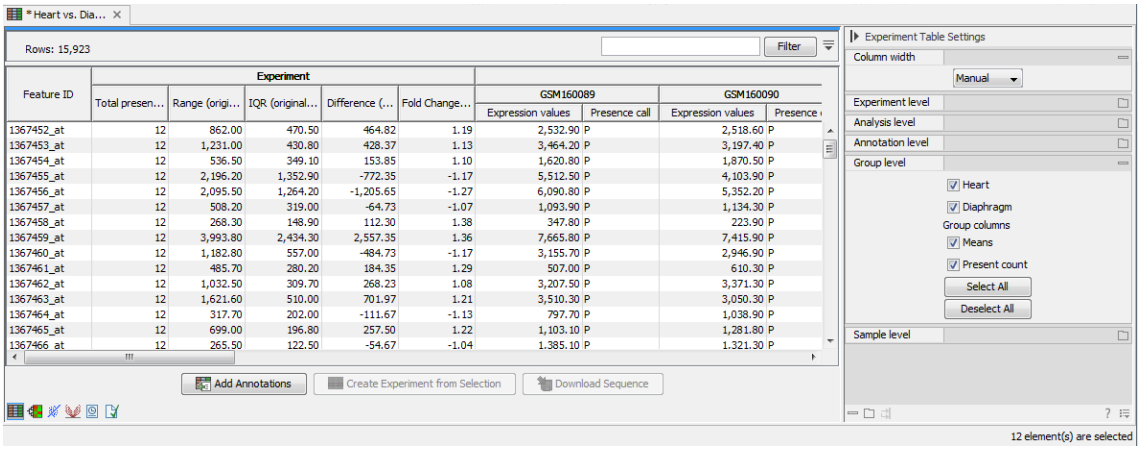


Figure 31.5: *Opening the experiment.*

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 31.6).

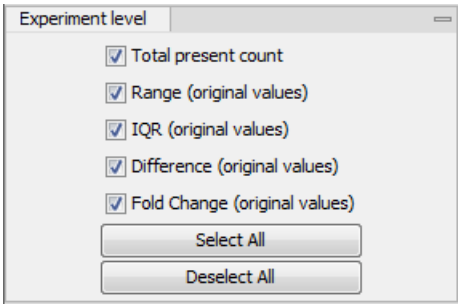


Figure 31.6: *The initial view of the experiment level for a two-group experiment.*

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 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 31.2.3 and 31.2.2, 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 31.5.5. 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 31.5).

An example of Analysis level settings is shown in figure 31.7.

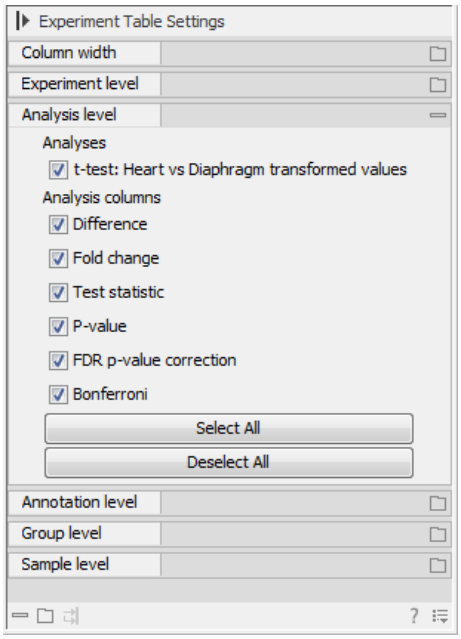


Figure 31.7: *An example of columns available under the Analysis level section.*

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 31.1.3), the annotations will be listed in the **Annotation level** group as shown in figure 31.8.

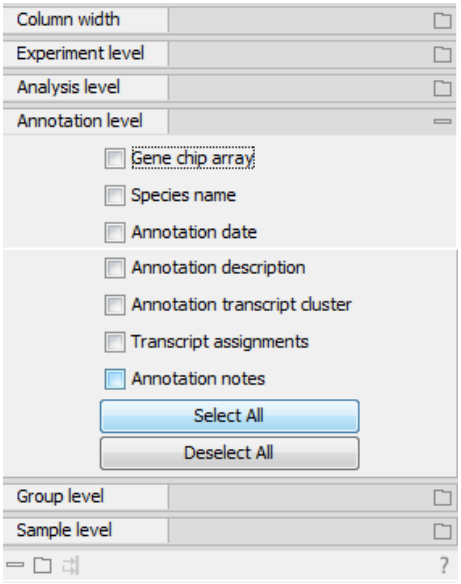


Figure 31.8: *An annotated experiment.*

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 31.5). 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 31.2.3 and 31.2.2, respectively), the means of the normalized and transformed values will also appear.

An example is shown in figure 31.9.

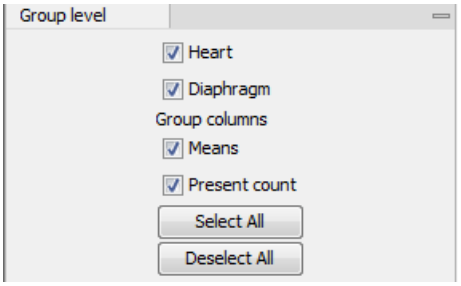


Figure 31.9: *Group level .*

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 31.2.3 and 31.2.2, respectively), the normalized and transformed values will also appear.

An example is shown in figure 31.10.

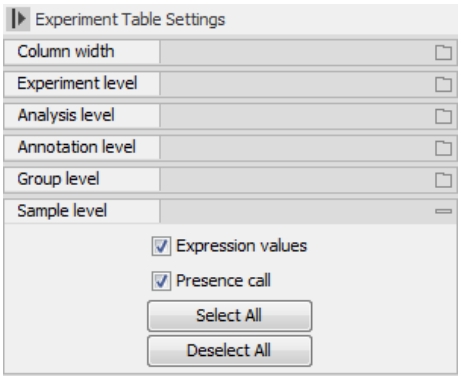


Figure 31.10: *Sample level when transformation and normalization has been performed.*

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 3.2).

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 31.11).

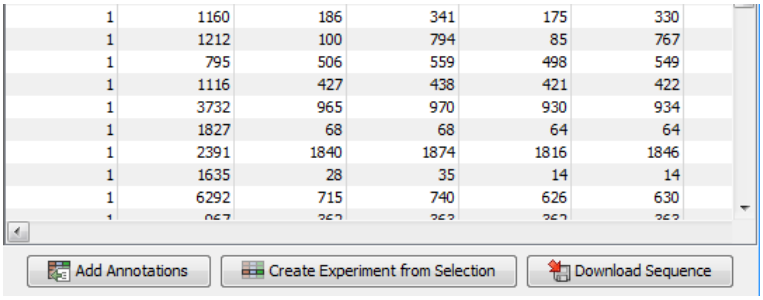


Figure 31.11: *Create a subset of the experiment by clicking the button at the bottom of the experiment table.*

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 31.12).

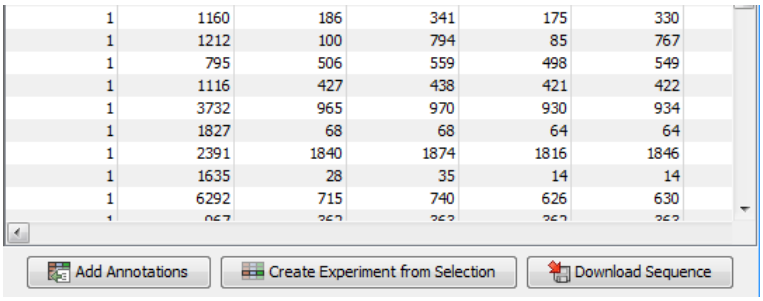


Figure 31.12: *Select sequences and press the download button.*

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 12. You can now use the downloaded sequences for further analysis in the Workbench.

31.1.3 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 31.1.1), or click:

Toolbox | **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 31.1.2. You can also add annotations by pressing the **Add Annotations** ( ) button at the bottom of the table (see figure 31.13).

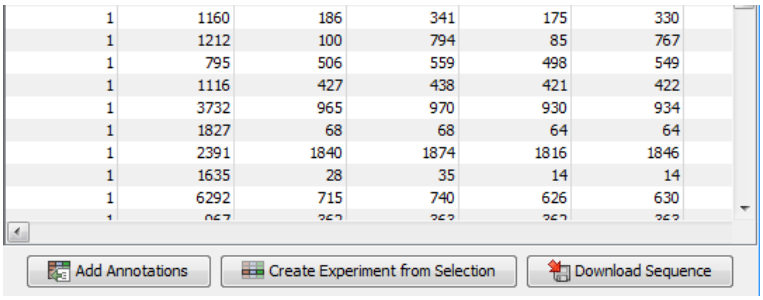


Figure 31.13: *Adding annotations by clicking the button at the bottom of the experiment table.*

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 31.14).

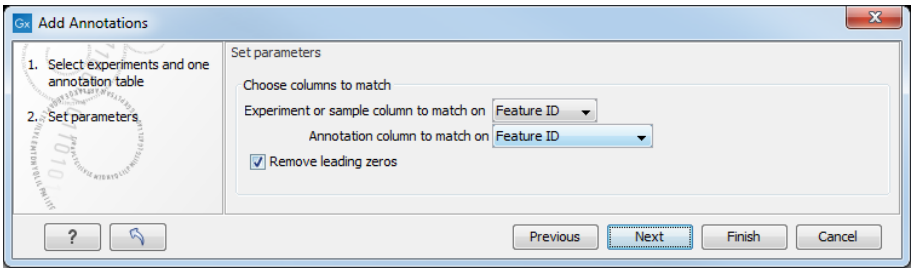


Figure 31.14: *Choosing how to match annotations with samples.*

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 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.

31.1.4 Scatter plot view of an experiment

At the bottom of the experiment table, you can switch between different views of the experiment (see figure 31.15).

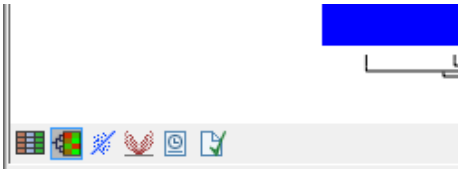


Figure 31.15: *An experiment can be viewed in several ways.*

One of the views is the **Scatter Plot** ( ). 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 31.16.

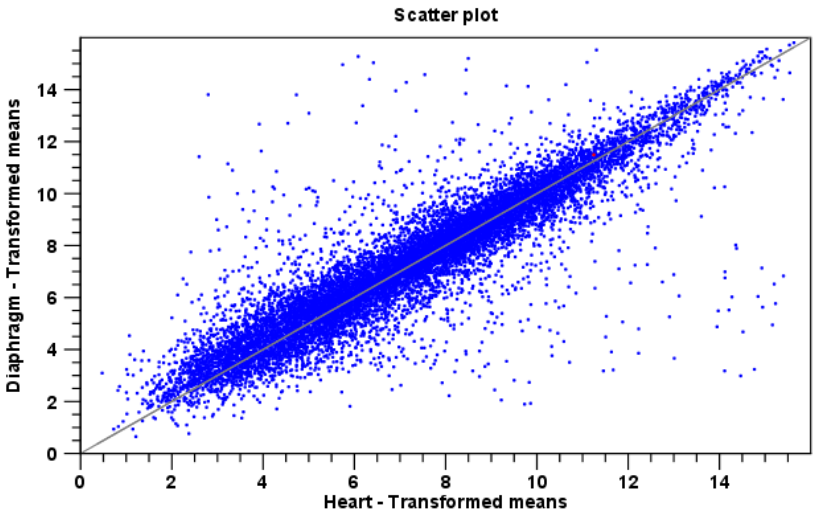


Figure 31.16: *A scatter plot of group means for two groups (transformed expression values).*

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.
- **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).

31.1.5 Cross-view selections

There are a number of different ways of looking at an experiment as shown in figure 31.17).

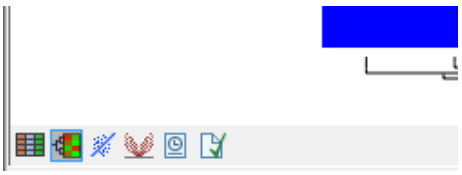


Figure 31.17: *An experiment can be viewed in several ways.*

Beside the **Experiment table** ( ) which is the default view, the views are: **Scatter plot** ( ), **Volcano plot** ( ) and the **Heat map** ( ). By pressing and holding the Ctrl (⌘ on Mac) button while you click one of the view buttons in figure 31.17, 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 31.18.

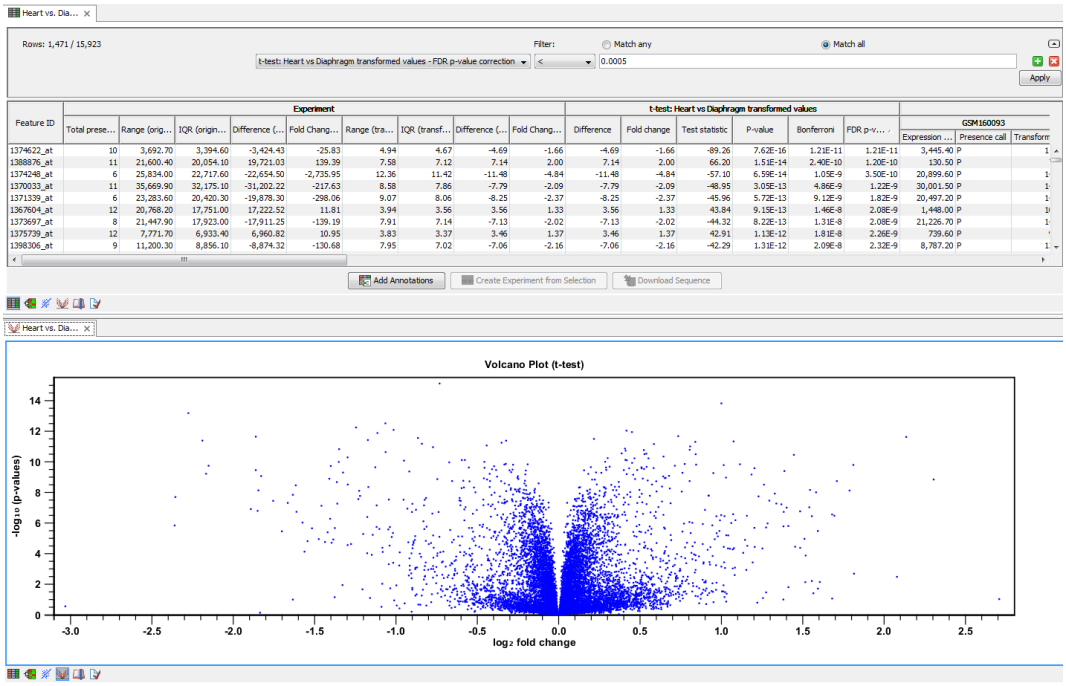


Figure 31.18: *A split view showing an experiment table at the top and a volcano plot at the bottom (note that you need to perform statistical analysis to show a volcano plot, see section 31.5).*

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 heat map, 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 Cntrl 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 31.1.2) you typically want to choose 'Difference'.

31.2 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).

31.2.1 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 31.19).

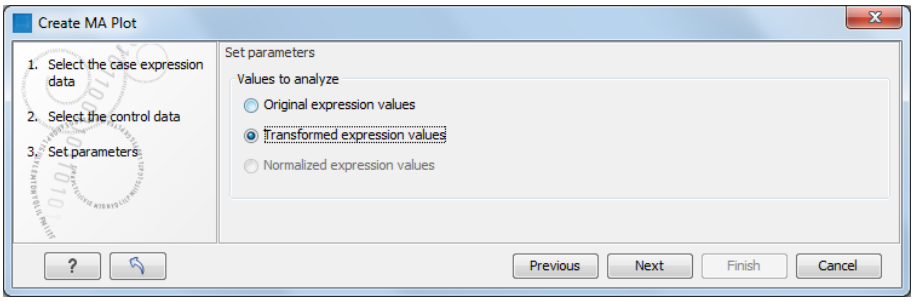


Figure 31.19: *Selecting which version of the expression values to analyze. In this case, the values have not been normalized, so it is not possible to select normalized values.*

In this case, the values have not been normalized, so it is not possible to select normalized values.

31.2.2 Transformation

The *CLC Genomics Workbench* lets you transform expression values based on logarithm and adding a constant:

Toolbox | **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 31.20.

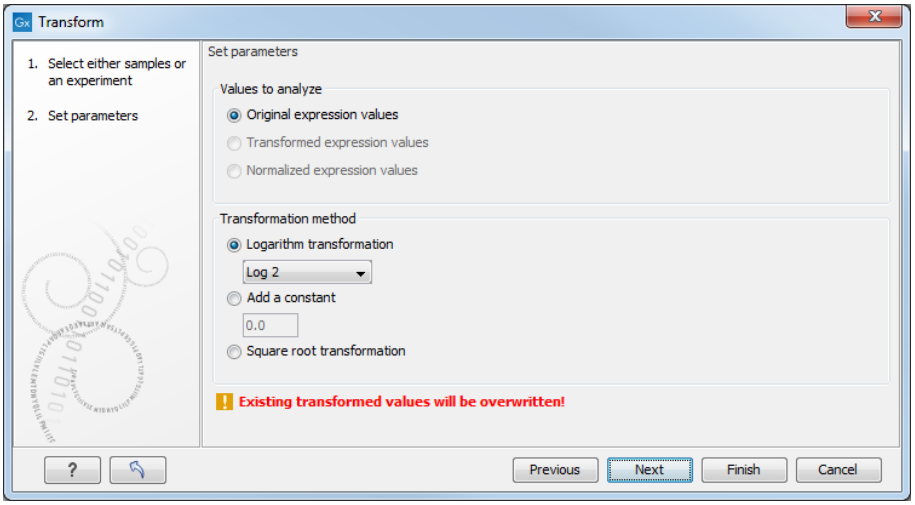


Figure 31.20: *Transforming expression values.*

At the top, you can select which values to transform (see section 31.2.1).

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.

31.2.3 Normalization

The *CLC Genomics Workbench* lets you normalize expression values.

To start the normalization:

Toolbox | **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 31.21.

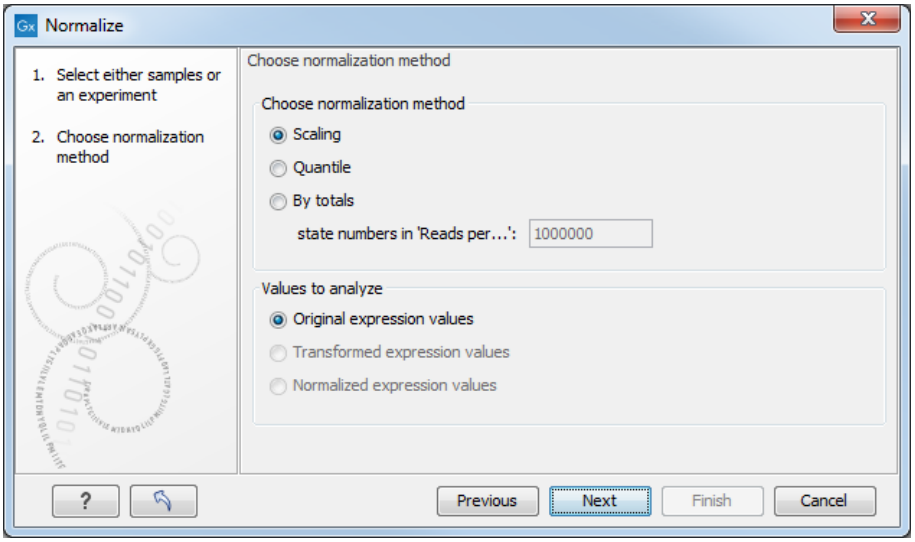


Figure 31.21: *Choosing normalization method.*

At the top, you can choose three kinds of normalization (for mathematical descriptions see [Bolstad et al., 2003]):

- **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').

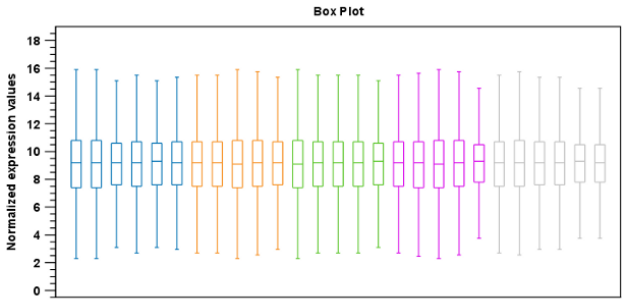


Figure 31.22: *Box plot after scaling normalization.*

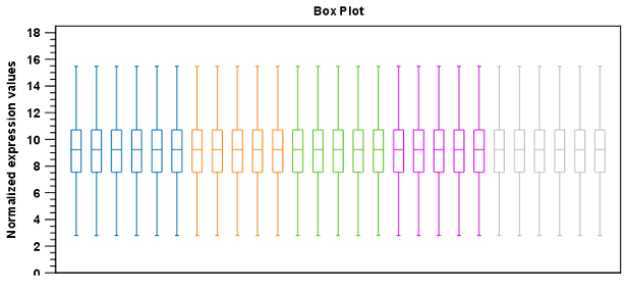


Figure 31.23: *Box plot after quantile normalization.*

Figures 31.22 and 31.23 show the effect on the distribution of expression values when using scaling or quantile normalization, respectively.

At the bottom of the dialog in figure 31.21, you can select which values to normalize (see section 31.2.1).

Clicking **Next** will display a dialog as shown in figure 31.24.

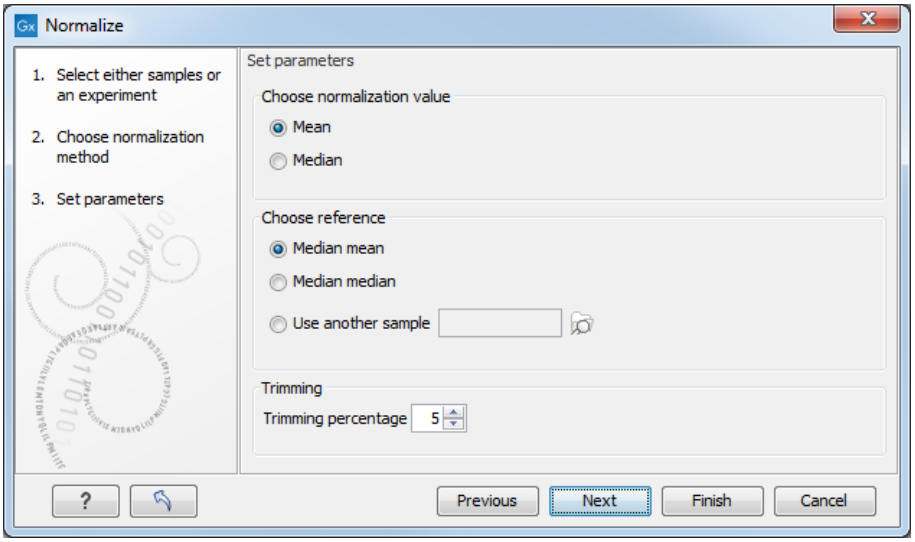


Figure 31.24: *Normalization settings.*

The following parameters can be set:

- **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.

31.3 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.

31.3.1 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 it's 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:

Toolbox | **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 31.25.

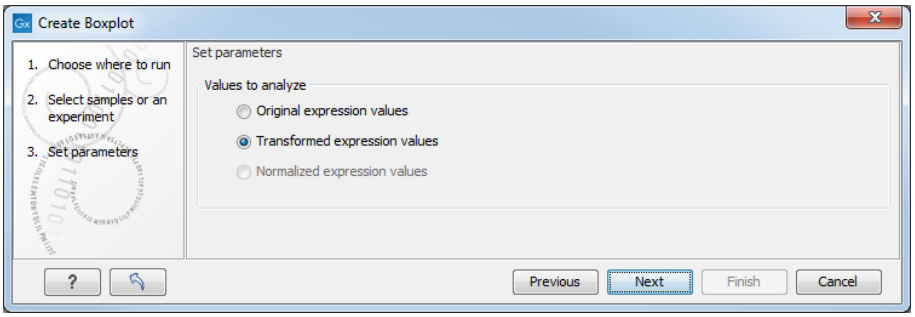


Figure 31.25: *Choosing values to analyze for the box plot.*

Here you select which values to use in the box plot (see section 31.2.1).

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 31.26.

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.

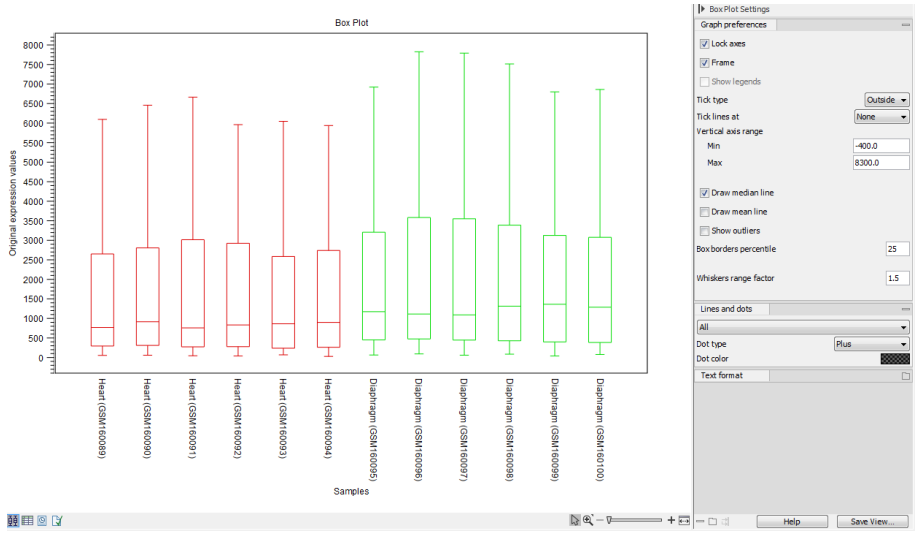


Figure 31.26: *A box plot of 12 samples in a two-group experiment, colored by group.*

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 preferences**, you can adjust the general properties of the box plot (see figure 31.27).

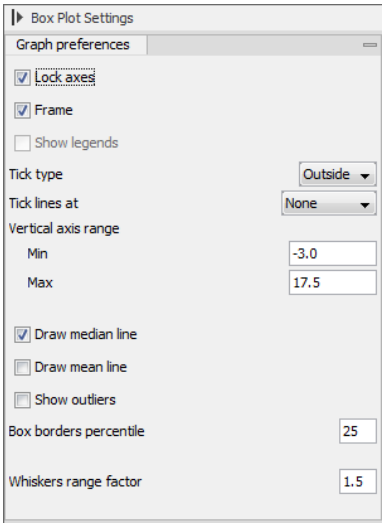


Figure 31.27: *Graph preferences for a box plot.*

- **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 31.28).

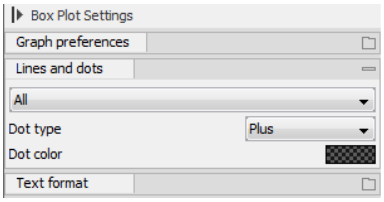


Figure 31.28: *Lines and dot preferences for a box plot.*

- **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 31.29, you can see a box plot for an experiment with 5 groups and 27 samples.

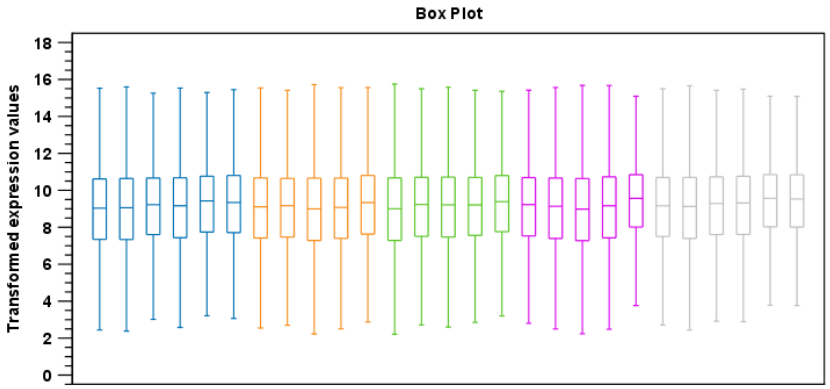


Figure 31.29: *Box plot for an experiment with 5 groups and 27 samples.*

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 31.30 shows a box plot for the same experiment after quantile normalization: the distributions have been brought into par.

In figure 31.31 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.

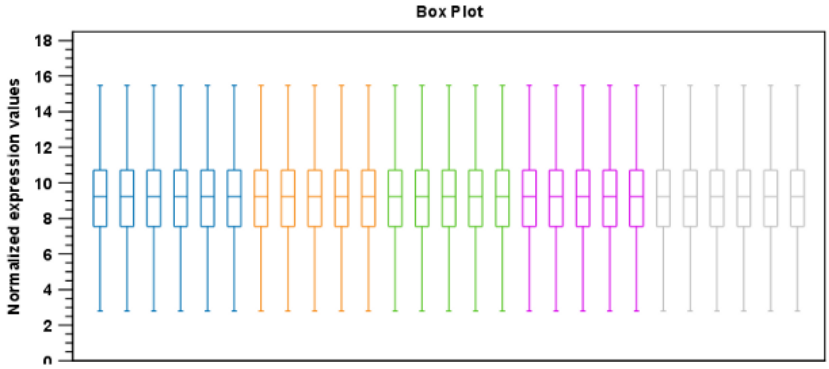


Figure 31.30: *Box plot after quantile normalization.*

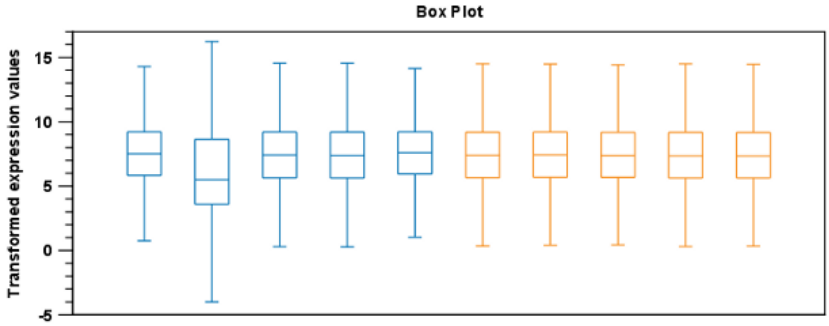


Figure 31.31: *Box plot for a two-group experiment with 5 samples.*

31.3.2 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:

Toolbox | **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 31.32. 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.

There are three kinds of **Distance measures**:

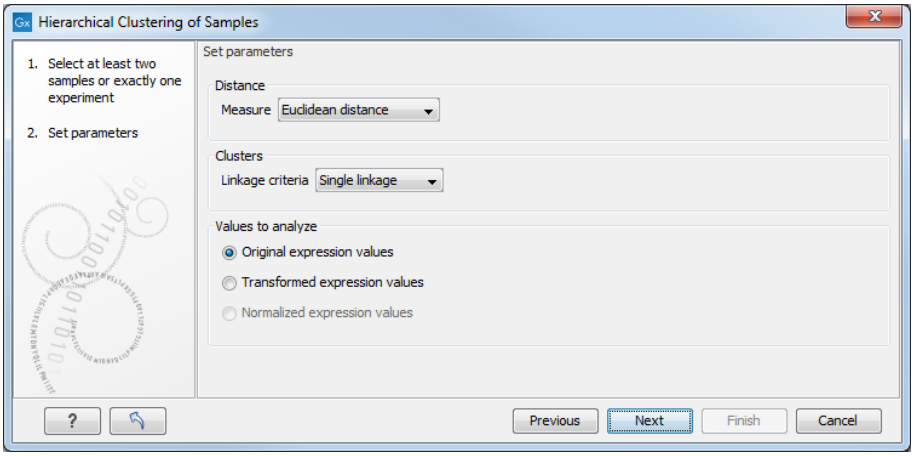


Figure 31.32: *Parameters for hierarchical clustering of samples.*

- **Euclidean distance**. The ordinary distance between two points - the length of the segment connecting them. If $u = (u_1, u_2, \ldots, u_n)$ and $v = (v_1, v_2, \ldots, v_n)$, then the Euclidean distance between u and v is

$$|u - v| = \sqrt{\sum_{i=1}^{n}(u_i - v_i)^2}.$$

- **1 - Pearson correlation**. The Pearson correlation coefficient between two elements $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$ is defined as

$$r = \frac{1}{n-1}\sum_{i=1}^{n}(\frac{x_i - \overline{x}}{s_x}) * (\frac{y_i - \overline{y}}{s_y})$$

  where $\overline{x}/\overline{y}$ is the average of values in x/y and s_x/s_y is the sample standard deviation of these values. It takes a value $\in [-1, 1]$. Highly correlated elements have a high absolute value of the Pearson correlation, and elements whose values are un-informative about each other have Pearson correlation 0. Using $1 - |Pearsoncorrelation|$ as distance measure means that elements that are highly correlated will have a short distance between them, and elements that have low correlation will be more distant from each other.

- **Manhattan distance**. The Manhattan distance between two points is the distance measured along axes at right angles. If $u = (u_1, u_2, \ldots, u_n)$ and $v = (v_1, v_2, \ldots, v_n)$, then the Manhattan distance between u and v is

$$|u - v| = \sum_{i=1}^{n}|u_i - v_i|.$$

The possible cluster linkages are:

- **Single linkage**. The distance between two clusters is computed as the distance between the two closest elements in the two clusters.
- **Average linkage**. The distance between two clusters is computed as the average distance between objects from the first cluster and objects from the second cluster. The averaging is performed over all pairs (x, y), where x is an object from the first cluster and y is an object from the second cluster.
- **Complete linkage**. The distance between two clusters is computed as the maximal object-to-object distance $d(x_i, y_j)$, where x_i comes from the first cluster, and y_j comes from the second cluster. In other words, the distance between two clusters is computed as the distance between the two farthest objects in the two clusters.

At the bottom, you can select which values to cluster (see section 31.2.1).

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 a stand-alone new heat map rather than added into the input experiment table.

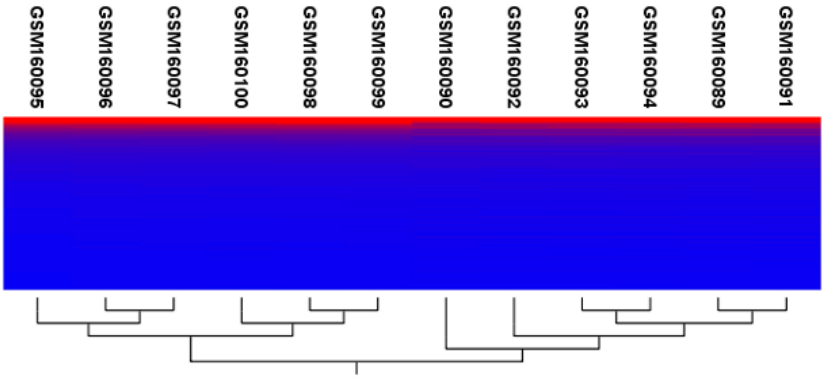


Figure 31.33: *Sample clustering.*

Result of hierarchical clustering of samples

The result of a sample clustering is shown in figure 31.33.

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 31.34).

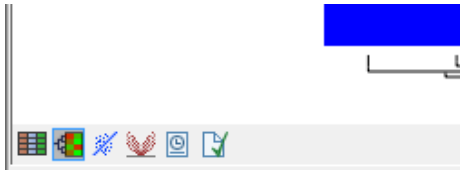


Figure 31.34: *Showing the hierarchical clustering of an experiment.*

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 31.33, 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 31.4.1).

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, researches 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 unclean 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 31.35).

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 31.36).

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.

- **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.

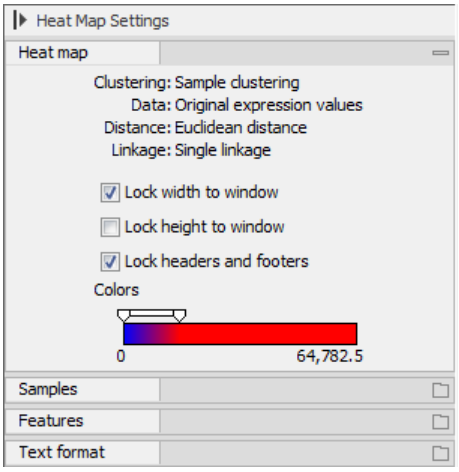


Figure 31.35: *Side Panel of heat map.*

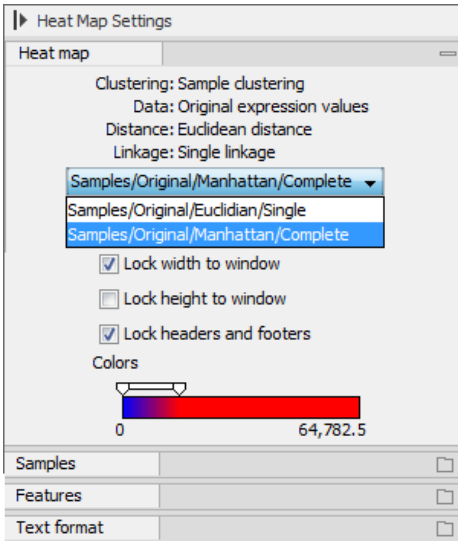


Figure 31.36: *When more than one clustering has been performed, there will be a list of heat maps to choose from.*

- **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 heatmap. 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.

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).

31.3.3 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:

Toolbox | **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 31.37.

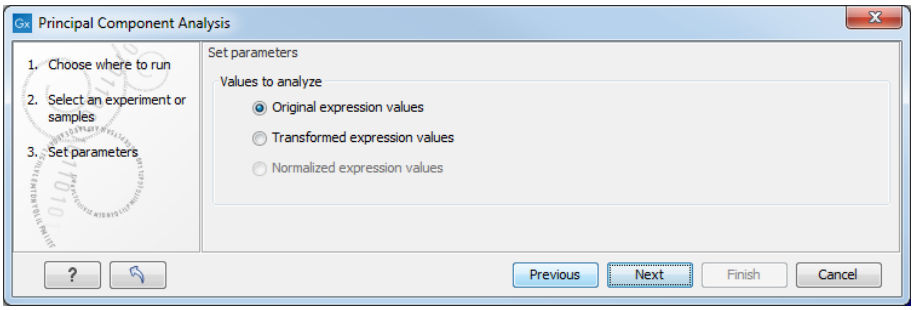


Figure 31.37: *Selecting which values the principal component analysis should be based on.*

In this dialog, you select the values to be used for the principal component analysis (see section 31.2.1).

Click **Finish** to start the tool.

Principal component analysis plot

This will create a principal component plot as shown in figure 31.38.

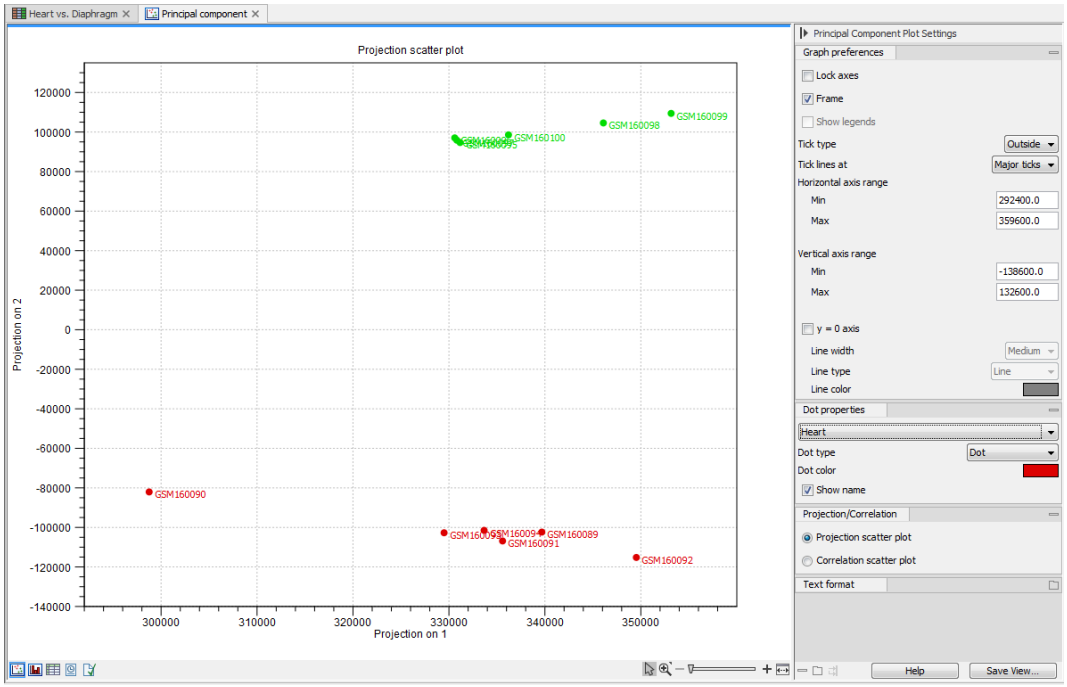


Figure 31.38: *A principal component analysis.*

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* 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 31.38 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 31.38, 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).

31.4 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.

31.4.1 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:

Toolbox | **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 31.1.2.

Clicking **Next** will display a dialog as shown in figure 31.39. 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

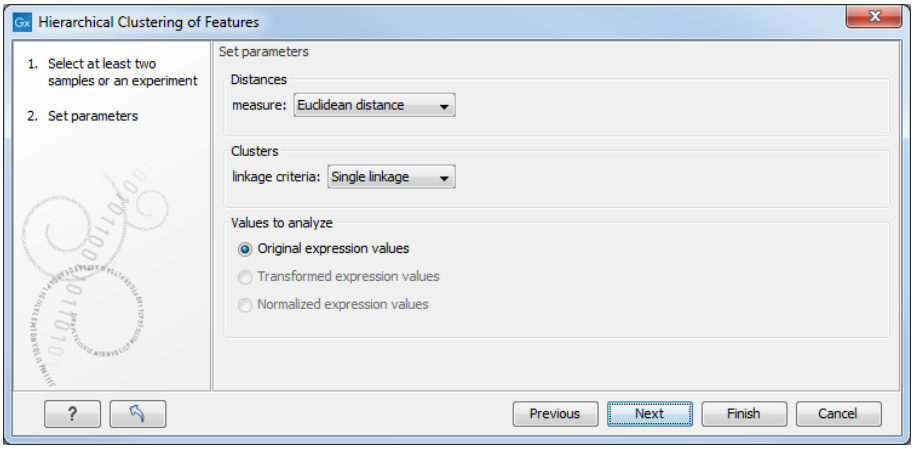


Figure 31.39: *Parameters for hierarchical clustering of features.*

cluster linkage specifies how you want the distance between two clusters, each consisting of a number of features, to be calculated.

There are three kinds of **Distance measures**:

- **Euclidean distance**. The ordinary distance between two points - the length of the segment connecting them. If $u = (u_1, u_2, \ldots, u_n)$ and $v = (v_1, v_2, \ldots, v_n)$, then the Euclidean distance between u and v is

$$|u - v| = \sqrt{\sum_{i=1}^{n}(u_i - v_i)^2}.$$

- **1 - Pearson correlation**. The Pearson correlation coefficient between two elements $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$ is defined as

$$r = \frac{1}{n-1}\sum_{i=1}^{n}(\frac{x_i - \overline{x}}{s_x}) * (\frac{y_i - \overline{y}}{s_y})$$

  where $\overline{x}/\overline{y}$ is the average of values in x/y and s_x/s_y is the sample standard deviation of these values. It takes a value $\in [-1, 1]$. Highly correlated elements have a high absolute value of the Pearson correlation, and elements whose values are un-informative about each other have Pearson correlation 0. Using $1 - |Pearsoncorrelation|$ as distance measure means that elements that are highly correlated will have a short distance between them, and elements that have low correlation will be more distant from each other.

- **Manhattan distance**. The Manhattan distance between two points is the distance measured along axes at right angles. If $u = (u_1, u_2, \ldots, u_n)$ and $v = (v_1, v_2, \ldots, v_n)$, then the Manhattan distance between u and v is

$$|u - v| = \sum_{i=1}^{n}|u_i - v_i|.$$

The possible cluster linkages are:

- **Single linkage**. The distance between two clusters is computed as the distance between the two closest elements in the two clusters.
- **Average linkage**. The distance between two clusters is computed as the average distance between objects from the first cluster and objects from the second cluster. The averaging is performed over all pairs (x, y), where x is an object from the first cluster and y is an object from the second cluster.
- **Complete linkage**. The distance between two clusters is computed as the maximal object-to-object distance $d(x_i, y_j)$, where x_i comes from the first cluster, and y_j comes from the second cluster. In other words, the distance between two clusters is computed as the distance between the two farthest objects in the two clusters.

At the bottom, you can select which values to cluster (see section 31.2.1).

Click **Finish** to start the tool.

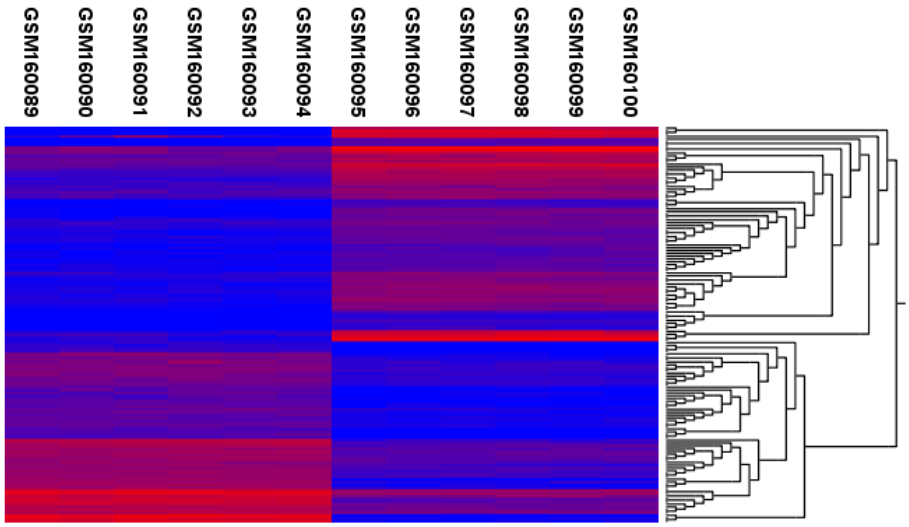


Figure 31.40: *Hierarchical clustering of features.*

Result of hierarchical clustering of features

The result of a feature clustering is shown in figure 31.40.

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 31.41).

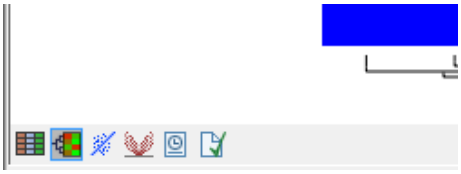


Figure 31.41: *Showing the hierarchical clustering of an experiment.*

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 31.40). In the heatmap each row corresponds to a feature and each column to a sample. The 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 31.42).

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 31.43).

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.

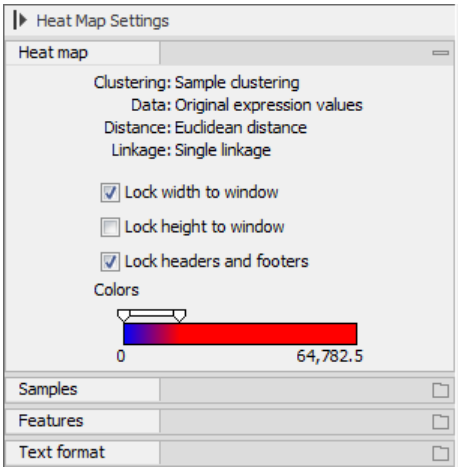


Figure 31.42: *Side Panel of heat map.*

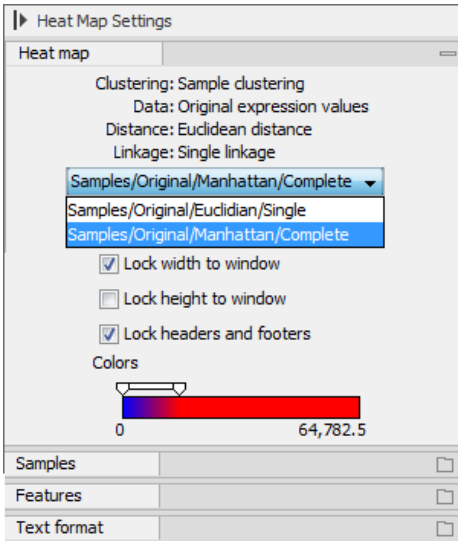


Figure 31.43: *When more than one clustering has been performed, there will be a list of heat maps to choose from.*

- **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 heatmap. 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.

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).

31.4.2 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.

Toolbox | **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 31.1.2.

Clicking **Next** will display a dialog as shown in figure 31.44.

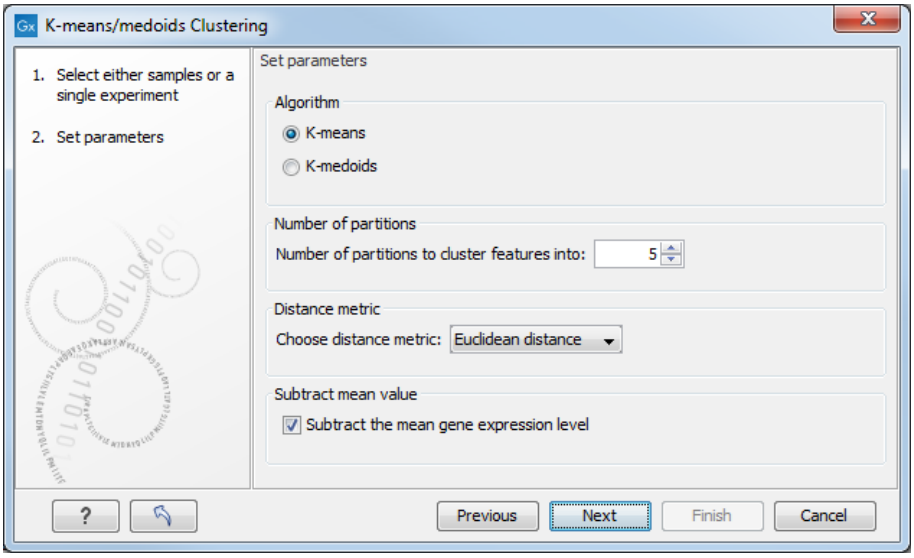


Figure 31.44: *Parameters for k-means/medoids clustering.*

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 $S_i, i = 1, 2, \ldots, k$ 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 $S_i, i = 1, 2, \ldots, k$ 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-agorithm 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 $u = (u_1, u_2, \ldots, u_n)$ and $v = (v_1, v_2, \ldots, v_n)$, 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 $u = (u_1, u_2, \ldots, u_n)$ and $v = (v_1, v_2, \ldots, v_n)$, 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 31.45.

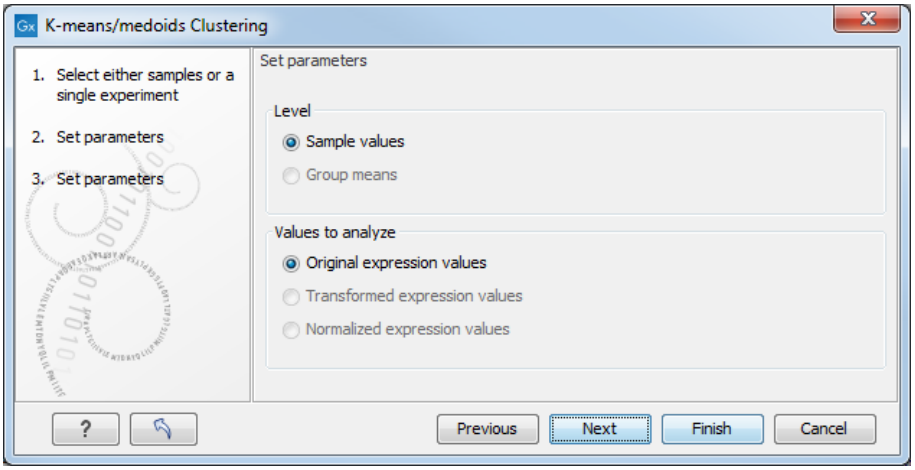


Figure 31.45: *Parameters for k-means/medoids clustering.*

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 31.2.1).

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 31.44) - there is one graph per cluster. Using drag and drop as explained in section 2.1.5, you can arrange the views to see more than one graph at the time.

Figure 31.46 shows an example where four clusters have been arranged side-by-side.

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 that 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 31.1.5.

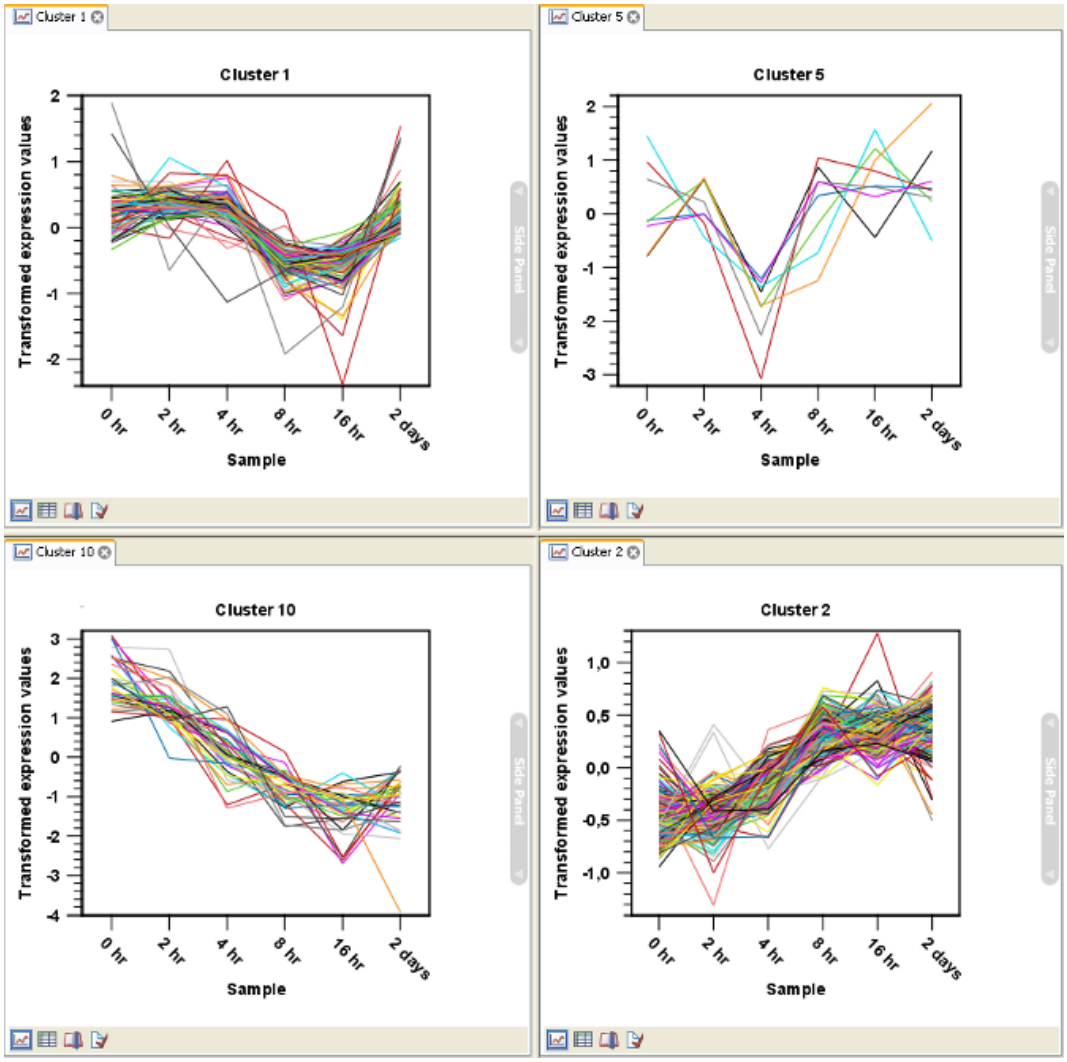


Figure 31.46: *Four clusters created by k-means/medoids clustering.*

31.5 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 31.5.1 and section 31.5.2) and tests that assume that the data is real-valued, has Gaussian distributions and compare means (described in section 31.5.3).

To run the statistical analysis:

Microarray Analysis ( )| **Statistical Analysis** ( ) | **Empirical Analysis of DGE** ( )

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 31.1.1).

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 Empirical analysis of DGE, 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.

31.5.1 Empirical analysis of DGE

The Empirical analysis of DGE tool implements the 'Exact Test' for two-group comparisons developed by Robinson and Smyth [Robinson and Smyth, 2008] and incorporated in the edgeR Bioconductor package [Robinson et al., 2010]. The test is applicable to count data only, and is designed specifically to deal with situations in which *many* features are studied simultaneously (e.g. genes in a genome) but where only a *few* biological replicates are available for each of the experimental groups studied.

The test uses the raw counts, and implicitly carries out normalization and transformation of these counts (see below for details). It is based on the assumption that the count data follows a Negative Binomial distribution, which in contrast to the Poisson distribution has the characteristic that it allows for a non-constant mean-variance relationship. The test is also appropriate for larger numbers of samples.

The 'Exact Test' of Robinson and Smyth is similar to Fisher's Exact Test, but also accounts for overdispersion caused by biological variability. Whereas Fisher's Exact Test compares the counts in one sample against those of another, the 'Exact Test' compares the counts in one *set* of count samples against those in another *set* of count samples. This is achieved by replacing the Hypergeometric distributions of Fisher's Exact Test by Negative binomial distributions, whereby the variability within each of the two groups of samples compared is taken into account. This only works if the dispersions in the two groups compared are identical. As this cannot generally be assumed to be the case for the *original* (nor for the normalized) data, pseudodata for which the dispersion *is* identical is generated from the original data, and the test is carried out on this pseudodata. The generation of the pseudodata is performed simultaneously with the estimation of the dispersion, in an iterative procedure called quantile-adjusted conditional maximum likelihood. Either a single common dispersion for all features may be assumed (as in [Robinson and Smyth, 2008]), or it may be assumed that the dispersion for each feature (e.g. gene) is a 'weighted average' of the common dispersion and feature (e.g. gene) specific dispersions (as suggested in [Robinson and Smyth, 2007]). The weight given to each of the components depends on the number of samples in the groups: the more samples there are in the groups, the higher the weight will be given to the gene-specific component.

The Exact Test in the edgeR Bioconductor package provides the user with the option to set a large number of parameters. The implementation of the 'Empirical analysis of DGE' algorithm in the Genomics Workbench uses for the most parts the default settings in the edgeR package, version 3.4.0. A detailed outline of the parameter settings is given in section 31.5.1).

Empirical analysis of DGE - implementation parameters

The 'Empirical analysis of DGE' algorithm in the *CLC Genomics Workbench* is a re-implementation of the "Exact Test", available as part of the edgeR Bioconductor package.

The parameter values used in the *CLC Genomics Workbench* implementation are the default values for the equivalent parameters in the edgeR Bioconductor implementation in all but one case. The exception is the estimateCommonDisp tol parameter, where the default is more stringent than that of edgeR. The advantage of using a more stringent value for this parameter is that the results will be more accurate. The disadvantage is that the algorithm will be slightly slower, however according to our performance tests, this change has only a marginal impact on the run time of the tool.

The parameter values used in the *CLC Genomics Workbench* implementation, with reference to the edgeR function names for clarity, are provided in the table below.

Function in BioC package	Parameter name	Value used and comments
calcNormFactors	method	"TMM"
	refColumn	NULL (automatically selected)
	logratioTrim	0.3
	sumTrim	0.05
	doWeighting	TRUE
	Acutoff	-1e10
estimateCommonDisp	tol	1e-14 (default in edgeR: 1e-6)
	rowsum.filter	Set by user in wizard ("Total count filter cutoff", default 5)
estimateTagewiseDisp	prior.df	10
	trend	"movingave"
	span	NULL
	method	"grid"
	grid.length	11
	grid.range	c(-6, 6)
mglmOneGroup	maxit	50
	tol	1e-10
aveLogCPM	prior.count	2
	dispersion	0.05
exactTest	pair	Set by user in wizard ("Exact test comparisons")
	dispersion	"auto" (tagwise if available, otherwise common)
	rejection.region	"doubletail"
	big.count	900
	prior.count	0.125

Running the Empirical analysis of DGE

First, find the **Empirical analysis of DGE** tool:

Toolbox | **Microarray Analysis** ( )| **Statistical Analysis** ( ) | **Empirical Analysis of DGE** ( )

The original count data for a full expression experiment are the expected input to the Empirical Analysis of DGE tool.

When Experiments created within the Workbench are used as input, the original count values are always used. Columns of such Experiments that contain transformed or normalized values are ignored.

If expression values are being imported from outside the Workbench for use with this test, the data should be original (non-transformed, non-normalized) counts.

Whether the data has been generated in the Workbench or outside the Workbench and imported, the full set of expression results should be used. Please do not run this test on a subset of values from the original sample data.

The reason that the complete set of original count data for samples should be used as input to this test is that the algorithm assumes that the counts on which it operates are Negative Binomially distributed. It implicitly normalizes and transforms these counts, so if the counts have been altered prior to submitting them to the Empirical Analysis of DGE tool, this assumption is likely to be compromised.

When running the Empirical analysis of DGE tool in the Genomics workbench, the user is asked to specify two parameters related to the estimation of the dispersion (figure 31.47). Of these, the 'Total count filter cut-off' specifies which features should be considered when estimating the common dispersion component. Features for which the counts across all samples are low are likely to contribute mostly with noise to the estimation, and features with a lower cummulative count across samples than the value specified will be ignored. When the check-box 'Estimate tag-wise dispersions' is checked, the dispersion estimate for each gene will be a weighted combination of the tag-wise and common dispersion, if the check-box is un-ticked the common dispersion will be used for all genes.

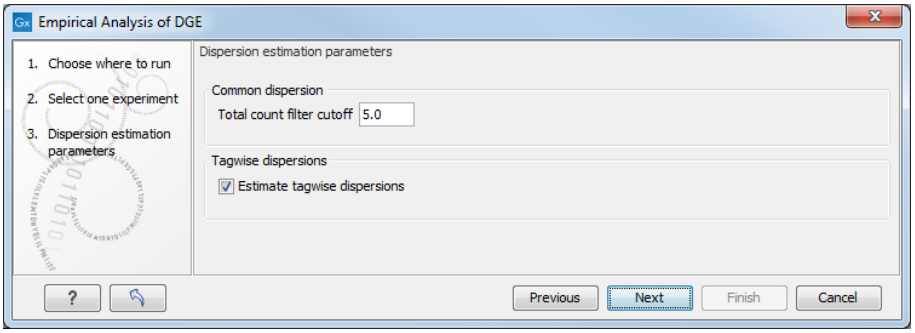


Figure 31.47: *Empirical analysis of DGE: setting the parameters related to dispersion.*

The Empirical analysis of DGE may be carried out between all pairs of groups (by clicking the 'All pairs' button) or for each group against a specified reference group (by clicking the 'Against reference' button) (figure 31.48). 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). Foe example, the All pairs option should be selected when you wish to perform the test of equality for group means for all of the pairs, e.g. if you would like to compare different tissues where each tissue is represented in a group. In this case there is no reference group, so the following comparisons will be performed:

- liver vs heart
- liver vs lung
- heart vs lung

The Against reference option should be selected when you wish to perform the test of equality for group means against one group, the reference, rather than all groups as above. Against reference could be used if you have a wild type and some mutant groups, e.g. Wild type, Mutant 1 and Mutant 2. In this case you might be interested in comparing the mutants to the wild type, but comparing the mutants to each other is not of interest. In this case the Wild Type group is considered the reference and the comparisons will be perfomed:

- Wild type vs Mutant 1
- Wild type vs Mutant 2

Note that with the Against reference option fewer comparisons are made, as in the above example where Mutant 1 vs. Mutant 2 is not considered.

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].

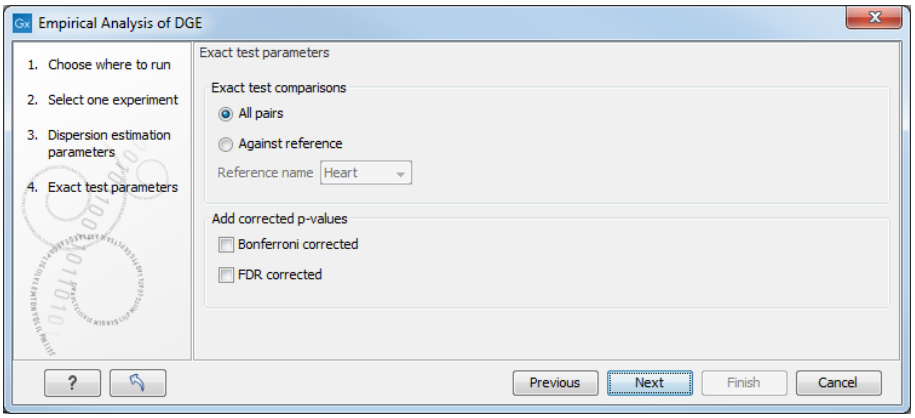


Figure 31.48: *Empirical analysis of DGE: setting comparisons and corrected p-value options.*

When the Empirical analysis of DGE is run three columns will be added to the experiment table for each pair of groups that are analyzed: the 'P-value', 'Fold change' and 'Weighted difference' columns. The 'P-value' holds the p-value for the Exact test. The 'Fold Change' and 'Weighted difference' columns are both calculated from the estimated relative abundances, which are derived internally in the Exact Test algorithm. They depend on both the sizes (depth of coverage/library size) of the samples, the magnitude of the counts and on the estimated negative binomial dispersion, so they cannot be obtained from the original counts by simple algebraic calculations.

The 'Fold Change' will tell you how many times bigger the relative abundance of group 2 is relative to that of group 1. If the relative abundance of group 2 is bigger than that of group 1 the fold change is the relative abundance of group 2 divided by that of group 1. If the relative abundance of group 2 is smaller than that of group 1 the fold change is the relative abundance of group 1 divided by that of group 2 with a negative sign. The 'weighted difference' column contains the difference between the relative abundance of group 2 and the relative abundance of group 1. In addition to the three automatically added columns, columns containing the Bonferroni and FDR corrected p-values will be added if that was specified by the user.

31.5.2 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 31.1.1), 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 you 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.

31.5.3 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 31.49.

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

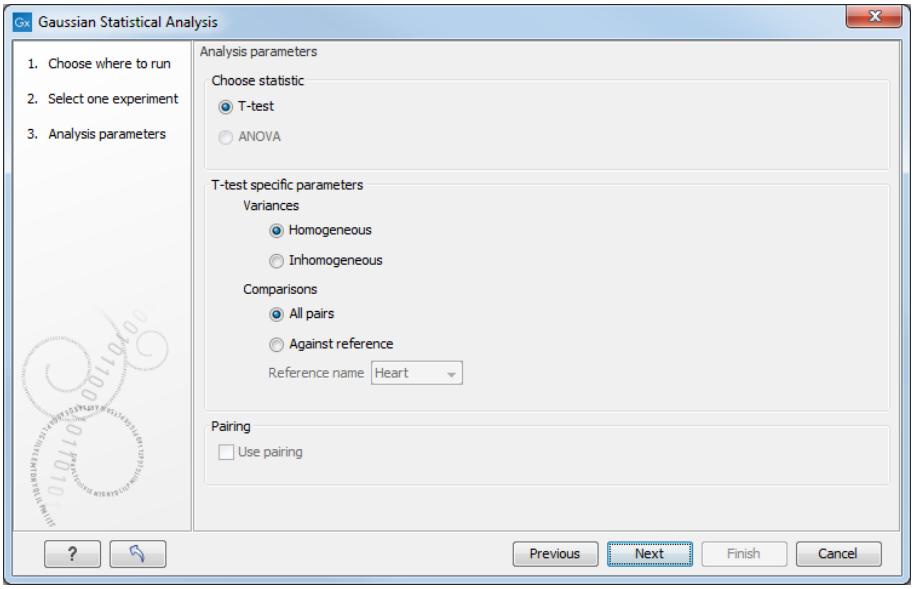


Figure 31.49: *Selecting a t-test.*

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 31.1.1) 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** as described above, 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 31.1.1) 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.

31.5.4 Corrected p-values

Clicking **Next** will display a dialog as shown in figure 31.50.

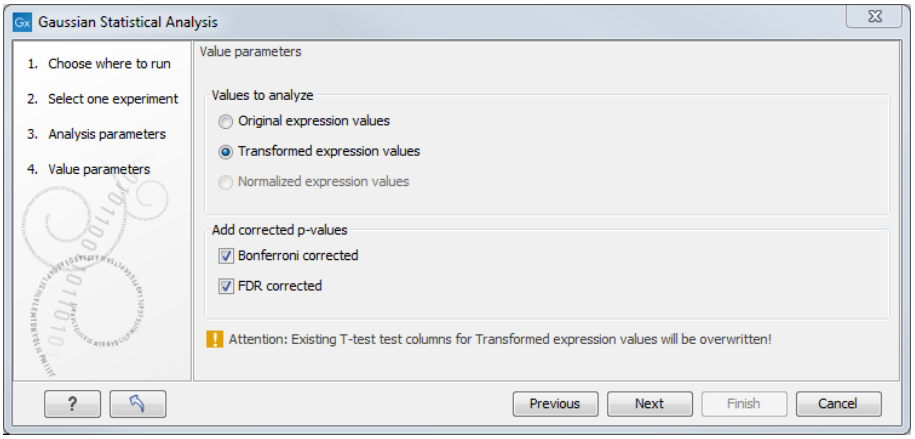


Figure 31.50: *Additional settings for the statistical analysis.*

At the top, you can select which values to analyze (see section 31.2.1).

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.

31.5.5 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 31.1.2). 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 3.2).

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 presented in section 31.1.4, 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 31.51.

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'

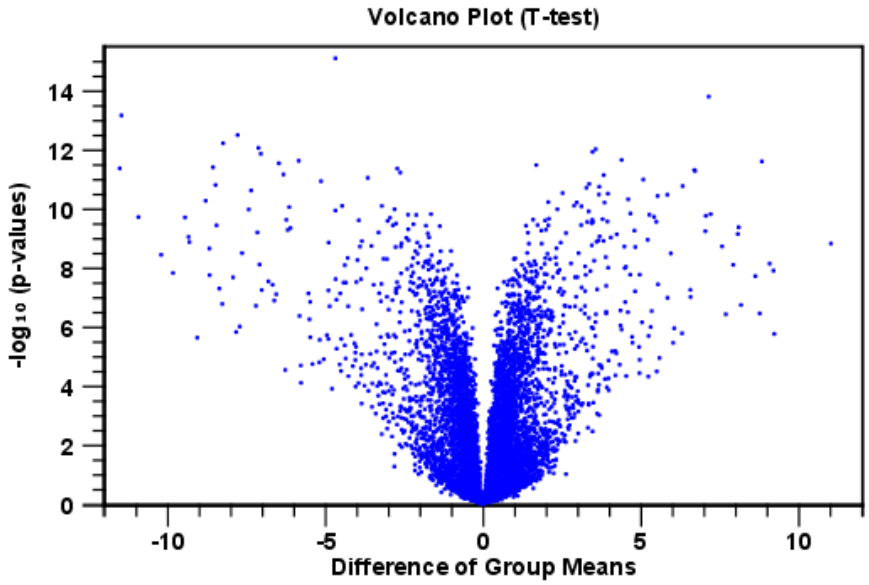


Figure 31.51: *Volcano plot.*

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 you data (Read the note on fold change in section 31.1.2).

The larger the difference in expression of a feature, the more extreme it's 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 31.1.2.

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).

31.6 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 31.1.3.

31.6.1 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 GOstats test of [Falcon and Gentleman, 2007].

Toolbox | **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 31.1.2).

Click **Next**. This will display the dialog shown in figure 31.52.

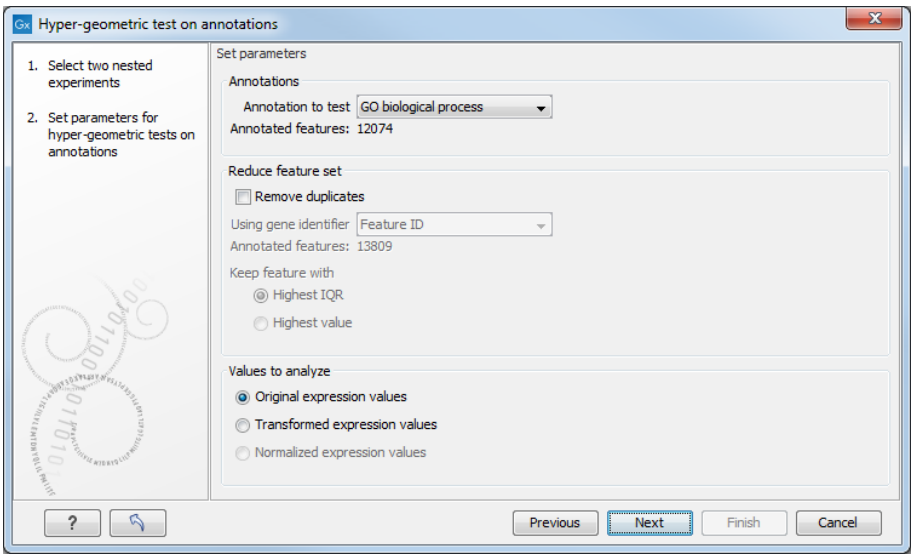


Figure 31.52: *Parameters for performing a hypergeometric test on annotations.*

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 31.2.1). 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 31.53.

*HyperG tests ...

Rows: 8,733 Hyper-Geometric tests for annotation category associations Filter

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 31.53: *The result of testing 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.
- **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.

Note that when testing for the significance of a particular GO term, we take into account that GO has a hierarchical structure. See section 30.8.1 for a detailed description on how to interpret potential discrepancies in the number of features in your results and the original GAF file.

31.6.2 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 than 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 some times 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 if differential expression, prior to running GSEA on it.

Toolbox | **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 31.54.

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.

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:

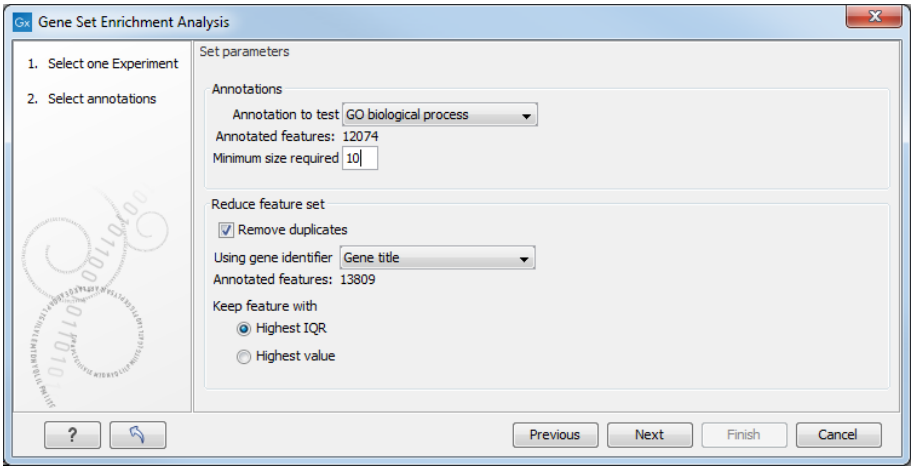


Figure 31.54: *Gene set enrichment analysis on GO biological process.*

- **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 31.55.

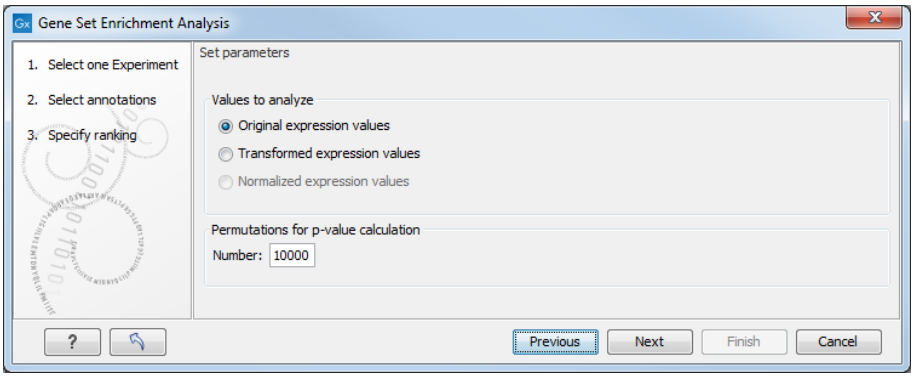


Figure 31.55: *Gene set enrichment analsysis parameters.*

At the top, you can select which values to analyze (see section 31.2.1).

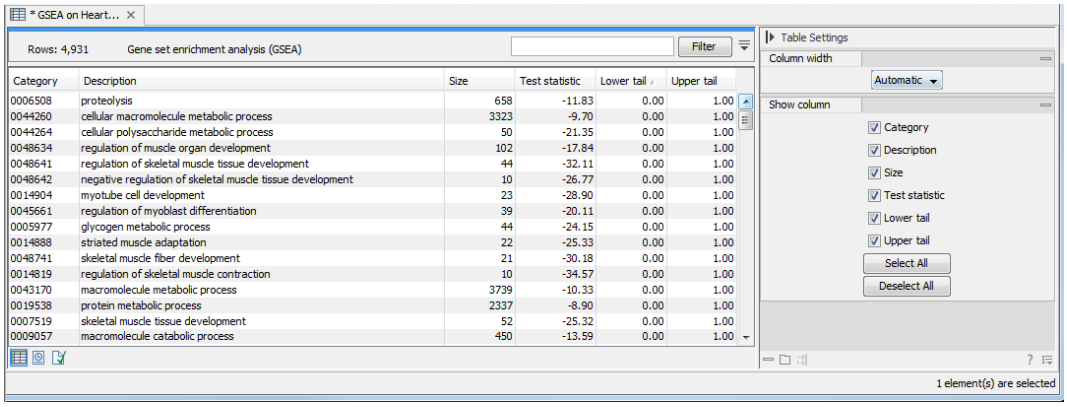
Below, you can set the **Permutations for p-value calculation**. For the GSEA test a p-value is calculated by permutation: p permuted data sets are generated, each consisting of the original features, but with the test statistics permuted. The GSEA test is run on each of the permuted data sets. The test 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).

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 31.56.

The table shows the following information:

- **Category**. This is the identifier for the category.



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 31.56: *The result of gene set enrichment analysis on GO biological process.*

- **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. Note that when testing for the significance of a particular GO term, we take into account that GO has a hierarchical structure. See section 30.8.1 for a detailed description on how to interpret potential discrepancies in the number of genes in your results and the original GAF file.

31.7 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.

31.7.1 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:

Toolbox | **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 31.57.

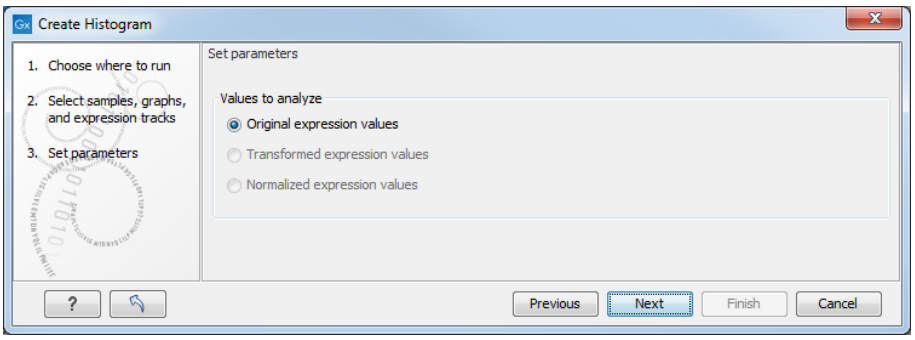


Figure 31.57: *Selecting which values the histogram should be based on.*

In this dialog, you select the values to be used for creating the histogram (see section 31.2.1).

Click **Finish** to start the tool.

Viewing histograms

The resulting histogram is shown in a figure 31.58

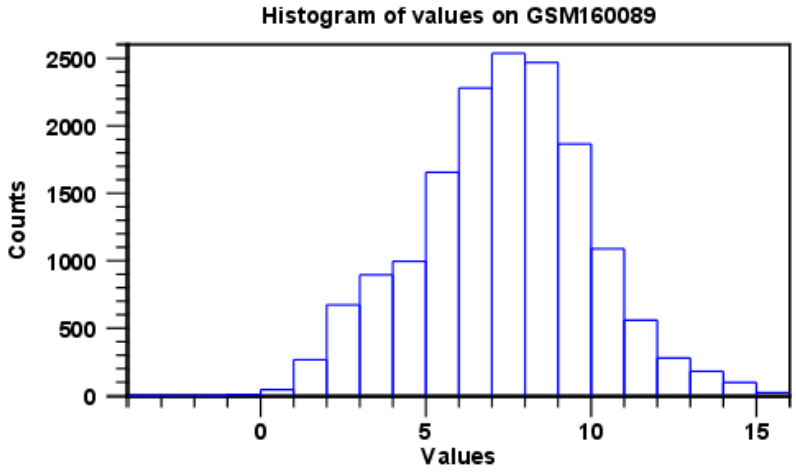


Figure 31.58: *Histogram showing the distribution of transformed expression values.*

The histogram shows the expression value on the x axis (in the case of figure 31.58 the transformed expression values) and the counts of these values on the y axis.

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 31.59.

The table lists the following properties:

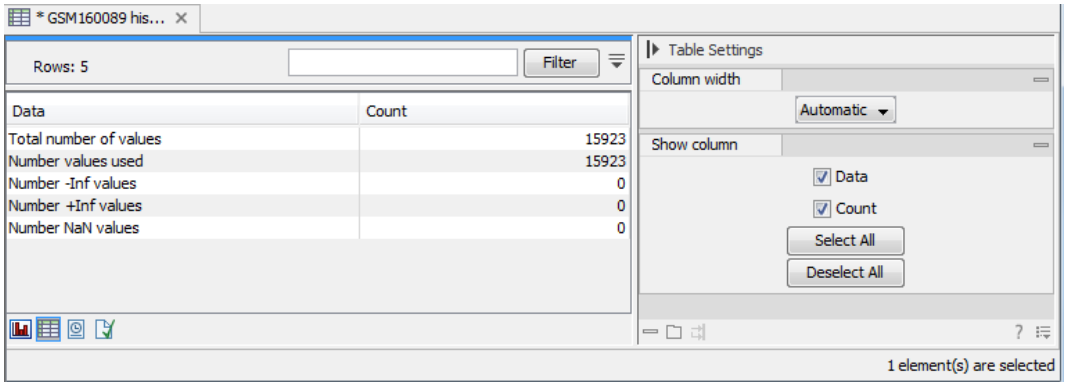


Figure 31.59: *Table view of a histogram.*

- **Number +Inf values**
- **Number -Inf values**
- **Number NaN values**
- **Number values used**
- **Total number of values**

31.7.2 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:

Toolbox | **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 31.60.

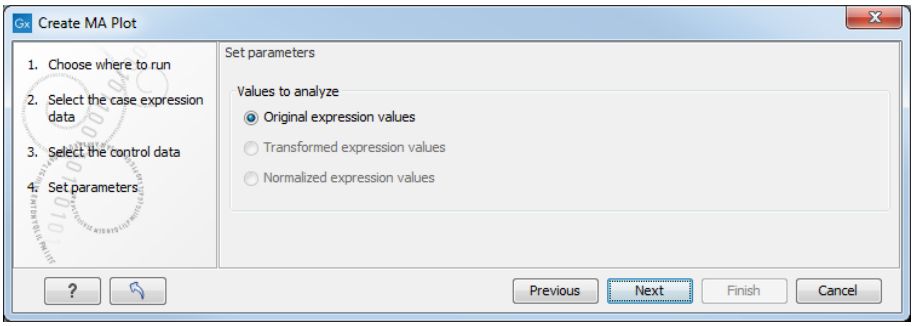


Figure 31.60: *Selecting which values the MA plot should be based on.*

In this dialog, you select the values to be used for creating the MA plot (see section 31.2.1).

Click **Finish** to start the tool.

Viewing MA plots

The resulting plot is shown in a figure 31.61.

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 31.61 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 31.2.2).

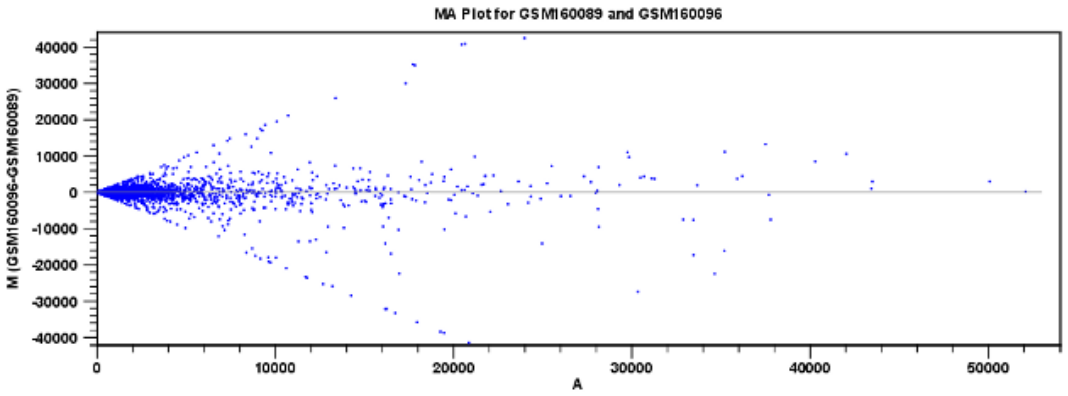


Figure 31.61: *MA plot based on original expression values.*

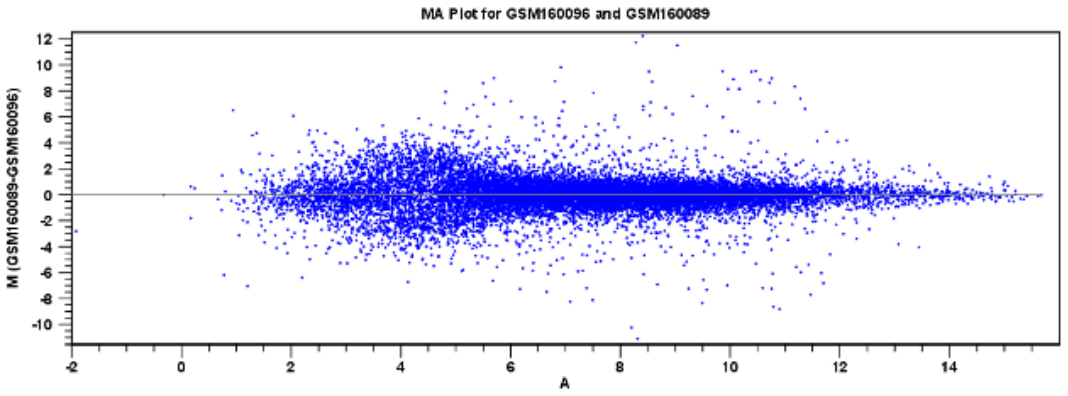


Figure 31.62: *MA plot based on transformed expression values.*

Figure 31.62 shows the same two samples where the MA plot has been created using log2 transformed values.

The much more symmetric and even spread indicates that the dependance 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).

31.7.3 Scatter plot

As described in section 31.1.4, an experiment can be viewed as a scatter plot. However, you can also create a "stand-alone" scatter plot of two samples:

Toolbox | **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 31.63.

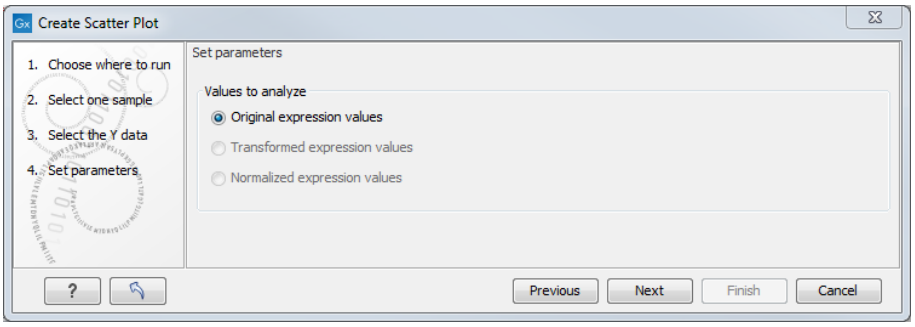


Figure 31.63: *Selecting which values the scatter plot should be based on.*

In this dialog, you select the values to be used for creating the scatter plot (see section 31.2.1).

Click **Finish** to start the tool.

For more information about the scatter plot view and how to interpret it, please see section 31.1.4.

Chapter 32

De Novo sequencing

Contents

32.1 The CLC de novo assembly algorithm 912
32.1.1 Resolve repeats using reads 914
32.1.2 Automatic paired distance estimation 916
32.1.3 Optimization of the graph using paired reads 917
32.1.4 AGP export 918
32.1.5 Bubble resolution 919
32.1.6 Converting the graph to contig sequences 920
32.1.7 Summary 920
32.2 De Novo Assembly 921
32.2.1 Best practices 921
32.2.2 Randomness in the results 923
32.2.3 De novo assembly parameters 923
32.2.4 De novo assembly report 925
32.2.5 De novo assembly output 926
32.3 Map Reads to Contigs 928

32.1 The CLC de novo assembly algorithm

Our 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 32.1.

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 32.2 shows an example where one node has two forward neighbors:

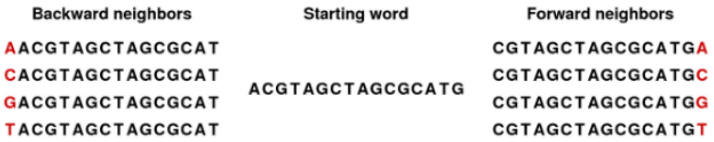


Figure 32.1: *The word in the middle is 16 bases long, and it shares the 15 first bases with the backward neighboring word and the last 15 bases with the forward neighboring word.*



Figure 32.2: *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 32.3.

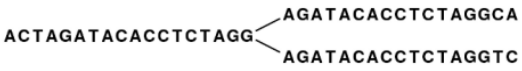


Figure 32.3: *The five nodes are compacted into three. Note that the first node is now 18 bases and the second nodes are each 17 bases.*

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 32.1.5) as shown in figure 32.4.



Figure 32.4: *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 32.5.



Figure 32.5: *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 32.2.3

32.1.1 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 subsections.

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 though 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 exits 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.

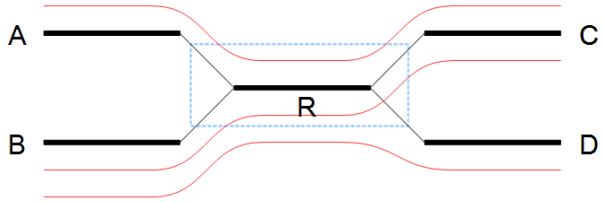


Figure 32.6: *A set of nodes.*

In the example in figure 32.6 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 32.7.

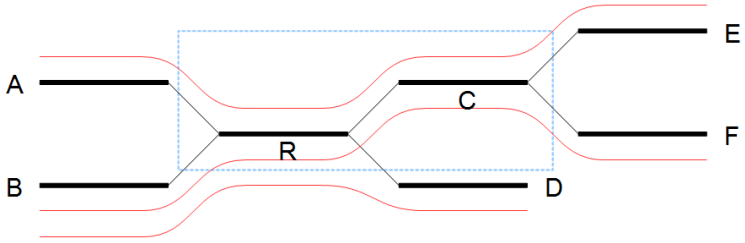


Figure 32.7: *Expanding the window to include more nodes.*

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 32.8.

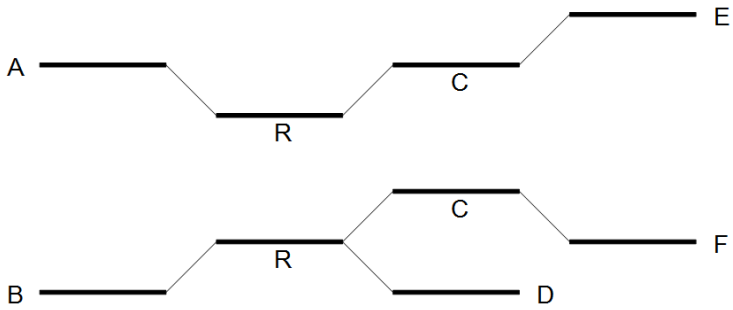


Figure 32.8: *Resolving the repeat.*

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.
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 32.9 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 32.10.

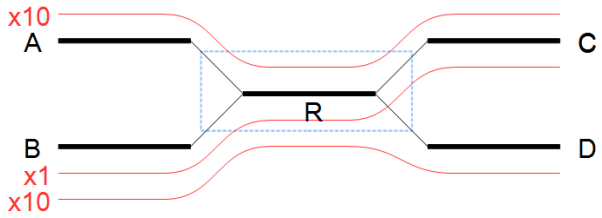


Figure 32.9: *A repeat with conflicts.*

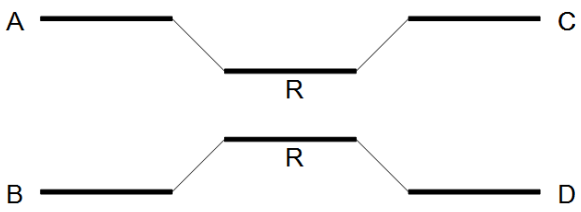


Figure 32.10: *Resolving a repeat with conflicts.*

32.1.2 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 32.11). 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 $H_{avg} = \frac{1}{|H|}\Sigma_d H(d)$ 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.

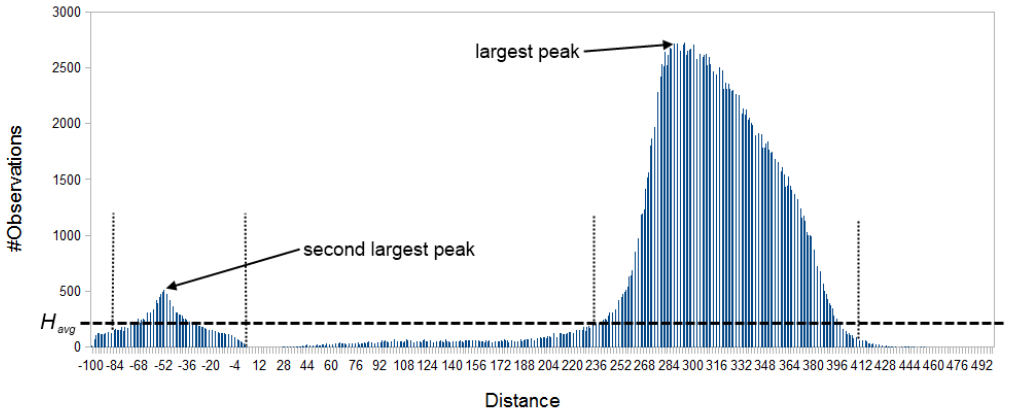


Figure 32.11: *Histogram of paired distances where H_{avg} is indicated by the horizontal dashed line. There is two peaks, one is at a negative distance while the other larger peak is at a positive distance. The extended interval $[k, l]$ for each peak is indicated by the vertical dotted lines.*

32.1.3 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 32.12).

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

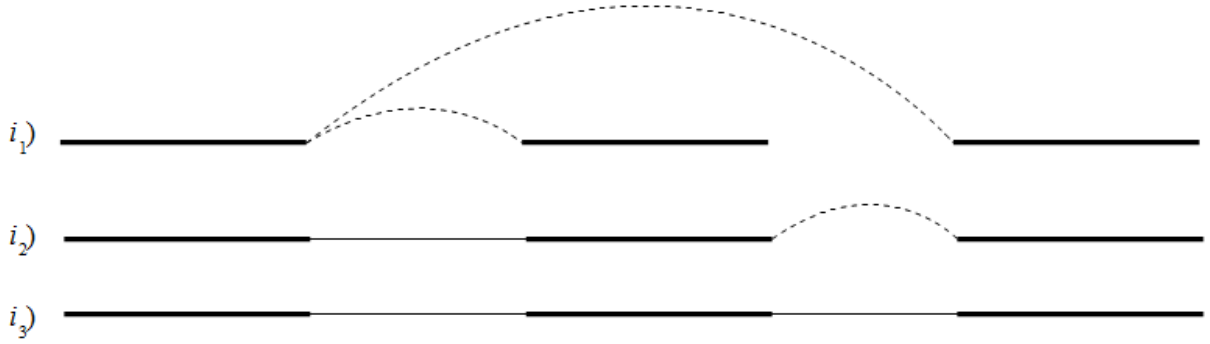


Figure 32.12: *Performing iterative scaffolding of the shortest gaps allows long pairs to be optimally used.* i_1 *shows three contigs with dashed arches indicating potential scaffolding.* i_2 *is after first iteration when the shortest gap has been closed and long potential scaffolding has been updated.* i_3 *is the final results with three contigs in one scaffold.*

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.

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 32.2.4). This information can also be exported to the AGP format (see section 32.1.4).

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.

32.1.4 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 32.13).

If the exporter encounters such a region, it will give a warning but not stop. If the exporter is executed on an assembly from GWB versions older than 6.5, it will often stop with an error saying that it encountered more than 10 N's which wasn't marked as a scaffold region. In this case the user would have to rerun the assembly with GWB version 6.5 or newer of the de novo assembler if they wish to be able to export to AGP.

Currently we output two types of annotations in AGP format:

- **Contig** a non-redundant sequence not containing any scaffolded regions.
- **Scaffold** the estimated gap region between two contigs.

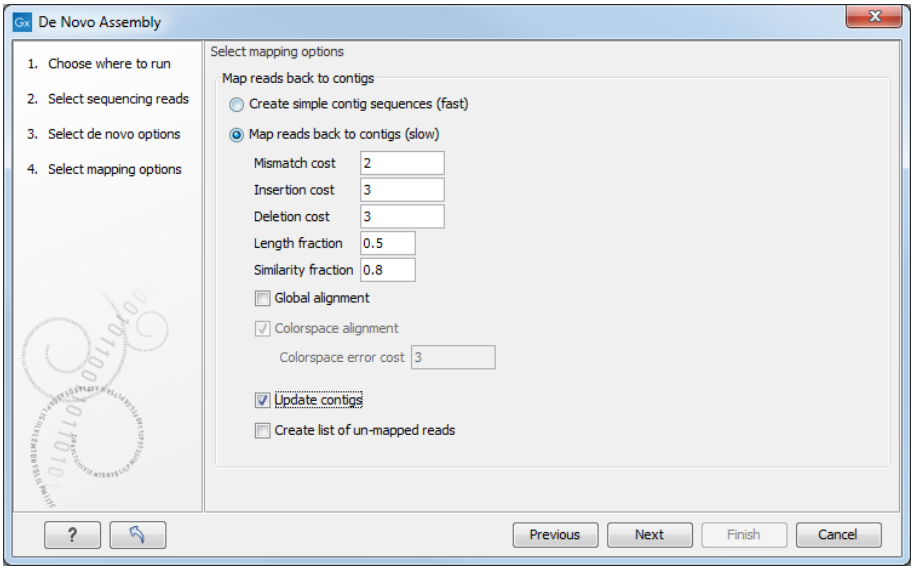


Figure 32.13: *Select "update contigs" by ticking the box if you want to resolve scaffolds based on a read mapping.*

32.1.5 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 32.14.



Figure 32.14: *A bubble caused by a heteroygous 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 32.15 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.



Figure 32.15: *Reads with systematic errors.*

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 32.16).

In this case, the bubble will be very large because there are no complete words in the regions between the homopolymer sites, and

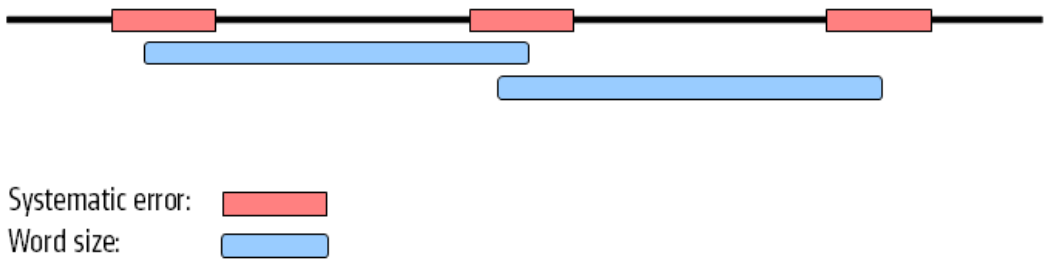


Figure 32.16: *Several sites of errors that are close together compared to the word size.*

the graph will look like figure 32.17.

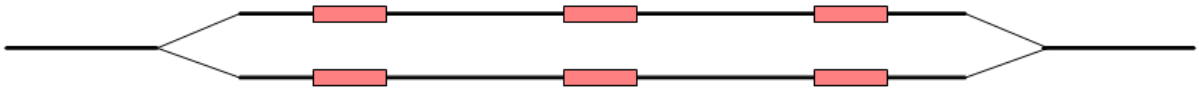


Figure 32.17: *The bubble in the graph gets very large.*

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 32.17, a bubble size spanning the three error sites will mean that the bubble will be resolved (see figure 32.18).

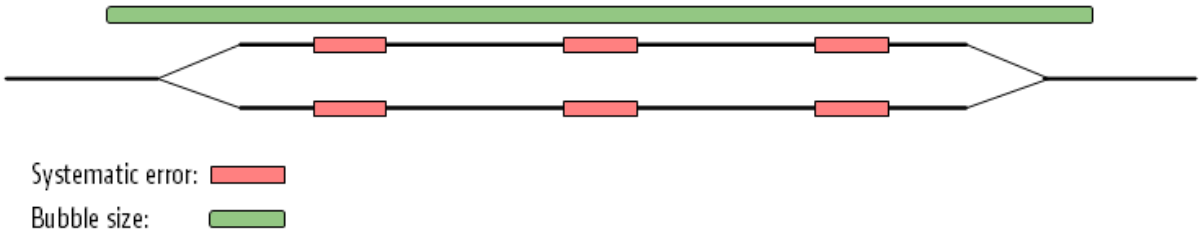


Figure 32.18: *The bubble size needs to be set high enough to encompass the three sites.*

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.

32.1.6 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 32.1.3.

32.1.7 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.

32.2 De Novo Assembly

The de novo assembly algorithm of *CLC Genomics Workbench* offers comprehensive support for a variety of data formats, including both short and long reads, and mixing of paired reads (both insert size and orientation).

The de novo assembly process has 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 32.1.
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 32.2.3).

Note: The De Novo Assembly tool was optimized for genomes up to the size and complexity of the human genome. Please contact ts-bioinformatics@qiagen.com if you would like to use the De Novo assembler with genomes that are larger and more complex than the human genome. We take into account such requests to assist future features prioritization.

32.2.1 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 6.3.7).

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 32.2.4) 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.

32.2.2 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.

32.2.3 De novo assembly parameters

To start the assembly:

Toolbox | **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 32.19.

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 32.1. 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 32.1.5. The automatic bubble size is set to 50, unless one of the following conditions apply:

- some of the reads are from either 454, 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 32.1.1 and section 32.1.3). 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.

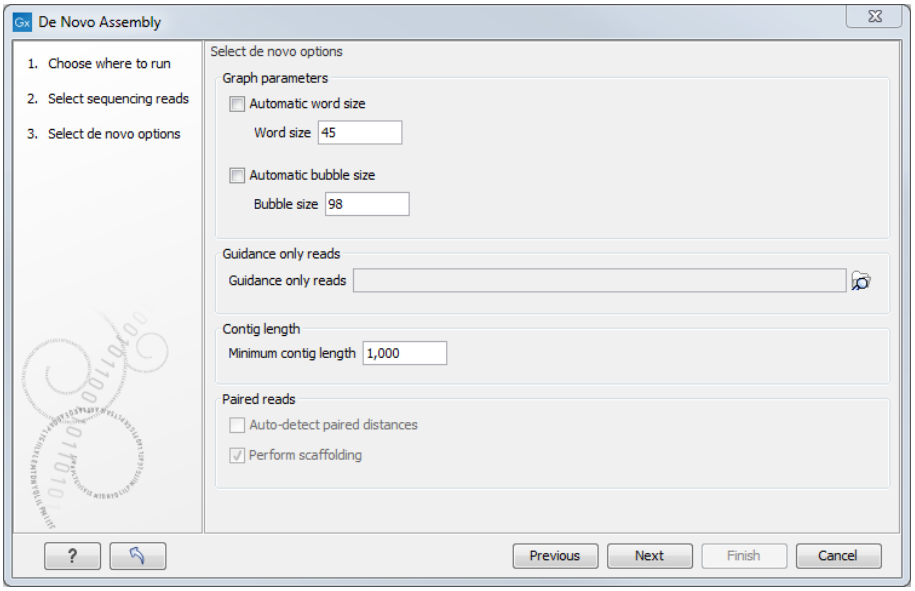


Figure 32.19: *Setting parameters for the assembly.*

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 32.1.3. 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 6.3.7).

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 26.2). 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 32.20

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 27.1.3. The placement of reads that map in more than one position equally well are placed randomly (see section 27.1.5) 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.

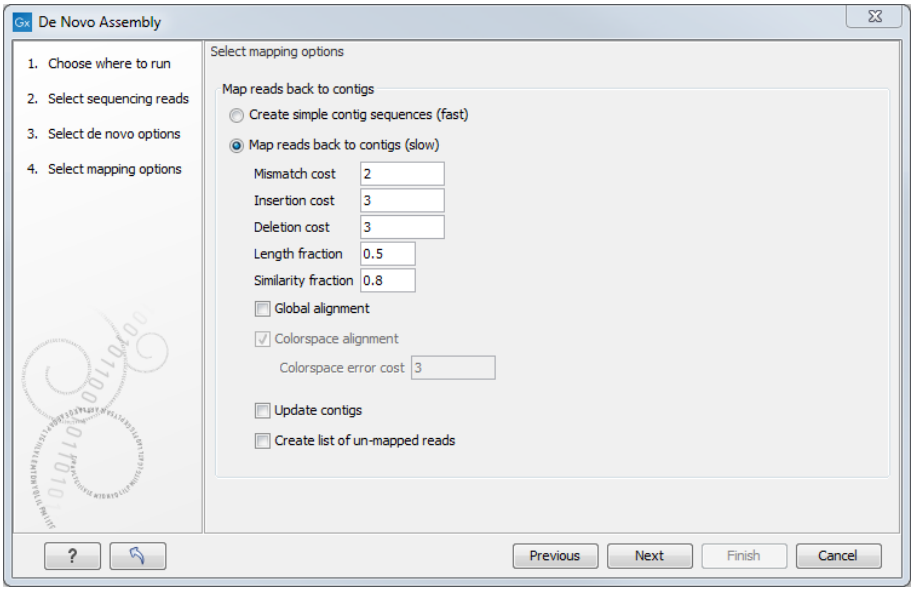


Figure 32.20: *Parameters for mapping reads back to the contigs.*

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 32.19 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

32.2.4 De novo assembly report

A denovo assembly reports looks like the one shown in figure 32.21.

The report contains the following information when both scaffolding and read mapping is performed:

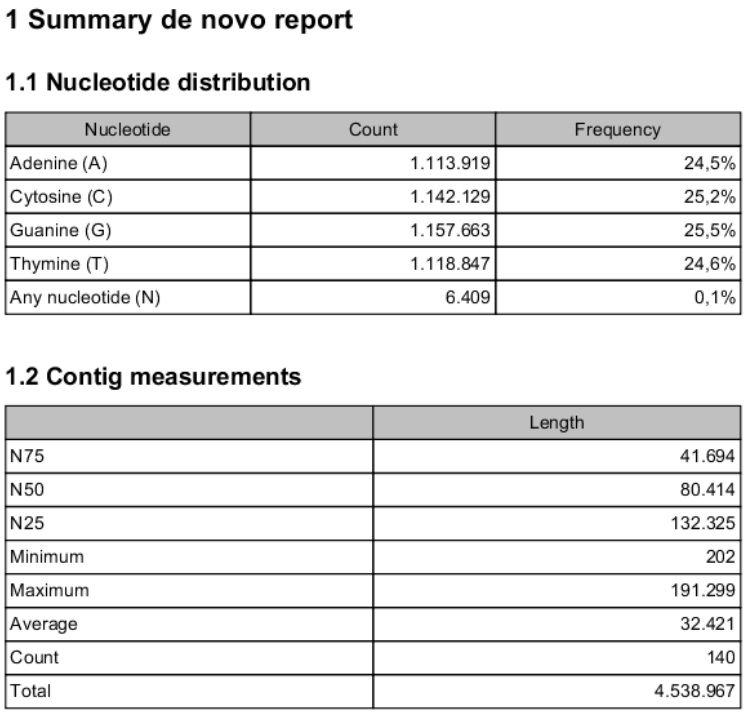
Nucleotide distribution This includes Ns when scaffolding has been performed.

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.

- **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.



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 32.21: *Creating a de novo assembly report.*

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 26.2.

32.2.5 De novo assembly output

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 32.22.

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.
- **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.

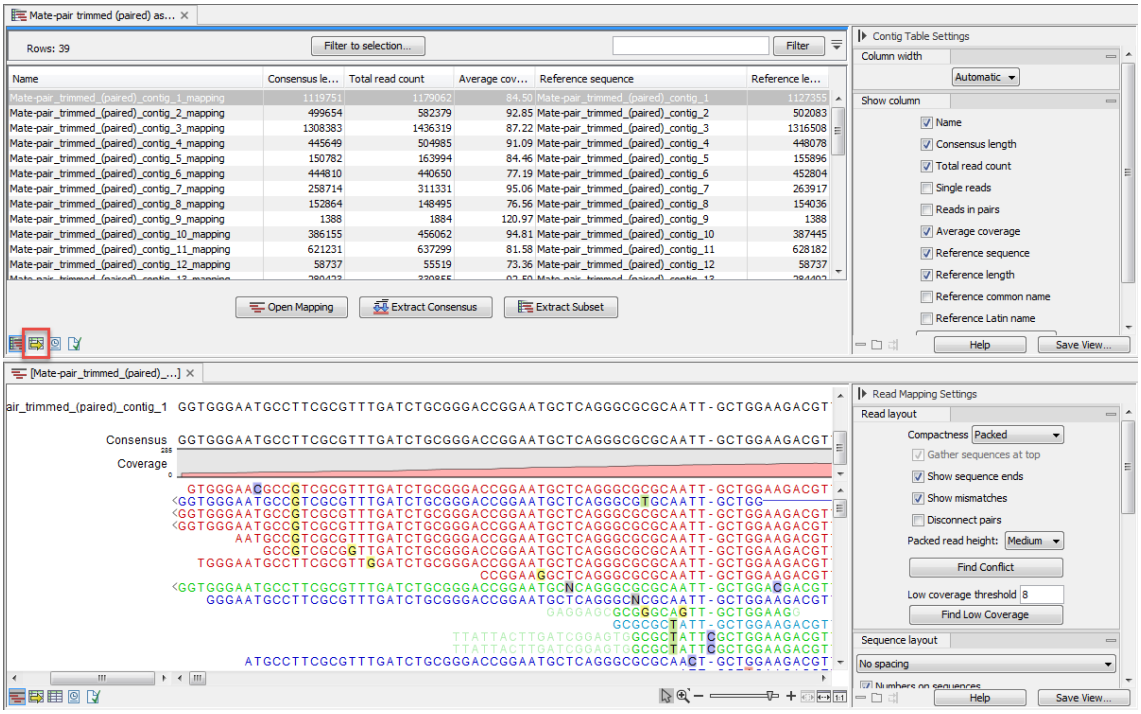


Figure 32.22: *A de novo assembly read mapping.*

- **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 27.6).
- **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 32.22). The annotations available in the table are the following (see figure 32.23):

- **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 by 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.

Simple contigs The output is a sequence list of the contigs generated (figure 32.24), that can also be seen as a table and an annotation table as described for the stand-alone read mapping described above.

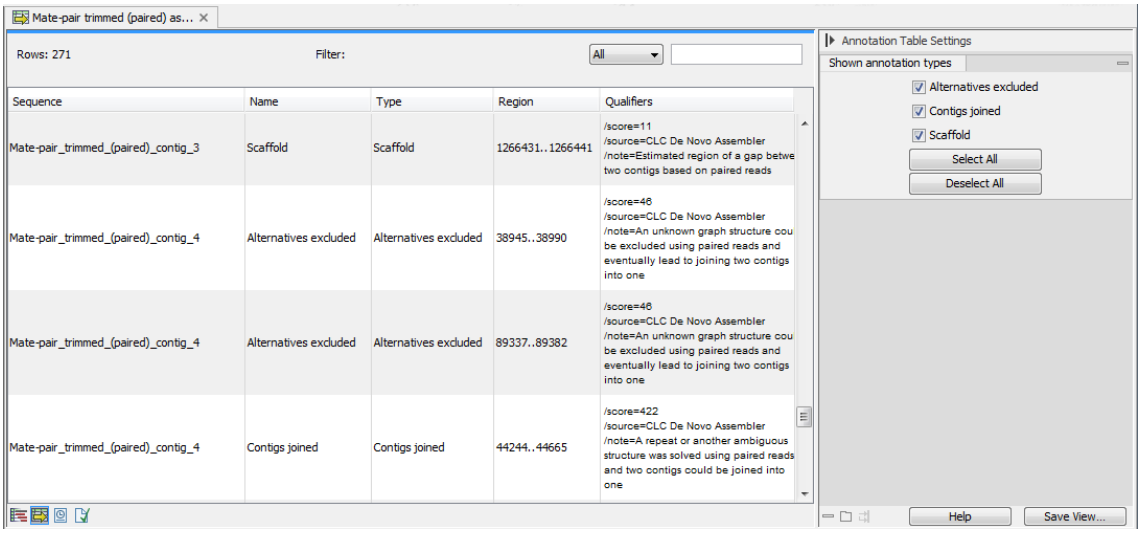


Figure 32.23: *A de novo assembly read mapping seen as annotation table.*

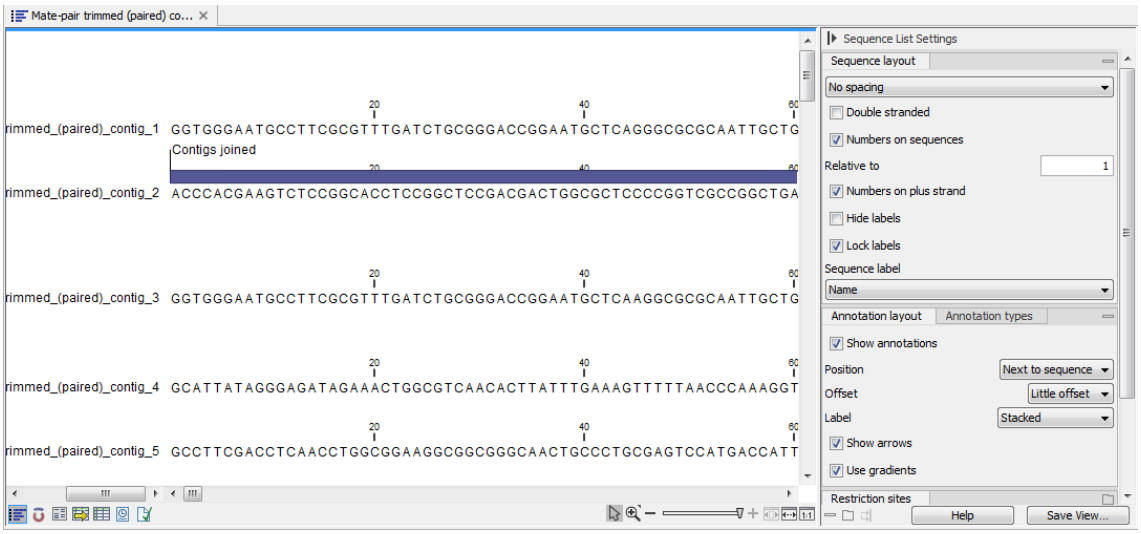


Figure 32.24: *A de novo assembly read mapping seen as annotation table.*

32.3 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 27.1).

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:

Toolbox | **De Novo Sequencing (** **)** | **Map Reads to Contigs (** **)**

This opens up the dialog in figure 32.25 where you select the reads you want to map to the contigs. Click **Next**.

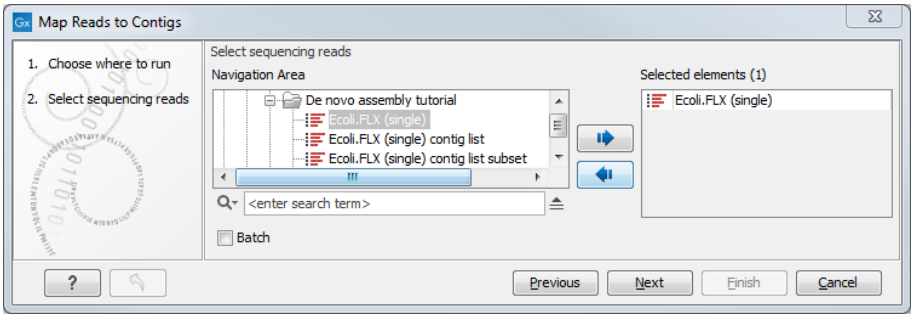


Figure 32.25: *Select reads. The contigs will be selected in the next step.*

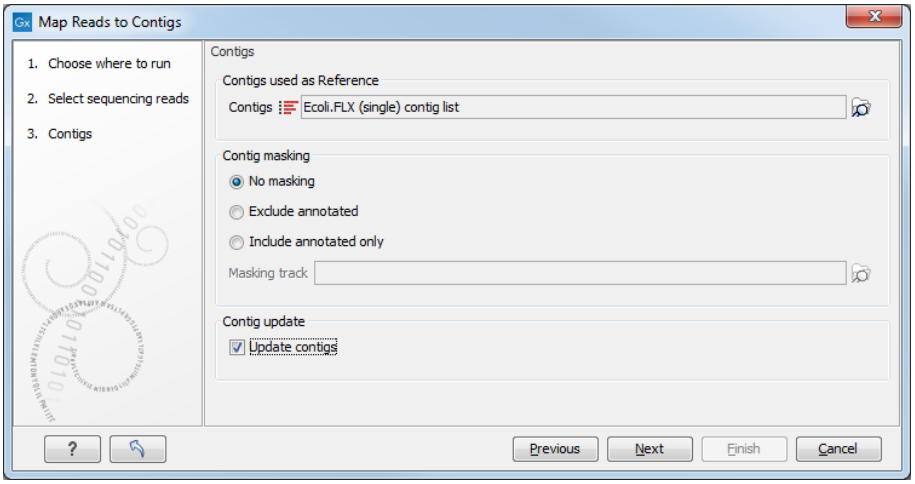


Figure 32.26: *Select contigs and specify whether to use masking and the "Update contigs" function.*

Select the contigs to map the reads against (figure 32.26).

Under "Contig masking", specify whether to include or exclude specific regions (for a description of this see section 27.1.2).

The contigs can be updated by selecting "Update contigs" at the bottom of the wizard. 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 32.27).
- 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.

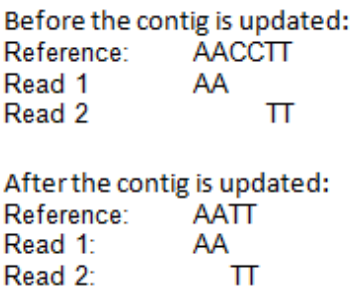


Figure 32.27: *When selecting "Update Contig" in the wizard, contigs will be updated according to the reads. This means that regions of a contig where no reads map will be removed.*

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 27.1.3).

The output from the Map Reads to Contigs tool can be a track or stand-alone read mappings as selected in the last dialog.

When stand-alone read mappings have been selected as output, it is possible to edit and delete contig sequences.

Figure 32.28 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.

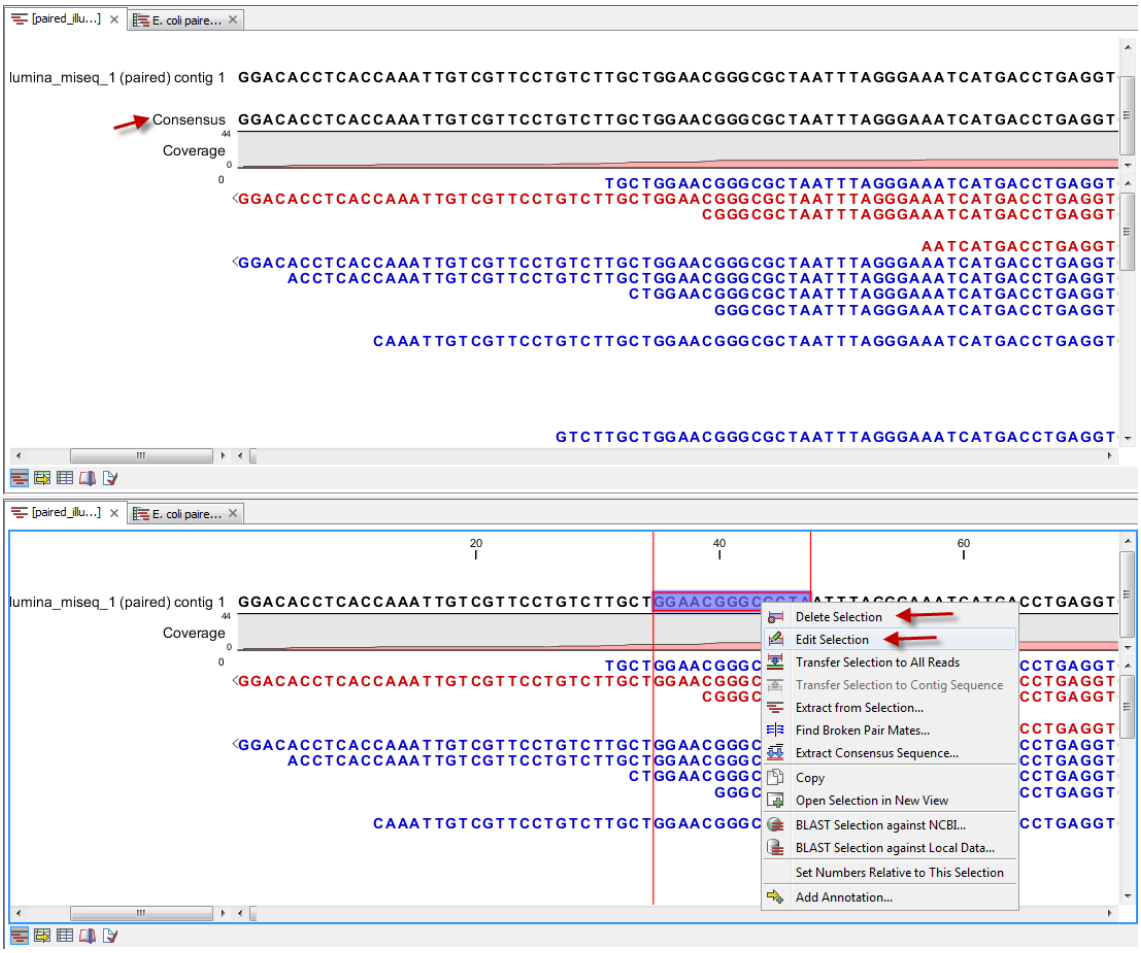


Figure 32.28: *Two different read mappings performed with Map Reads to Reference (top) and Map Reads to Contigs (bottom). The differences are highlighted with red arrows.*

Chapter 33

Epigenomics analysis

Contents

33.1 Histone Chip-Seq . . . 931
33.2 ChIP-Seq Analysis . . . 933
33.2.1 Quality Control of ChIP-Seq data . . . 934
33.2.2 Learning peak shapes . . . 934
33.2.3 Applying peak shape filters to call peaks . . . 935
33.2.4 Running the Transcription Factor ChIP-Seq tool . . . 935
33.2.5 Peak track . . . 937
33.3 Annotate with nearby gene information . . . 938
33.4 Bisulfite Sequencing . . . 939
33.4.1 Detecting DNA methylation . . . 939
33.4.2 Map Bisulfite Reads to Reference . . . 942
33.4.3 Call Methylation Levels . . . 951
33.4.4 Create RRBS-fragment Track . . . 957
33.5 Advanced Peak Shape Tools . . . 957
33.5.1 Learn Peak Shape Filter . . . 958
33.5.2 Apply Peak Shape Filter . . . 959
33.5.3 Score Regions . . . 961

33.1 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:

Toolbox | **Epigenomics Analysis (**) | **Histone ChIP-Seq (**)

In the first wizard window, select the mapped ChIP-Seq reads as input data (figure 33.1). 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 9.3).

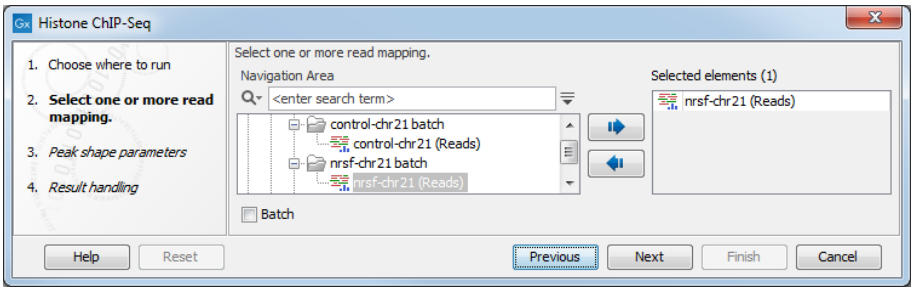


Figure 33.1: *Selecting input tracks for the Histone ChIP-Seq tool.*

In the second step (figure 33.2), 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.

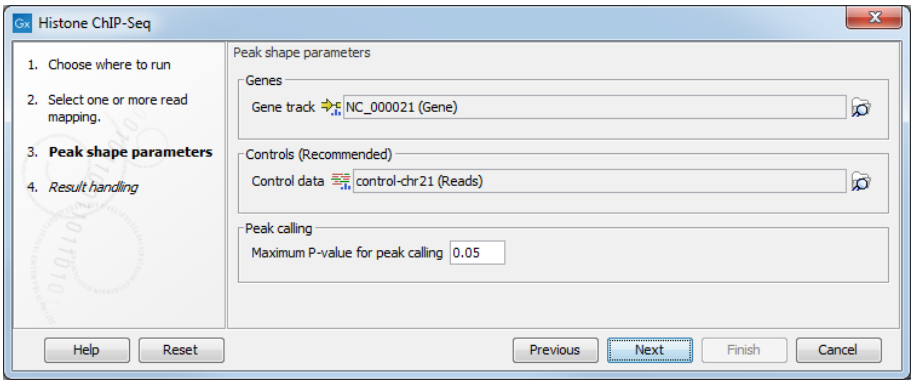


Figure 33.2: *Setting peak shape parameters.*

The output options are shown in figure 33.3.

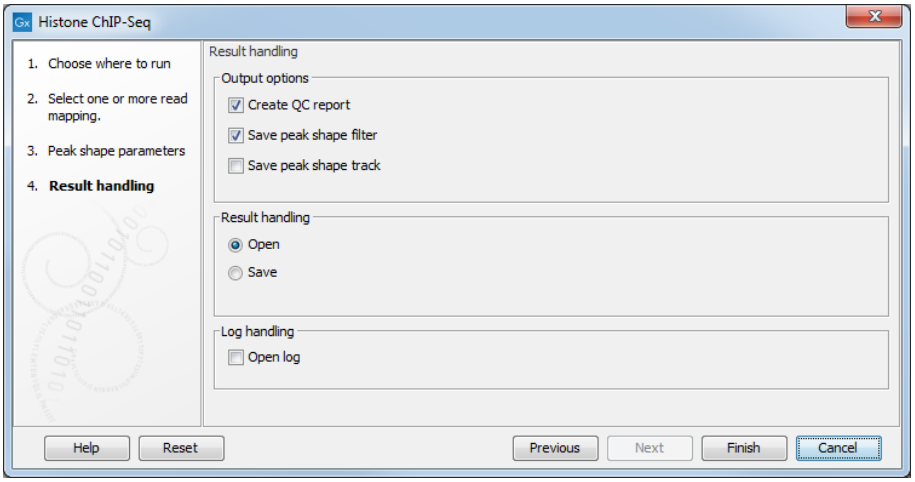


Figure 33.3: *Setting up result handling.*

- **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 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, please refer to section 33.2.2, or to 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 33.4)

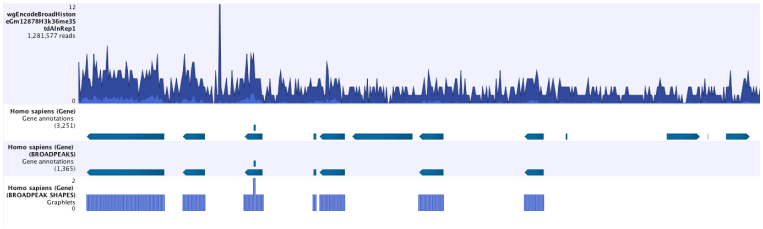


Figure 33.4: *Example of Histone ChIP-Seq output.*

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.

33.2 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 immunopreciptation 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 33.5).

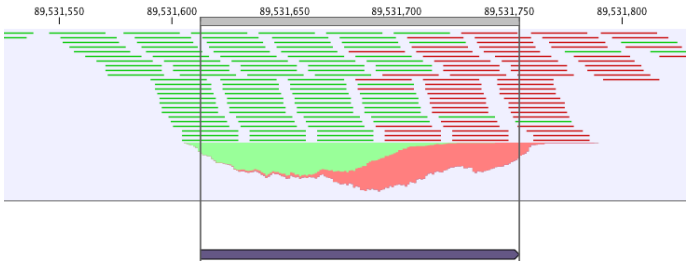


Figure 33.5: *Distribution of forward (green) and reverse (red) reads around a binding site of the transcription factor NRSF.*

The tool makes use of this characteristic shape to identify enriched regions (peaks) in ChIP-Seq data.

33.2.1 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 33.6), 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 33.6) for successful ChIP-Seq experiments.

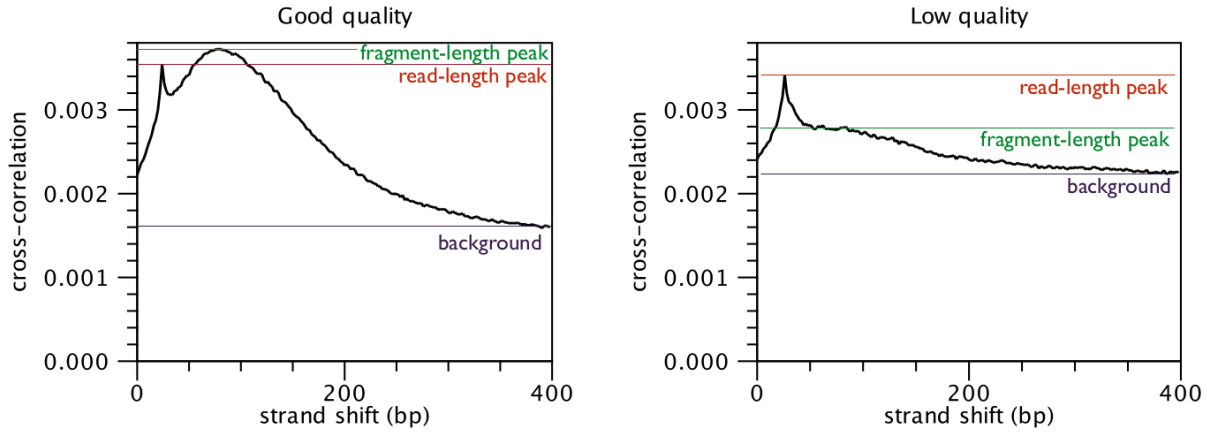


Figure 33.6: *Difference in cross-correlation profiles in ChIP experiments of good and low quality.*

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].

33.2.2 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 33.7.

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 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:

$$h = \left(\frac{R_p + R_n}{2} \right)^{-1} \left(\mathbb{E}[X_p] - \mathbb{E}[X_n] \right), \tag{33.1}$$

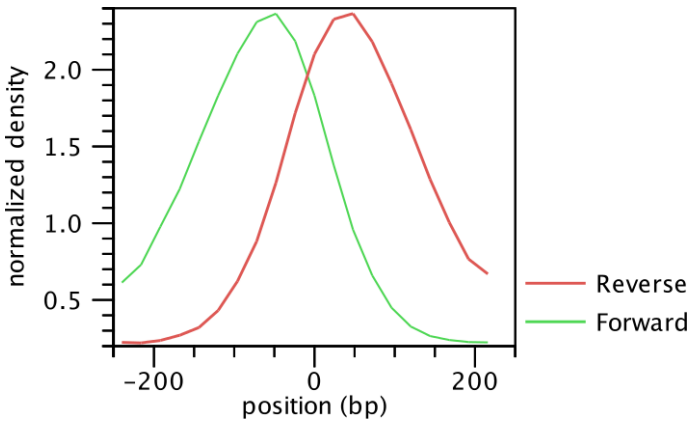


Figure 33.7: *Average peak shape of the transcription factor NRSF.*

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 33.8.

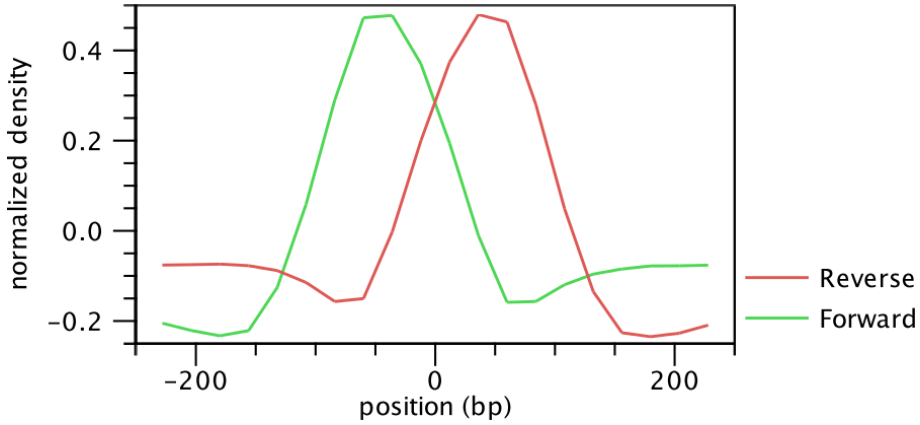


Figure 33.8: *Peak shape filter for the transcription factor NRSF.*

33.2.3 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 p-value $= \Phi(-$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 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.

33.2.4 Running the Transcription Factor ChIP-Seq tool

To run the Transcription Factor ChIP-Seq tool:

Toolbox | **Epigenomics Analysis** ( ) | **Transcription Factor ChIP-Seq** ( )

This will open up the wizard shown in figure 33.9 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.

Click **Next** to go to the next wizard step (shown in figure 33.10).

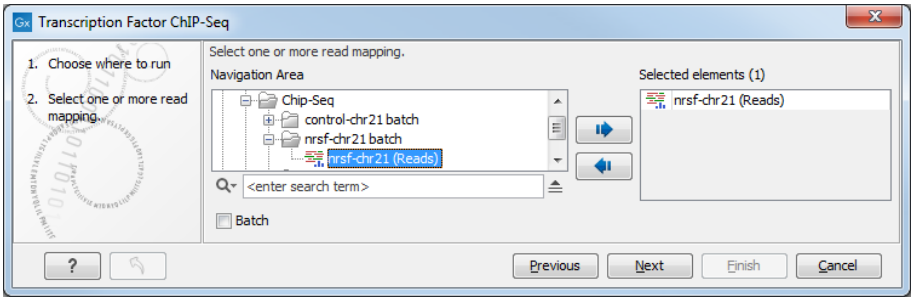


Figure 33.9: *Select the input data for the Transcription Factor ChIP-Seq tool.*

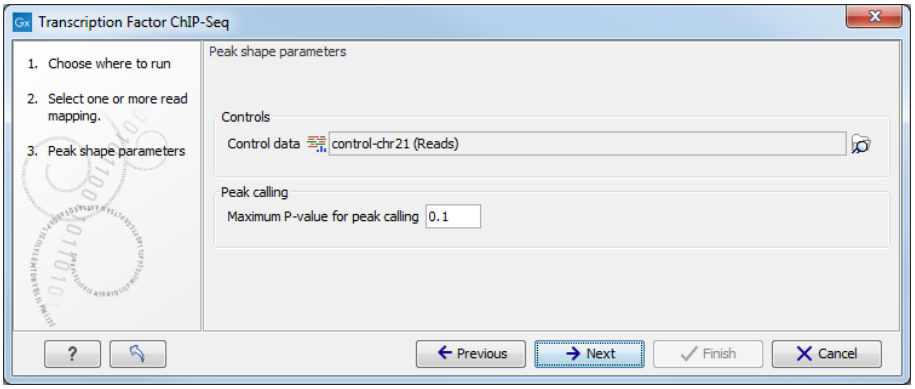


Figure 33.10: *Options for the Transcription Factor ChIP-Seq tool.*

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 33.11.

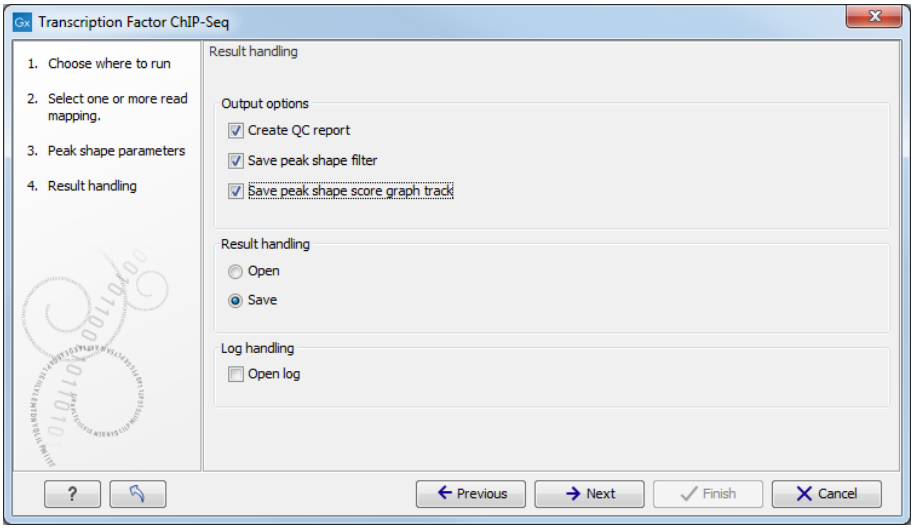


Figure 33.11: *Output options for the Transcription Factor ChIP-Seq tool.*

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 33.2.1.

- **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 33.2.3.

- **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 33.2.3.

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.

33.2.5 Peak track

The main result of the Transcription Factor ChIP-Seq algorithm is an annotation track containing the peaks. For each peak the following quantities are reported in the table, which can be opened by clicking on the table icon () in the lower left corner of the peak annotation track. For more details on some of the values included in the table, see section 33.2.3.

- **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.

The peak annotation track is most informative when combined with the read mapping in a Track List (you can see how to create a track list here: http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Track_lists.html).

A Track List containing the mapped reads, the Peak track, and the Peak shape score track is shown in figure 33.12.

Note that if you make a split view of the table and the peak annotation track (by holding down the Ctrl key (Cmd on Mac) while clicking on the table icon () in the lower left corner of the peak annotation track), you will be able to browse through the peaks by clicking in the table, as the peak annotation track and the table are connected. As a result the view will jump to the position of the peak selected in the table.

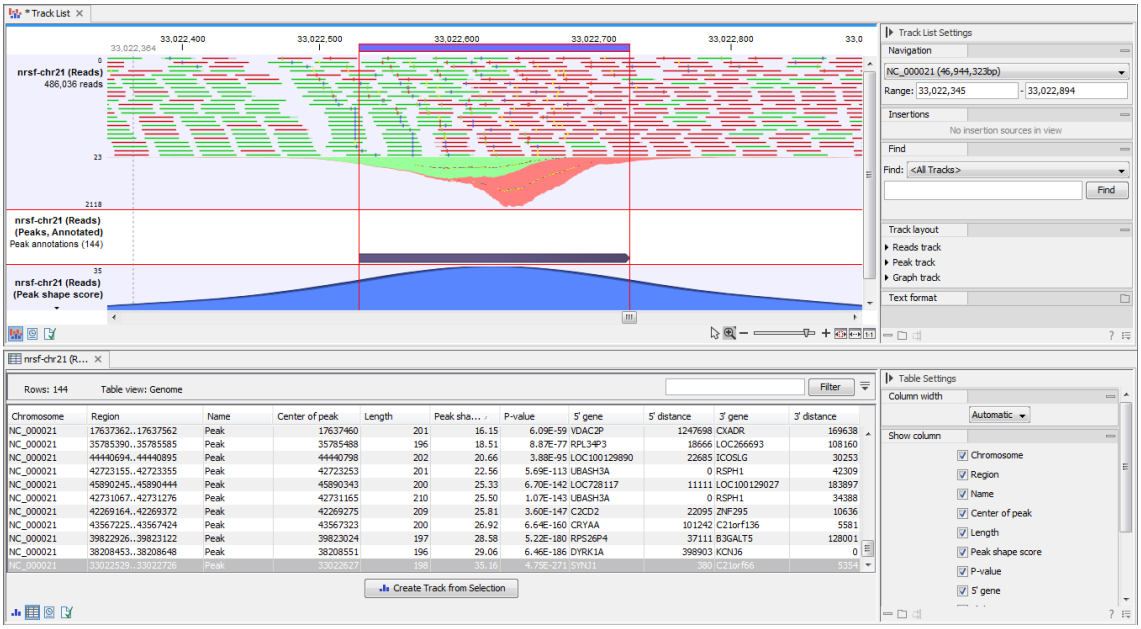


Figure 33.12: *Inspection of the result of the Transcription Factor ChIP-Seq tool.*

33.3 Annotate with nearby gene information

This tool will create a copy of the annotation track ( ) used as input and add information about nearby genes.

Toolbox | **Epigenomics Analysis ( )** | **Annotate with Nearby Gene Information ( )**

First, select the track you wish to annotate and click **Next**. The tool was designed for ChIP-Seq analysis, but you can choose any kind of annotation track as input. Next, select a gene track with a compatible genome (figure 33.13).

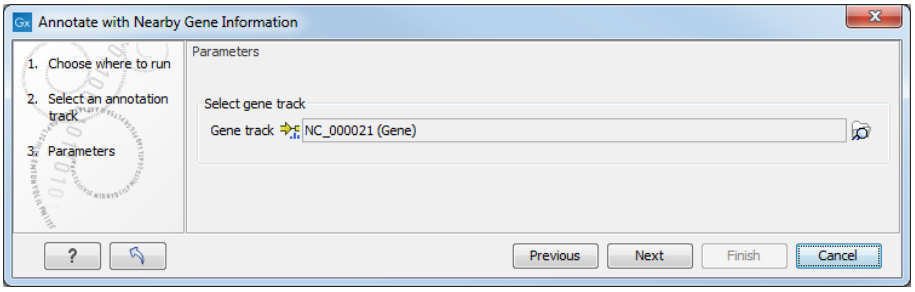


Figure 33.13: *Select gene track.*

The result of this tool is a new annotation track with all the annotations from the input track and with additional information about nearby genes and four columns will be added to the table view:

- **5' gene** The name of the nearest upstream gene.
- **5' distance** The distance from the nearest upstream gene or 0 if the feature overlaps the nearest gene. The distance value is determined by the shortest distance between an end of the gene annotation and an end of the peak annotation, regardless of the annotation orientation (see figure 33.14).
- **3' gene** The name of the nearest downstream gene.

- **3' distance** The distance from the nearest downstream gene or 0 if the feature overlaps the nearest downstream gene. The distance value is determined by the shortest distance between an end of the gene annotation and an end of the peak annotation, regardless of the annotation orientation.

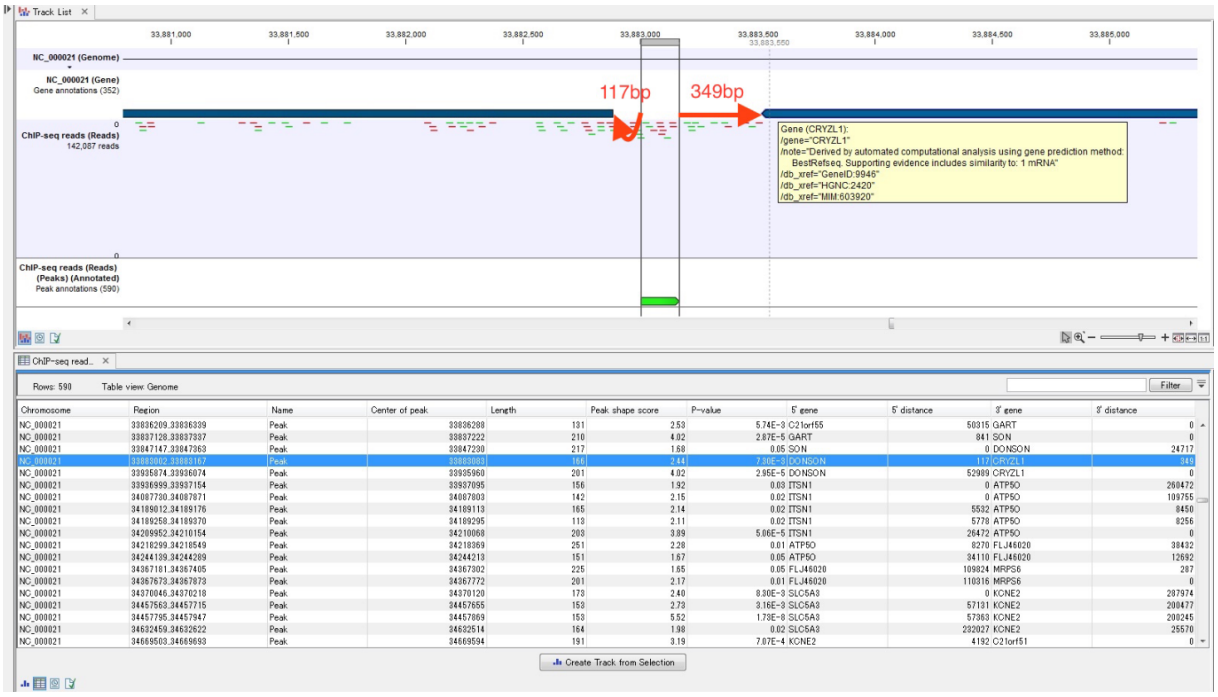


Figure 33.14: *The figure is a split view between a table view and a track list showing the gene annotation track, the ChIP-Seq reads, and the annotated ChIP peaks.The red arrows and numbers illustrate the 5' distance and 3' distance.*

33.4 Bisulfite Sequencing

33.4.1 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 33.15). 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.

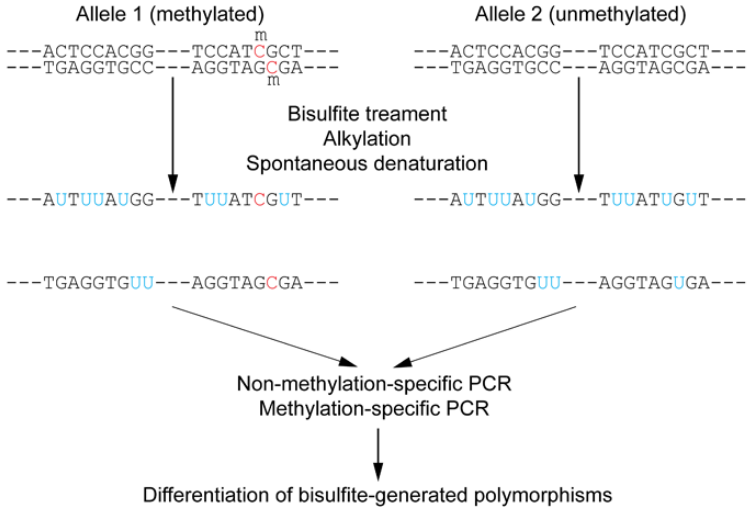


Figure 33.15: *Outline of bisulfite conversion of sample sequence of genomic DNA. Nucleotides in blue are unmethylated cytosines converted to uracils by bisulfite, while red nucleotides are 5-methylcytosines resistant to conversion. Source:* https://en.wikipedia.org/wiki/Bisulfite_sequencing

Figure 33.16 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.

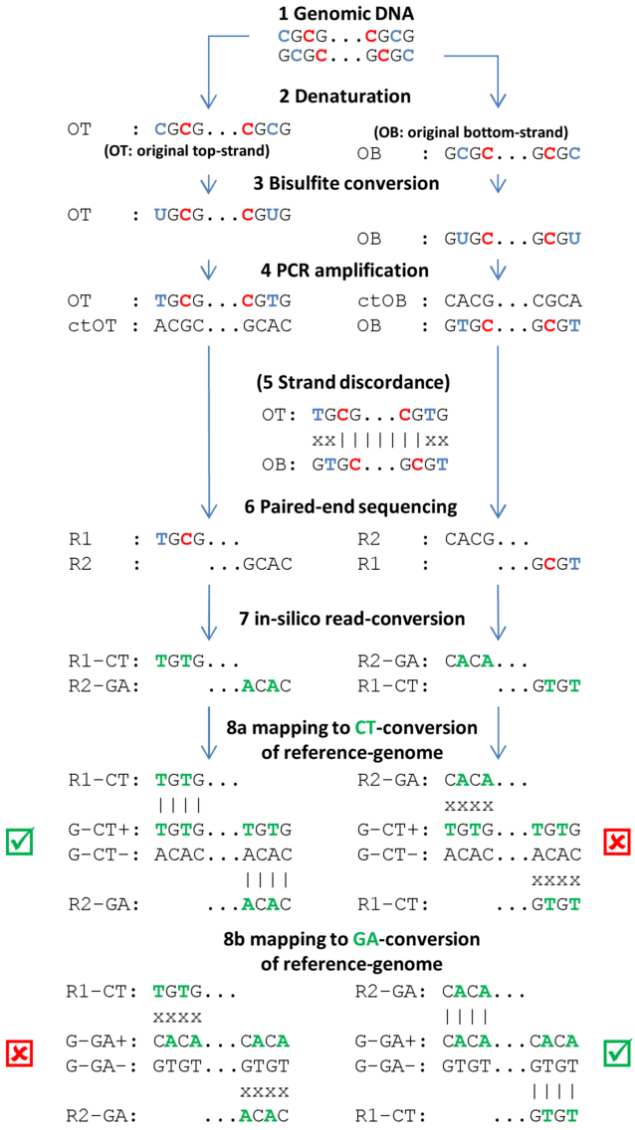


Figure 33.16: *Individual steps of BS-seq workflow include denaturation of fragmented sample DNA, bisulfite conversion, subsequent amplification, sequencing and mapping of resulting DNA-fragments. (See text for explanations). Methylated cytosines are drawn in red, unmethylated cytosines and respective uracils/thymidines in blue. DNA-nucleotides that are in-silico converted (during read mapping) are given in green.*

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 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.

33.4.2 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 33.4.3) 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:

Toolbox | **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 33.17). Please note that reads longer than 100,000 bases are not supported.

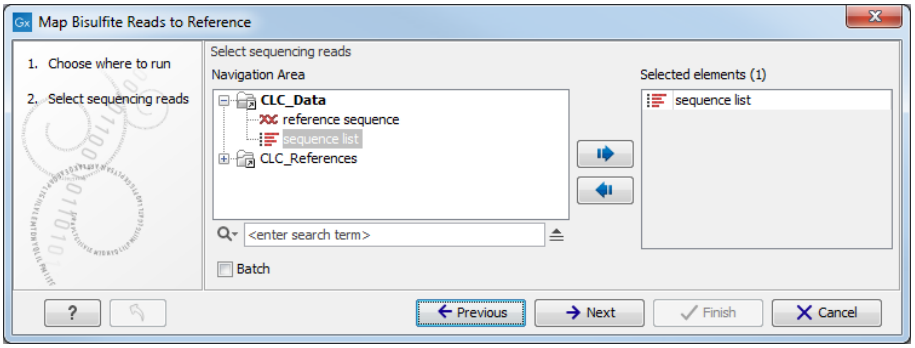


Figure 33.17: *Specifying the reads as input. You can also choose to work in batch.*

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:

- 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 33.18.

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.

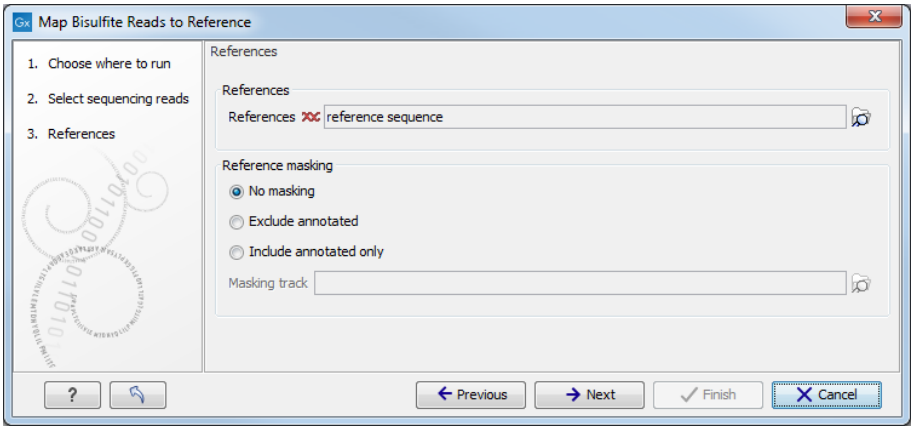


Figure 33.18: *Specifying the reference sequences and masking.*

- 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 33.18 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 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 33.19).

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.

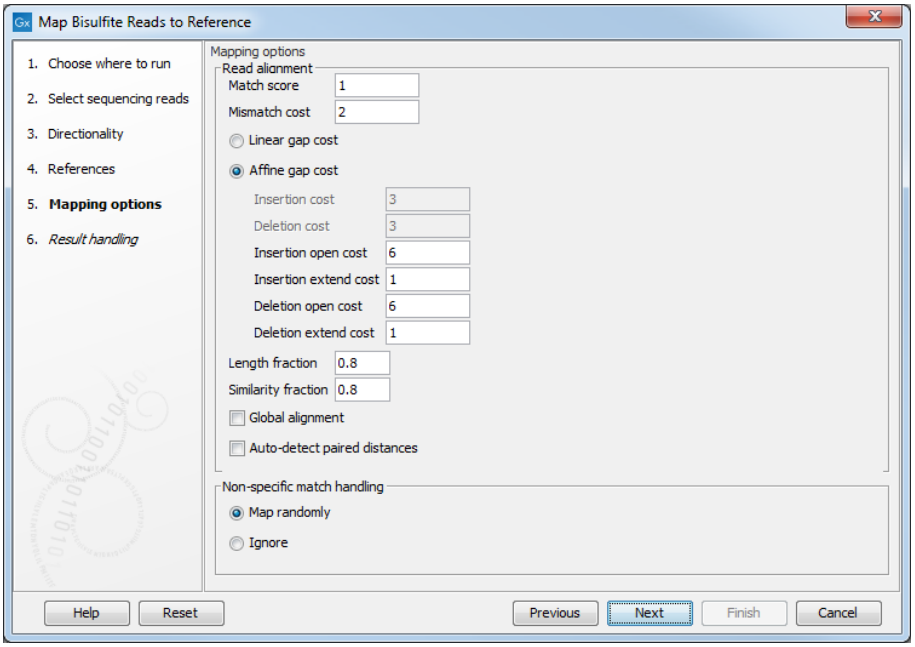


Figure 33.19: *Setting parameters for the bisulfite read mapping.*

- **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 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 $\ell \cdot$ `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 $\ell \cdot$ `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
`Insertion open cost` + $\ell \cdot$ `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.

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.

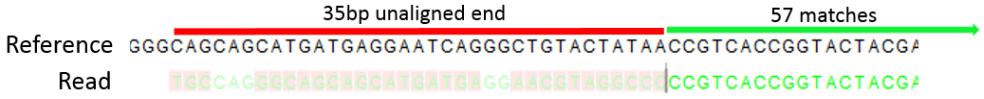


Figure 33.20: *An alignment of a read where a region of 35bp at the start of the read is unaligned while the remaining 57 nucleotides matches the reference.*

Figure 33.20 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 33.21 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 33.20 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 33.21 would become 61 and 58, respectively, and thus make it optimal.

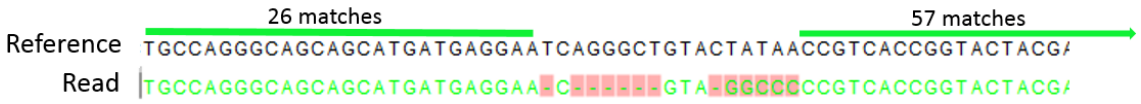


Figure 33.21: *An alignment of a read containing a region with several mismatches and deletions. By reducing the default cost of either mismatches or deletions the read mapper can make an alignment that spans the full length of the read.*

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 33.22).

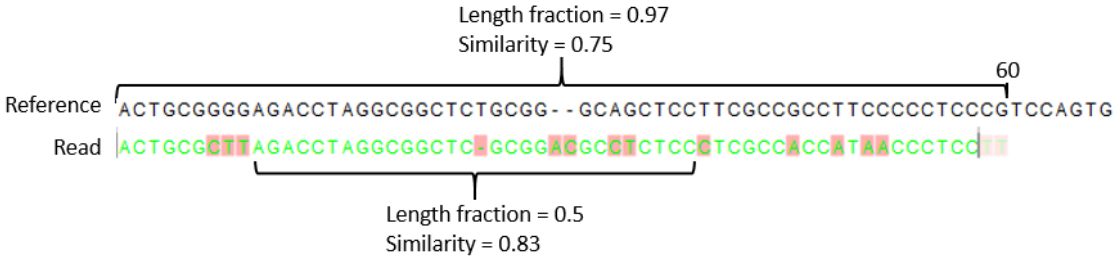


Figure 33.22: *A read containing 59 nucleotides where the total alignment length is 60. The part of the alignment that gave rise to the optimal score has length 58 which excludes 2 bases at the left end of the read. The length fraction of the matching region in this example is therefore 58/60 = 0.97. 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 of paired distances. Figure 33.23 shows an example of the distribution of intervals with and without automatic pair distance interval estimation.

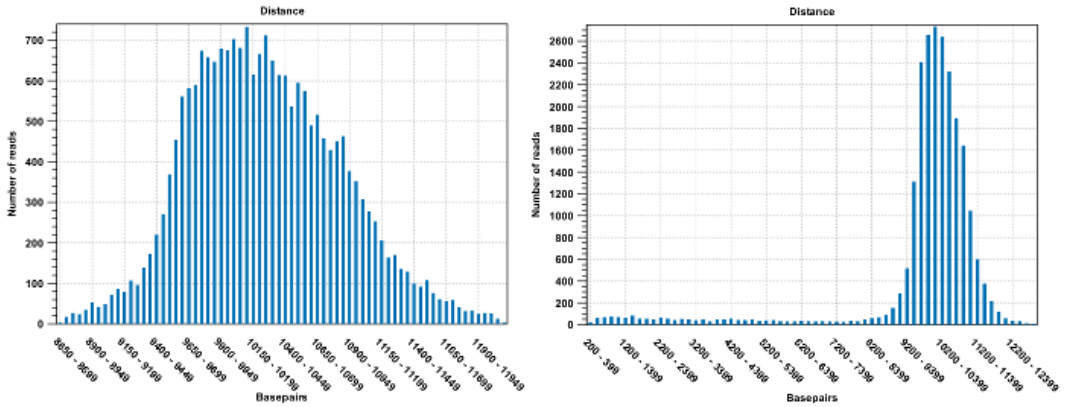


Figure 33.23: *To the left: mapping with a narrower distance interval estimated by the workbench. To the right: mapping with a large paired distance interval (note the large right tail of the distribution).*

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 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 27.1.5 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 27.1.4).

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 33.24.



Figure 33.24: *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 33.25.



Figure 33.25: *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 http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Summary_mapping_report.html.
- **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.

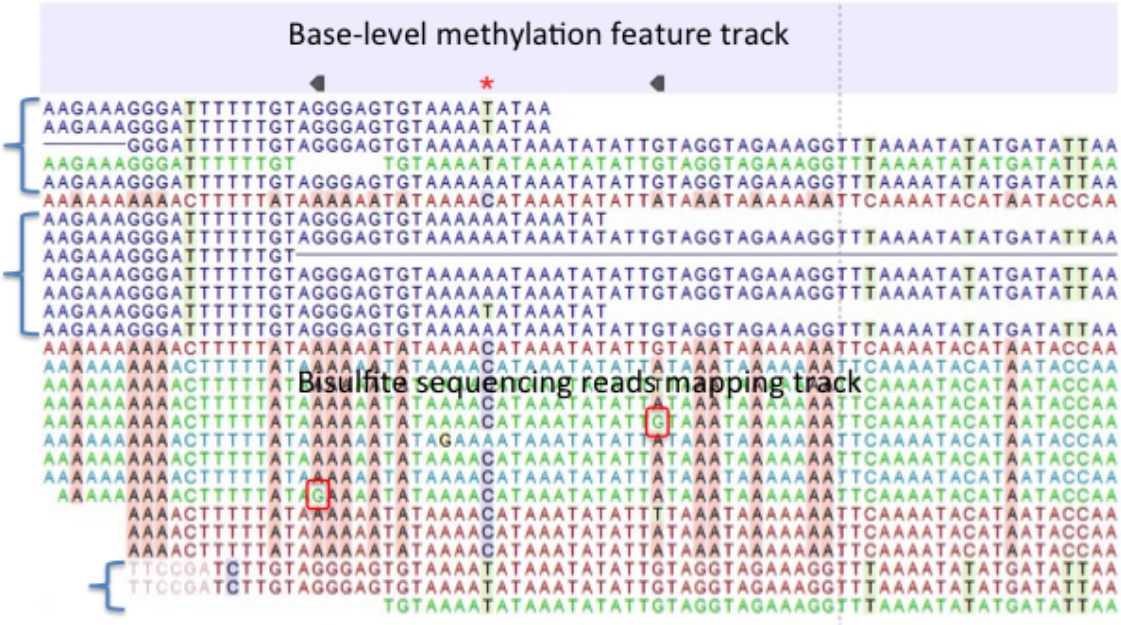


Figure 33.26: *A typical directional shotgun BS-seq mapping, together with the base-level methylation calling feature track on top.*

Figure 33.26 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.

33.4.3 Call Methylation Levels

The tool takes as input one or more read mappings created by the Map Bisulfite Reads to Reference tool (see section 33.4.2). 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 33.27:

- 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.

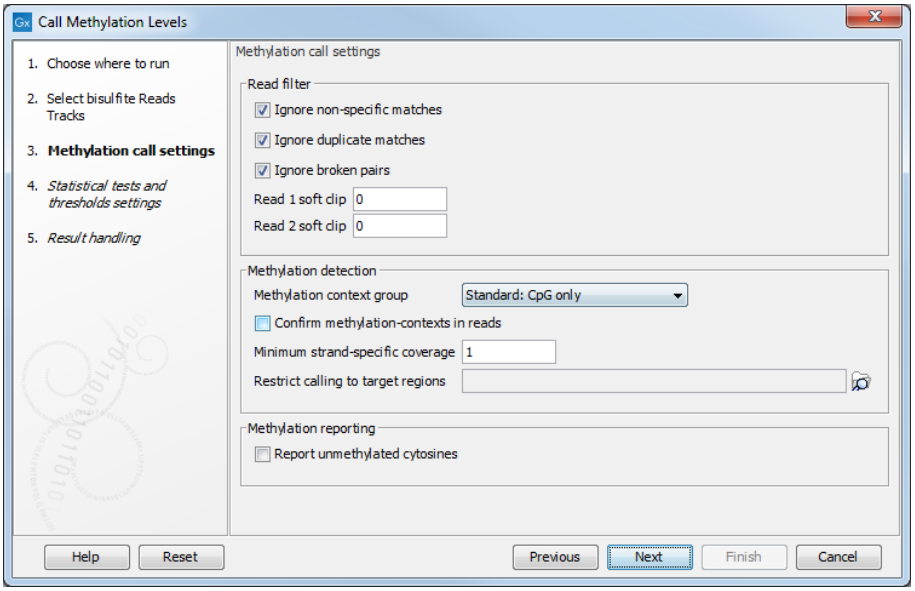


Figure 33.27: *Methylation call settings.*

  - **Ignore duplicate matches** Multiple reads with identical mapping coordinates will be counted once only.
  - **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, the tool Create RRBS-fragment Track tool (see section 33.4.4) 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 33.28.

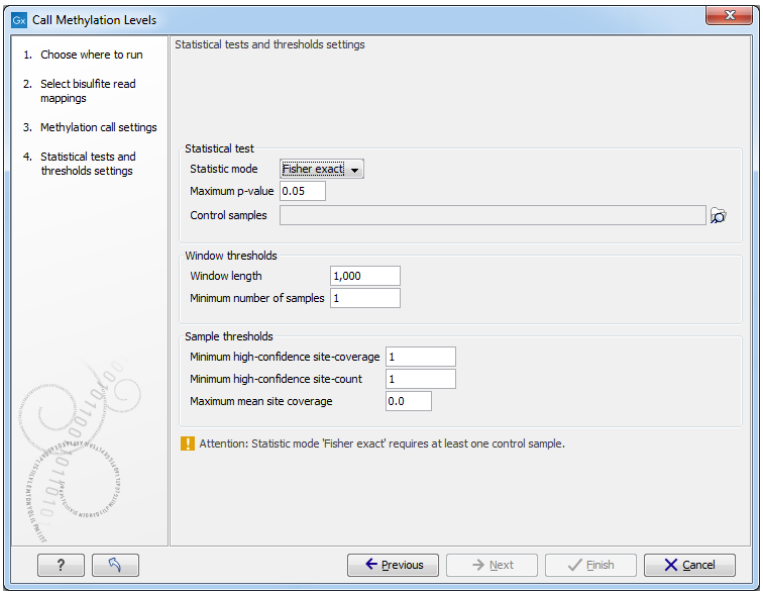


Figure 33.28: *The statistical tests and thresholds settings.*

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.

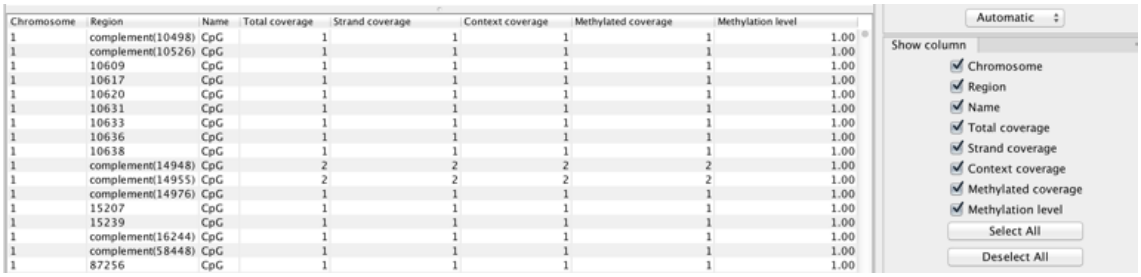
Window thresholds

- **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 chosen by default. It will provide a base level methylation track for each read mapping supplied, i.e., case or control (see figure 33.29 for a table view of the track).



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 33.29: *Output table.*

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.

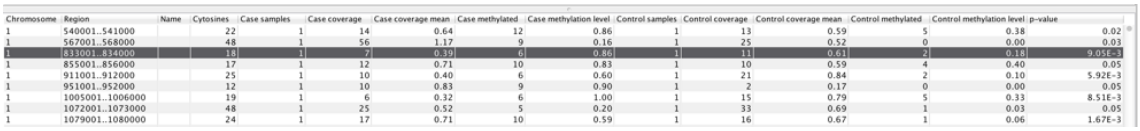
The columns of the methylation levels track table view indicate:

- **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 33.30.



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		18	1	7	0.39	6	0.86	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 33.30: *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.

33.4.4 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 33.31 shows the options for generating a track with predicted, and selected features corresponding to restriction digest of the selected genome.

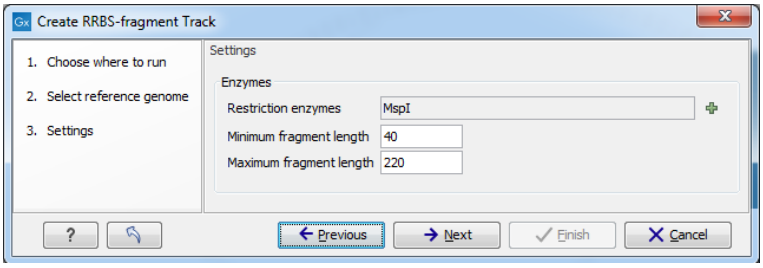


Figure 33.31: *Create RRBS-fragment track settings.*

Restriction enzymes The enzymes used in reduced-represention 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.

33.5 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 see 33.2)), 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:

Toolbox | **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.

33.5.1 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. The procedure used to build the peak shape filter is described here in 33.2.2)).

An example of such filter is shown in figure 33.32.

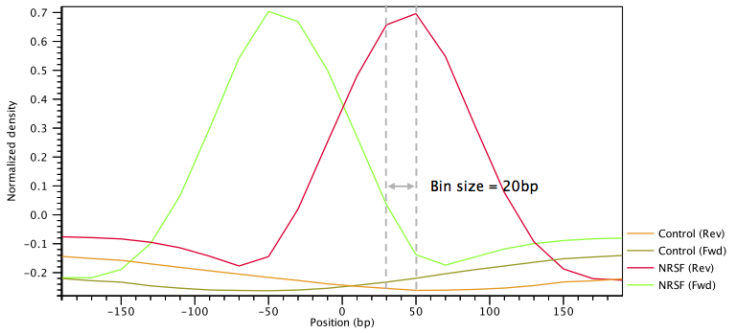


Figure 33.32: *Example of a peak shape filter with a window size of 400bp made up of 20 bins of size 20bp each. The filter was built from ChIP-Seq data of the transcription factor NRFS and a control ChIP-Seq experiment.*

To run the Learn Peak Shape Filter tool:

> **Toolbox** | **Epigenomics Analysis** ( ) | **Advanced Peak Shape Tools** ( ) | **Learn Peak Shape Filter** ( )

This will open up the wizard shown in figure 33.33 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.

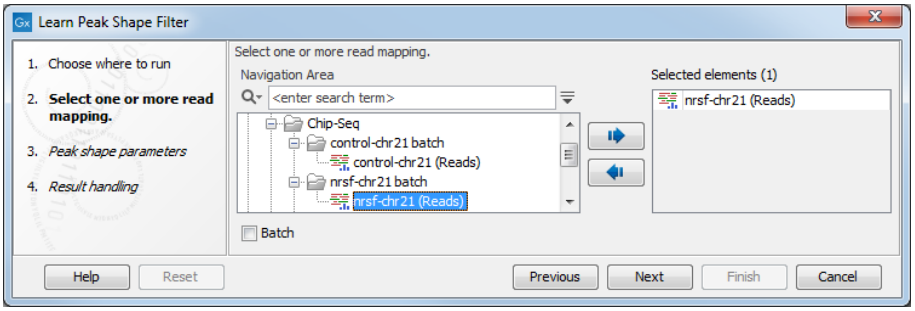


Figure 33.33: *Select the input data for the Learn Peak Shape Filter tool.*

Click **Next** to go to the next wizard step (shown in figure 33.34).

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 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

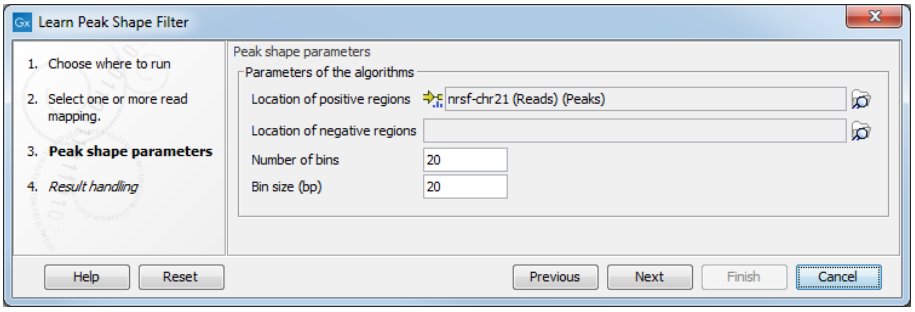


Figure 33.34: *Options for Learn Peak Shape Filter.*

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 Window size $=$ Bin size $\times$ Number of bins (see figure 33.32).

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.

33.5.2 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 in 33.2.3)).

To run the Apply Peak Shape Filter tool:

> **Toolbox** | **Epigenomics Analysis** ( ) | **Advanced Peak Shape Tools** ( ) | **Apply Peak Shape Filter** ( )

This will open up the wizard shown in figure 33.35 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.

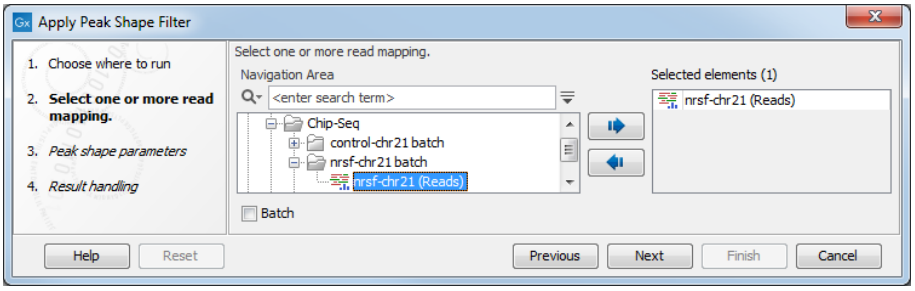


Figure 33.35: *Select the input data for Apply Peak Shape Filter.*

Click **Next** to go to the next wizard step (shown in figure 33.36).

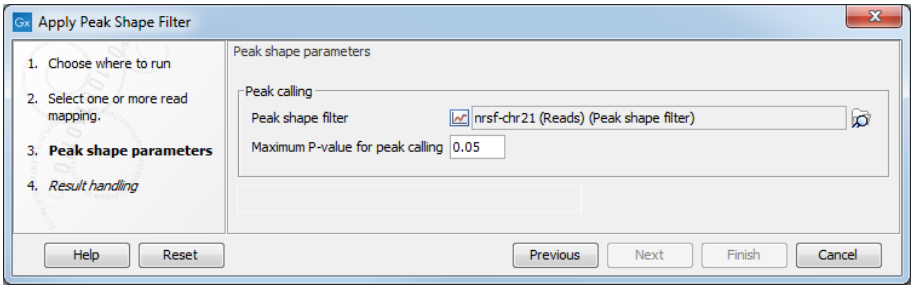


Figure 33.36: *Options for Apply Peak Shape Filter.*

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 33.37.

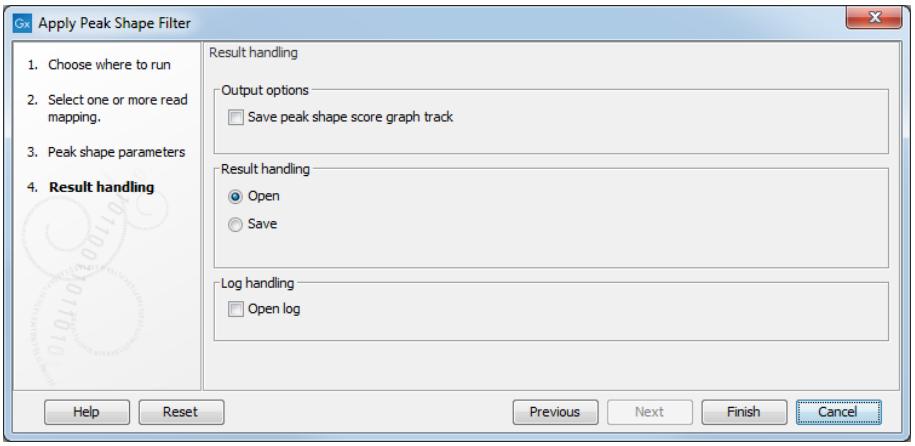


Figure 33.37: *Output options for Apply Peak Shape Filter.*

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.

For more information on the **Peak track** ( ), see in 33.2.5).

33.5.3 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:

> **Toolbox** | **Epigenomics Analysis** ( ) | **Advanced Peak Shape Tools** ( ) | **Score Regions** ( )

This will open up the wizard shown in figure 33.38 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.

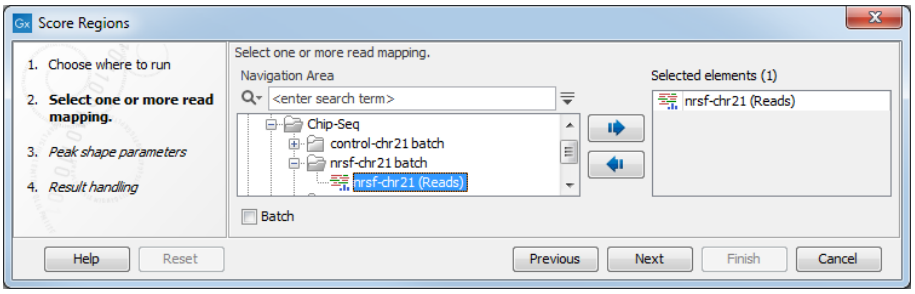


Figure 33.38: *Select the input data for Score Regions.*

Click **Next** to go to the next wizard step (shown in figure 33.39).

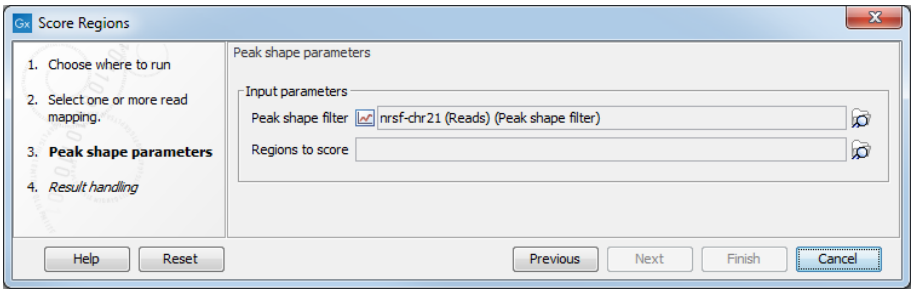


Figure 33.39: *Options for Score Regions.*

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 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 34

Utility tools

Contents

34.1 Batch Rename 963
34.2 Extract Annotations 969
34.3 Sample reads 970
34.4 Extract Reads 972
34.5 Merge Overlapping Pairs 974

34.1 Batch Rename

With the Batch Rename tool it is possible to rename your data in a batch fashion.

To run the batch rename tool:

Toolbox | **Utility Tools** ( ) | **Batch Rename** ( )

This will open the dialog shown in figure 34.1 where you can select the input data.

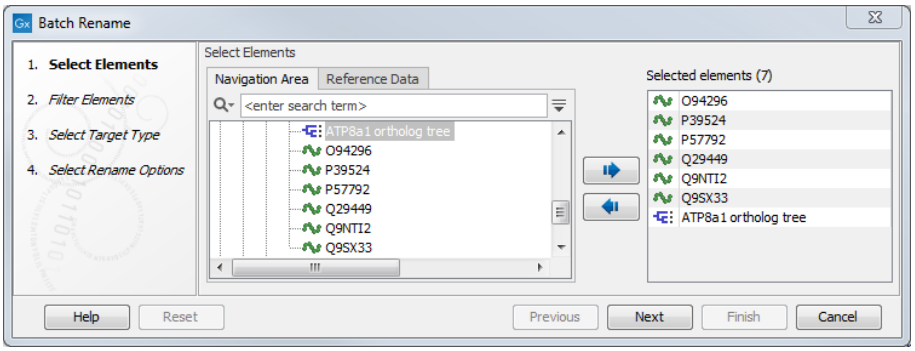


Figure 34.1: *Selecting the data you want to rename.*

Click **Next** to go to the next dialog (see figure 34.2).

Here, one can choose to include or exclude only some of the datat previously selected to work on. For small numbers of data elements, this would not usually be necessary. However, if many data objects were selected at the previous step (to save time when choosing many data elements)

you could use the include or exclude functionality at this point so that only certain data elements will be be acted on by the batch rename tool.

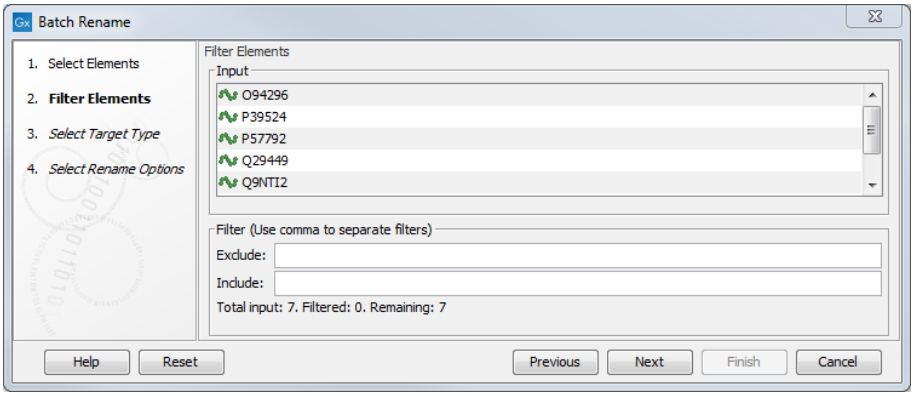


Figure 34.2: *Select how to filter the input data.*

The **Include** and **Exclude** filters take the text entered into the respective fields and search for matches in the names of the data elements selected in the first wizard step. Thus, you could enter the full names of particular data elements, or just partial names. Any elements where a match is found to the term or terms in the **Include** field will have the batch renaming applied to them. Any elements where a match is found to the term or terms in the **Exclude** field will not have the batch renaming applied to them.

For both filters, if you wish to filter on more than one term at the time, the individual terms must be separated with a comma - and without using a space after the comma. An example is shown in figure 34.3.

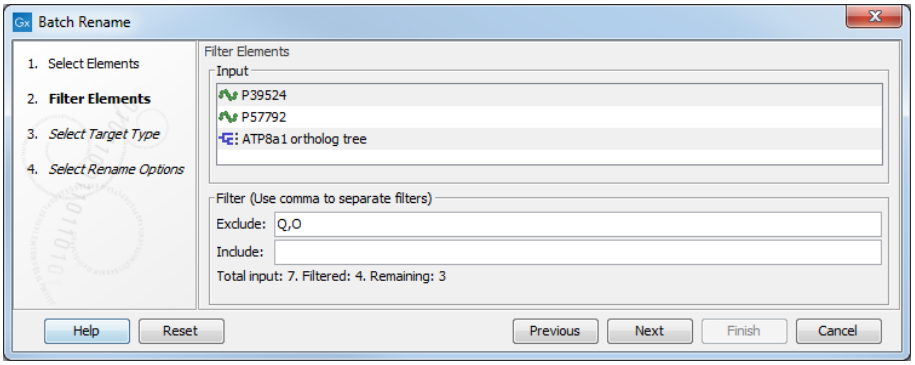


Figure 34.3: *Items remaining after filtering away items with a Q or an O in their name.*

In the "Select Target type" step, you can specify at which level you wish to perform the renaming. For a single sequence this is straightforward because it has just one name, and you would use the **Rename elements** option. But if you have a sequence list - as in the example shown in figure 34.4 - you could choose either to rename the list (using **Rename elements**) or the sequences in the list (using **Rename sequences in sequence lists**). The same goes for alignments (using **Rename sequences in alignments**) and read mappings (using **Rename reads in mappings**). For read mappings, there is also an option to **Rename reference sequence in mappings**.

Click **Next** to open the last dialog (see figure 34.5). For each text field, you can press Shift+F1 (Shift + Fn + F1 on Mac) to get a drop-down list of advanced placeholder options.

At this step you can select between three different renaming options.

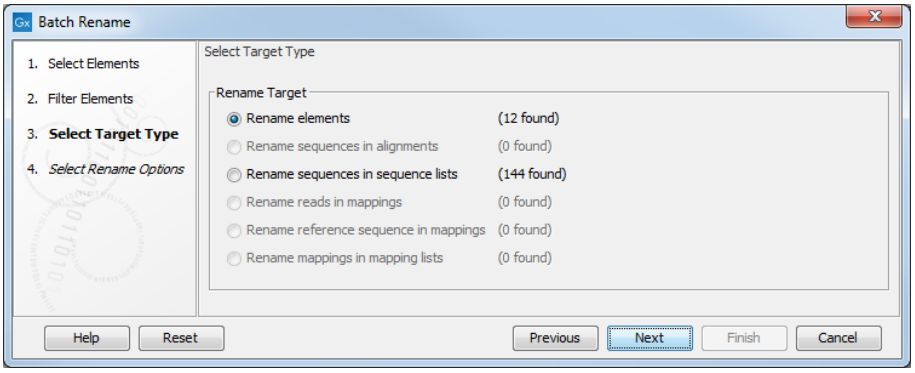


Figure 34.4: *In this example, as we only have one category represented, the other target type options are disabled.*

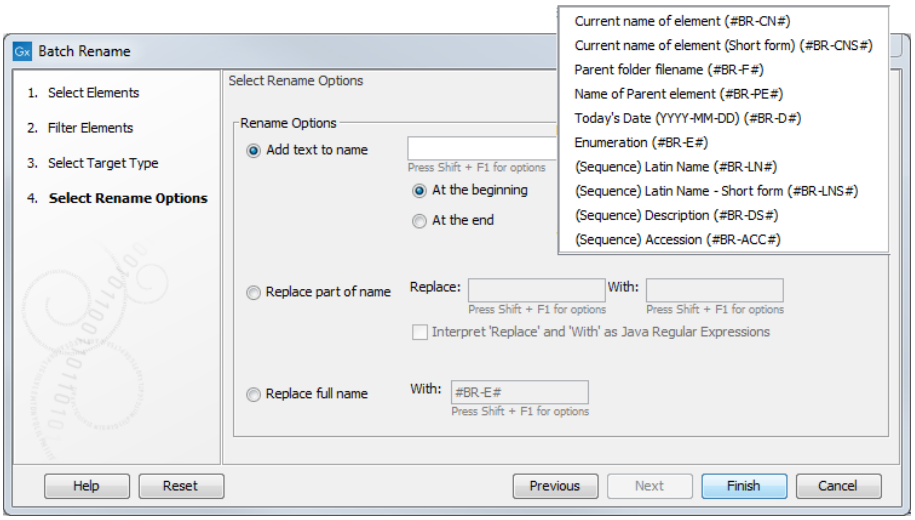


Figure 34.5: *Rename options. At this step you can choose whether you wish to add text to the existing name, replace some of the name, or replace the entire name.*

- **Add text to name** This option will add text at the beginning or the end of the existing name, depending on which you choose. Pressing Shift + F1 (Shift + Fn + F1 on Mac) will enable a list of different renaming options as shown in figure 34.5. The different options are presented as e.g. #BR-E#, which means "Batch Rename - Enumeration" = the current name is kept, and if "At the end" was selected, consecutive numbers will be added directly after the existing name (without introducing a space between the existing name and the new addition). Please note, that the numbering will follow the order of which the data were selected in the first dialog under "Select Elements".

- **Replace part of name** This is an advanced function that allows to replace e.g. data with completely different names in one go with a new name. This is shown in figure 34.6.

  The option to replace part of a name is based on regular expressions. Regular expressions allows you to describe text in a flexible manner. For more details, please see: `http://docs.oracle.com/javase/tutorial/essential/regex/`.

  By clicking in either the From: or To: box and pressing the Shift and F1 keys at the same time (Shift + Fn + F1 on Mac), you will see a drop down list of renaming possibilities. The options listed for the From: field are some of the most commonly used regular expressions. Other standard regular expressions are also admissible in this field. **Note!** Please ensure that if you choose any of these options, or other regular expressions, that you check in the

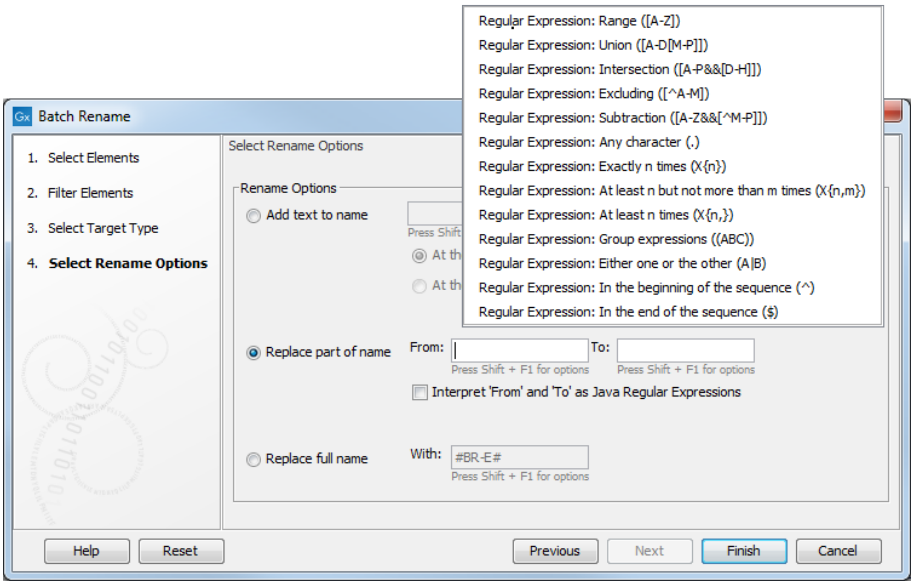


Figure 34.6: *Advanced options for rename of only part of a name.*

box labeled "Interpret From as Java Regular Expressions". If you do not check this box, then the characters you enter in the From box will be interpreted literally. For example, a full stop or period (.) is interpreted as that character (.) when this box is not checked, but is interpreted as meaning any single character when the box is checked.

An example: if you enter "From:" "Regular Expression: Range ([A-Z]) "To:" "Enumeration (#BR-E#)", titles containing any (capital) letters will be renamed to consecutive numbers. A more advanced example where the sequences shown in the images earlier in this manual are used is shown in figure 34.7. The sequence names contain both a capital letter, small letters and a number. In the first, the number is kept and a date is added in front of the number.

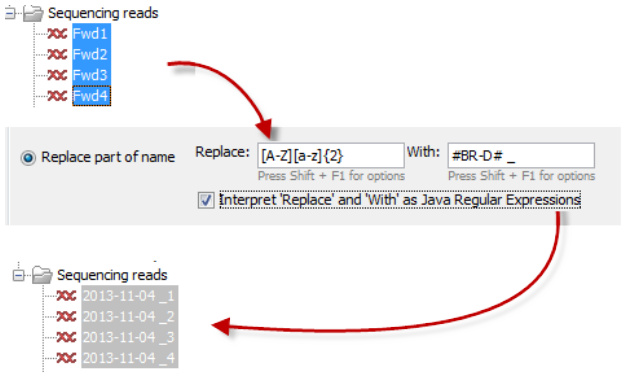


Figure 34.7: *The letters "Fwd" are removed from the primer name and replaced with "Today's date". In this case we have also inserted a space and a "_" between the date and the number, which was the only thing that was left from the original primer name. Please note that in order to be able to remove both of the small letters in the primer name, you have to add 2, which indicates that [a-z] should be performed twice.*

As this can be a bit difficult to grasp, we will look at three more examples. We will use the name **1N R1_0001** in the two first examples:

1. First we want to **keep only the first 4 none-whitespace characters of the name(s)**. To

do this write the following in the "From" and "To" fields shown in figure 34.7:

From: `(^\S{4}).*`

Nomenclature: `^: start of the line, \S: none-whitespace characters, {4}: 4 characters, .*: everything after the pattern`

To: `$1`

Nomenclature: `$1: the first group in the "From" field`

The result of this is that *1NR1* is kept, whereas the space between "N" and "R1" and *_0001* have been discarded.

2. If we would like to **keep only the last 4 characters of the name(s)**:

   From: `(.*)(.{4}$)`

   Nomenclature: `(.*): the first group, (.{4}$): any 4 character at the end of the line as the second group`

   To: `$2`

   Nomenclature: `$2: the second group in the "From" field`

   The result of this is that *0001* is kept, whereas *1N R1_* has been discarded.

3. Now we would like to **replace the first letter followed by 9 numbers in the name "p140101034_1R_AMR" with the parent folder name, which in this case is "AmericanSamples"**.

   From: `([A-Z]\d{9})(.*)`

   Nomenclature: `[A-Z]: any character`

   (as long as they are part of [A-Z][a-z], the CLC software do not differentiate between upper/lower case),

   `\d{9}: any 9 digit numbers, (.*): the second group of the name (anything after the "([A-Z]\d{9})" pattern ).`

   To: `#BR-F#$2`

   Nomenclature: `$2: the second group in the "From" field.`

   The result of this is that we have replaced "p140101034" with "AmericanSamples" and as a result have changed the name from p140101034_1R_AMR to AmericanSamples_1R_AMR.

4. If we want to extract and use the text "sample-code" for the new name from the following "1234_sample-code_5678" with Java Regular Expressions:

   From: `(^[^_]+)_([^_]+)_(.*)`

   Nomenclature: `^[^_]+`: Starting from the beginning and match anything before the first underscore,

   `[^_]+`: will match anything and stop before the second underscore.

   After the second underscore, the match will include the rest of the name.

   To: `$2`

   Nomenclature: `$2`: includes the match from the second group, which is flanked between the first and the second underscore symbols of the name.

- **Replace full name** Allows replacement of the entire name with the name that is either typed directly into the text field, or with options that can be selected when pressing Shift + F1 (Shift + Fn + F1 on Mac). Figure 34.8 shows an example where a combination of "Shift +F1" (Shift + Fn + F1 on Mac) options (#BR-D# and#BR-E#) are used together with user-defined text (RNA-Seq).

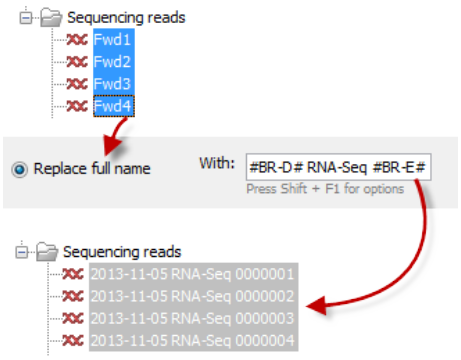


Figure 34.8: *The entire name is removed from the primer names and is replaced with "Today's date" (#BR-D#), the userdefined text: RNA-Seq, and the addition of consecutive numbers (#BR-E#). In this case we have inserted a space between the date, the user-defined text and the added number. If commas were inserted instead, the commas would be part of the new name as everything that is typed into the text field will be used in the new name when renaming the entire name.*

Click **Finish** to start renaming. Please note that the **rename cannot be undone** and that it does not show up in the **History** ( ).

34.2 Extract Annotations

The **Extract annotations** tool makes it very easy to extract parts of a sequence (or several sequences) based on its annotations. In just a few steps, it becomes possible to:

- extract all tRNA genes from a genome.
- automatically add flanking regions to the annotated sequences.
- search for specific words in all available annotations.
- extract nucleotide sequences of differentially expressed genes or transcripts when using RNA-seq statistical comparisons as input.

The output is a sequence list that contains sequences carrying the annotation specified (including the flanking regions, if this option was selected).

To extract annotations from a sequence, go to:

Toolbox | **Utility Tools** ( ) | **Extract Annotations ( )**

This opens the dialog shown in figure 34.9 where you can select one or more annotated sequences, or annotation tracks, or variant tracks, or statistical comparisons. Click **Next**.

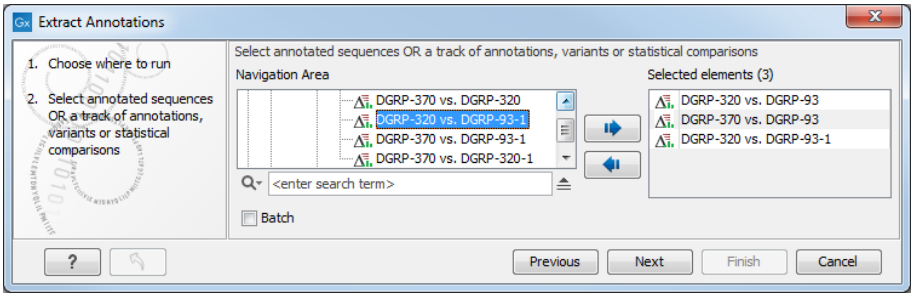


Figure 34.9: *Select one or more annotated sequences, annotations, variant tracks or (in this figure) statistical comparisons.*

If you selected tracks as input, the next step will ask for a reference sequence track to use for extracting the annotations, or which annotations to use if an annotated sequence was selected as input (figure 34.10).

- **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:

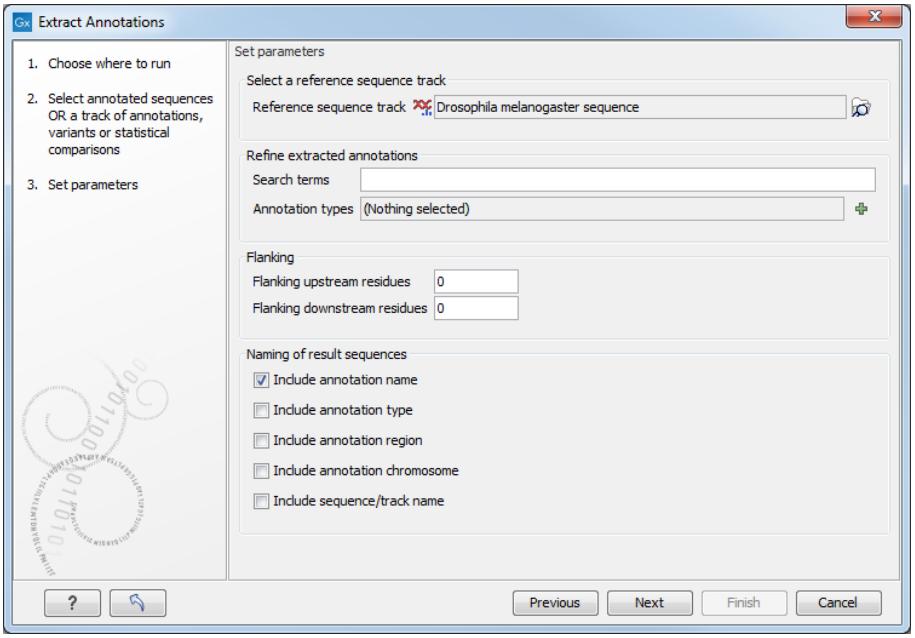


Figure 34.10: *Adjusting parameters for extract annotations.*

- **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.

34.3 Sample reads

The **Sample Reads** tool is very useful for reducing the size of data sets. It allows the user to reduce the number of reads in a sequence list to an absolute number of reads, or a pre-defined percentage of the original list.

A reduced sequence list can be relevant when editing contigs with a high coverage, as an excessive amount of reads can be overly resource-demanding and can hinder read mapping editor functions. Sampling reads, and subsequently mapping the sampled reads to contigs with the Map Reads to Contig tool can maintain the speed of many edit operations in the contig aligner without any loss of important information. Similarly, in the case of de novo assembly, very high coverage in a location will increase the probability this location is seen as a sequencing error. You can thus use the Sample Reads tool to reduce coverage to a maximum average coverage of 100x using the following calculation:

1. Multiply the estimated size of the genome you intend to assemble by 100. This value will be the total number of bases you should use as input for assembly.
2. Divide the total input bases the average length of your sequencing reads.
3. Use this number as input for the number of reads to obtain as output from the Sample Reads tool.

To learn more about the Map Reads to Contig tool and the De Novo Assembly tool, see chapter 32.

To run the **Sample Reads** tool: **Toolbox** | **Utility Tools** ( ) | **Sample reads** ( )

This opens the dialog shown in figure 34.11. Select the relevant reads and click **Next**.

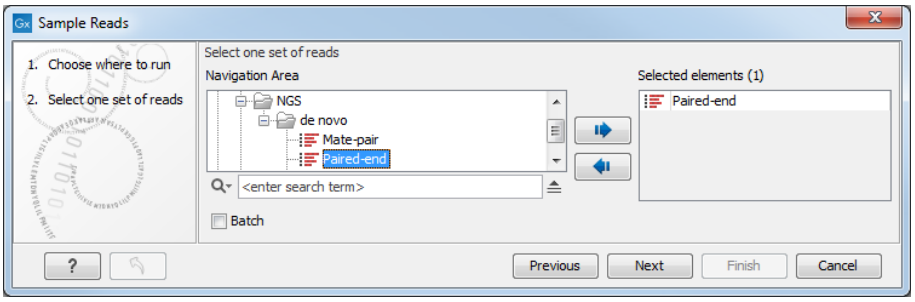


Figure 34.11: *Select the reads.*

This leads to the **Sample size parameters** step shown in figure 34.12.

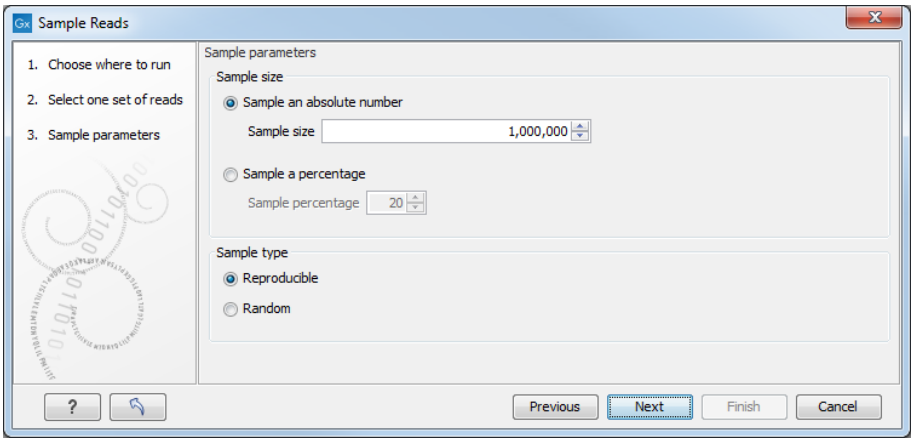


Figure 34.12: *Specify the sample size parameters.*

In the **Sample size parameters** it is possible to sample down to either an absolute number of reads, or a sample percentage of the original file, with sample percentage defining the percentage of the sub-sample paired reads (for single end reads) and pairs of reads (for paired reads).

It is also possible to choose between a **Reproducible** or **Random** output. The Reproducible option will always returns the same sample (i.e., containing the same reads) for a given sample size or percentage; the Random option returns a different sample each time the tool is run.

After the **Result handling** step, click **Finish**.

34.4 Extract Reads

This tool can be used to extract subsets of reads based on annotations. When extracting reads with a specific annotation, the annotation will function as a tag pulling out all the reads with the overlapping annotation (or, when handling paired read data, all the pairs of reads). To launch the tool, go to:

Toolbox | **Utility Tools** ( ) | **Extract Reads** ( )

Read mapping tracks can be used as input (figure 34.13).

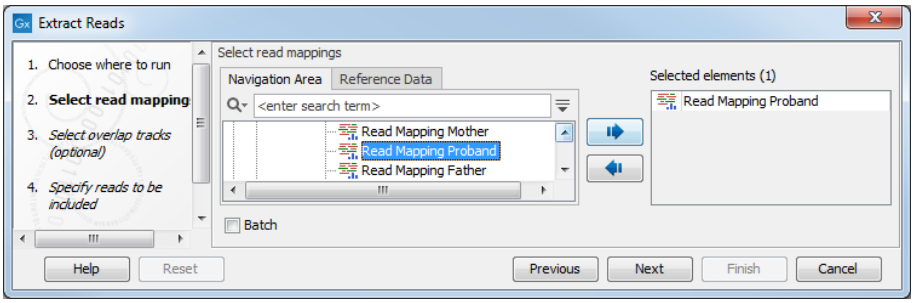


Figure 34.13: *Select a read mapping. Only one read mapping can be selected at the time.*

The next step offers the possibility to select one or moe annotated track(s) to be used for pulling out reads and specify which reads to include (figure 34.14). Note that it is also possible to select here a RNA-seq statistical comparison.

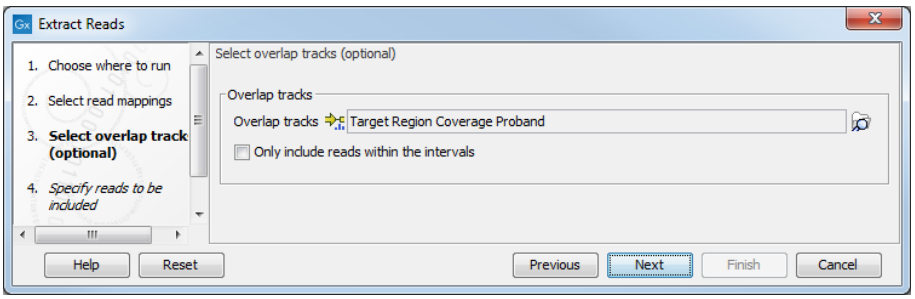


Figure 34.14: *Select the track(s) containing the annotation(s) of interest. Multiple tracks can be selected at the same time.*

With the options "Only include reads within the intervals", it is possible to select whether only reads within the intervals should be extracted, or whether reads continuing outside the annotated region should be extracted. The difference between the options can be seen in figure 34.15.

In the next dialog, specify which reads should be included in the output. They are all selected by default as seen in figure 34.16.

Match specificity

- Include specific matches Reads that only are mapped to one position.



Figure 34.15: *Output from the Extract Reads tool. Top: The read mapping used as input. Middle: Output when "Only include reads within intervals" has been deselected. Bottom: Output when "Only include reads within intervals" has been selected.*

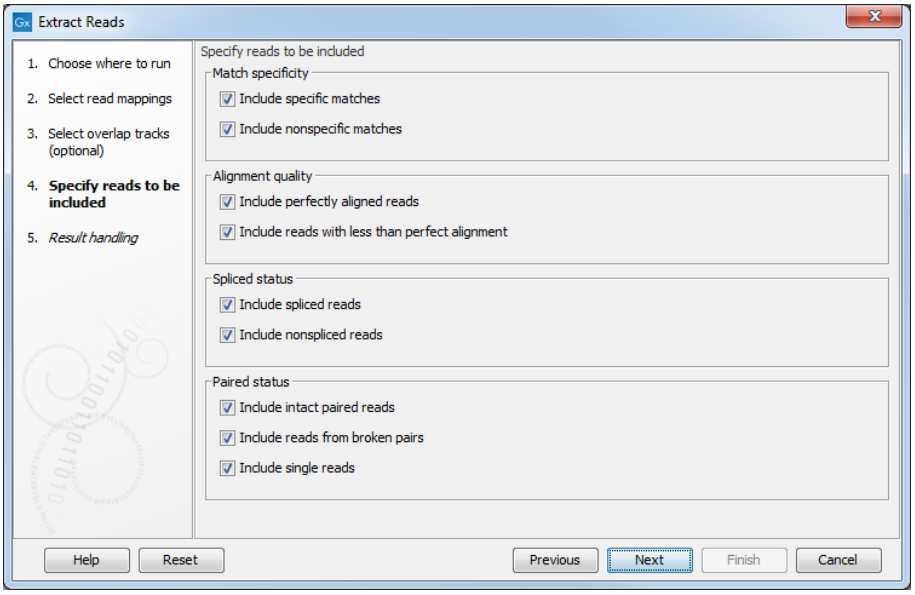


Figure 34.16: *Options to include or exclude specific types of reads from the output.*

- Include non-specific matches Reads that have multiple equally good alignments to the reference. These reads are colored yellow per default.

Alignment quality

- Include perfectly aligned reads Reads where *the full read* is perfectly aligned to the reference sequence (or consensus sequence for de novo assemblies). Note that at the end of the contig, reads may extend beyond the contig (this is not visible unless you make a selection on the read and observe the position numbering in the status bar). Such reads are not considered perfectly aligned reads because they don't align

in their entire length.

- Include reads with less than perfect alignment Reads with mismatches, insertions or deletions, or with unaligned nucleotides at the ends (the faded part of a read).

Spliced status

- Include spliced reads Reads that are across an intron.
- Include non spliced reads Reads that are not across an intron.

Paired status

- Include intact paired reads When paired reads are placed within the paired distance specified, they will fall into this category. Per default, these reads are colored in blue.
- Include reads from broken pairs When a pair is broken, either because only one read in the pair matches, or because the distance or relative orientation is wrong, the reads are placed and colored as single reads, but you can still extract them by checking this box.
- Include single reads This will include reads that are marked as single reads (as opposed to paired reads). Note that paired reads that have been broken during assembly are not included in this category. Single reads that come from trimming paired sequence lists are included in this category.

In the last step, you can choose to output a reads track or sequence list(s).

34.5 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*, and it will work perfectly fine (see details for variant detection in section 28.1.3).

However, in some situations it can be useful to merge the overlapping pair into one sequence read instead. The benefit is that you get longer reads, and that the quality improves (normally the quality drops towards the end of a read, and by overlapping the ends of two reads, the consensus read now reflects two read ends instead of just one).

In the *CLC Genomics Workbench*, there is a tool for merging overlapping reads, which are in forward-reverse orientation:

Toolbox | **Utility Tools** () | **Merge Overlapping Pairs** ()

Select one or more sequence lists with paired end sequencing reads as input.

Please note that read pairs have to be in forward-reverse orientation. Please also note that after merging the merged reads will always be in the forward orientation. As an aside, 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 nucleotide**s option.

Clicking **Next** allows you to set parameters as displayed in figure 34.17.

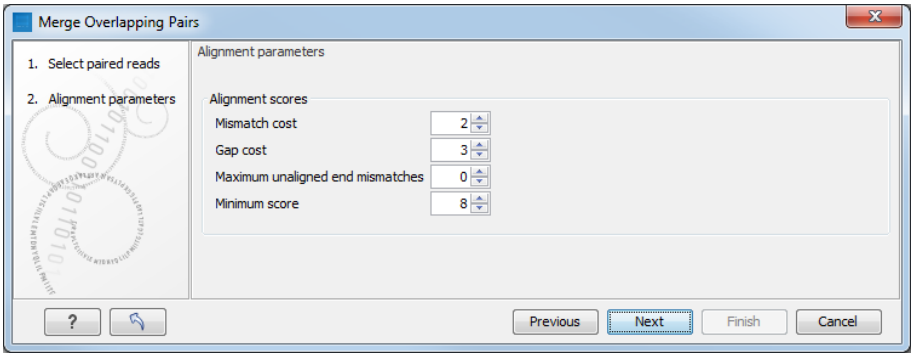


Figure 34.17: *Setting parameters for merging overlapping pairs.*

In order to understand how these parameters should be set, an explanation of the merging algorithm is needed: 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 parameters are used to 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 clicking **Next** you can select whether a report should be generated as part of the output. 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 34.18 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 34.5.
- A plot of read lengths. This shows the distribution of read lengths for the pairs that have been merged:
  - The length of the overlap.

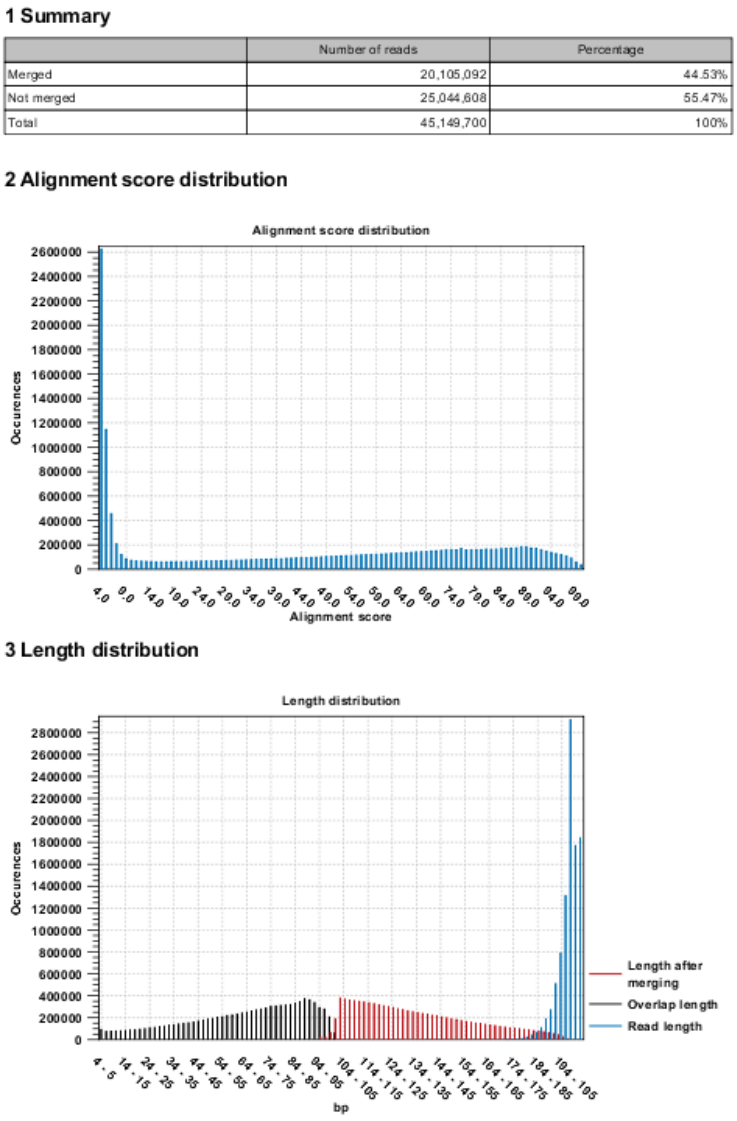


Figure 34.18: *Report of overlapping pairs.*

– The length of the merged read.
– The combined length of the two reads in the pair before merging.

Chapter 35

Legacy tools

Contents

35.1 Compare Sample Variant Tracks . . . 978
35.2 Remove Reference Variants . . . 980
35.3 Import Roche 454 . . . 980
35.4 Create Combined RNA-Seq Report . . . 982
35.5 Create Track from Experiment . . . 984
35.6 Small RNA Analysis . . . 989
35.6.1 Extract and count . . . 990
35.6.2 Downloading miRBase . . . 993
35.6.3 Annotating and merging small RNA samples . . . 995
35.6.4 Working with the small RNA sample . . . 1004
35.6.5 Exploring novel miRNAs . . . 1006
35.7 Batch launching workflows with multiple inputs . . . 1007

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 of the Workbench Toolbox, 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.

35.1 Compare Sample Variant Tracks

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 of the Toolbox, and its name has "(legacy)" appended to it. If you have concerns about the future retirement of this tool, please contact QIAGEN Bioinformatics Support team at ts-bioinformatics@qiagen.com.

The **Compare Sample Variant Tracks** tool allows you to compare two samples and filter away the variants that are either identical or different. If you want to focus on variants that are found in both samples you should choose the option "Keep variants that are the same". These variants will be the same no matter which sample was selected as the first input (the "variant track") or the second input (the "comparison track", which also is a variant track).

If you want to focus on variants that differ between the two tracks, the order in which the two samples are selected matters, as the tool will not combine the variants in the two tracks, but will filter out one. This is illustrated in figure 35.1.

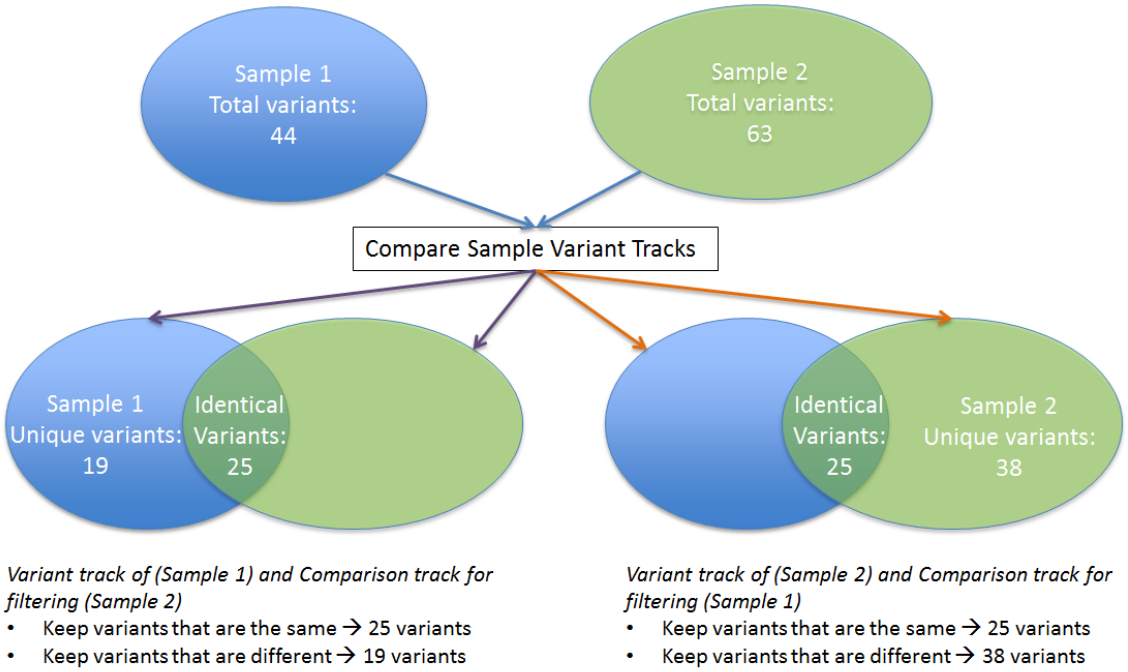


Figure 35.1: *The figure illustrates the different types of output you will get depending on which choices you have made in the wizard steps.*

To run the tool:

Legacy Tools () | **Compare Sample Variant Tracks**

In the first step of the dialog, you select the variant track that should be taken as input. Clicking **Next** shows the dialog in figure 35.2.

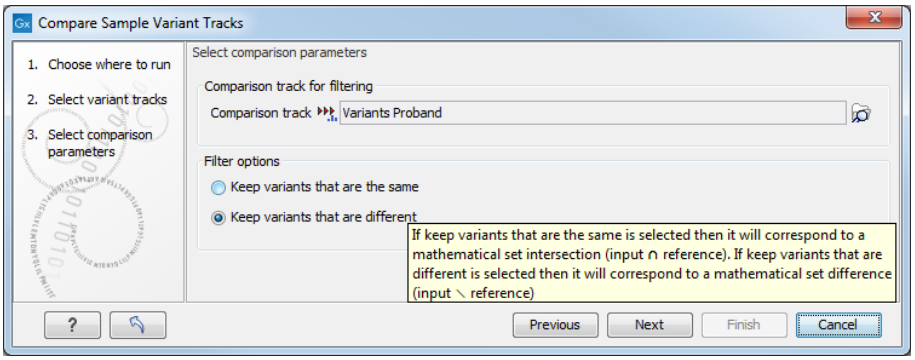


Figure 35.2: *Comparing against variants in "Variants Proband". A tooltipdescribes the difference between keeping variants that are the same or the ones that are different.*

At the top, select the comparison track. Below, you can choose whether the result should be the variants from the input that match the comparison track, or whether it should be the variants that are different from the variant track. The match criterion here is an exact match on the position and allele sequence.

35.2 Remove Reference Variants

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 of the Toolbox, and its name has "(legacy)" appended to it. If you have concerns about the future retirement of this tool, please contact QIAGEN Bioinformatics Support team at ts-bioinformatics@qiagen.com.

The variant tracks produced by the variant detection tools of *CLC Genomics Workbench* include reference alleles complementing a non-reference allele (i.e. a heterozygous variant where only one allele is different from the reference). In some situations, this information is not necessary, and these reference allele variants can be filtered away.

Legacy Tools | **Remove Reference Variants**

This opens a dialog where you can select a variant track that should be filtered.

Click **Next** and **Finish** to create a new track without the reference variants.

For removing reference variants whose alternate allele has already been filtered, use the tool Remove Homozygous Reference Variants. It is also possible to configure Filter on Custom Criteria with a criteria set to "`Reference`allele = No" to mimic the functionality of Remove Reference Variants.

35.3 Import Roche 454

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 of the Toolbox, and its name has "(legacy)" appended to it. If you have concerns about the future retirement of this tool, please contact QIAGEN Bioinformatics Support team at ts-bioinformatics@qiagen.com.

To open the Roche 454 Import tool, go to :

Legacy Tools | **Roche 454**

Choosing the Roche 454 import will open the dialog shown in figure 35.3.

We support import of two kinds of data from 454 GS FLX systems:

- Flowgram files (`.sff`) which contain both sequence data and quality scores amongst others. However, the flowgram information is currently not used by *CLC Genomics Workbench*. There is an extra option to make use of clipping information (this will remove parts of the sequence as specified in the .sff file).
- Fasta/qual files:
  - 454 FASTA files (`.fna`) which contain the sequence data.
  - Quality files (`.qual`) which contain the quality scores.

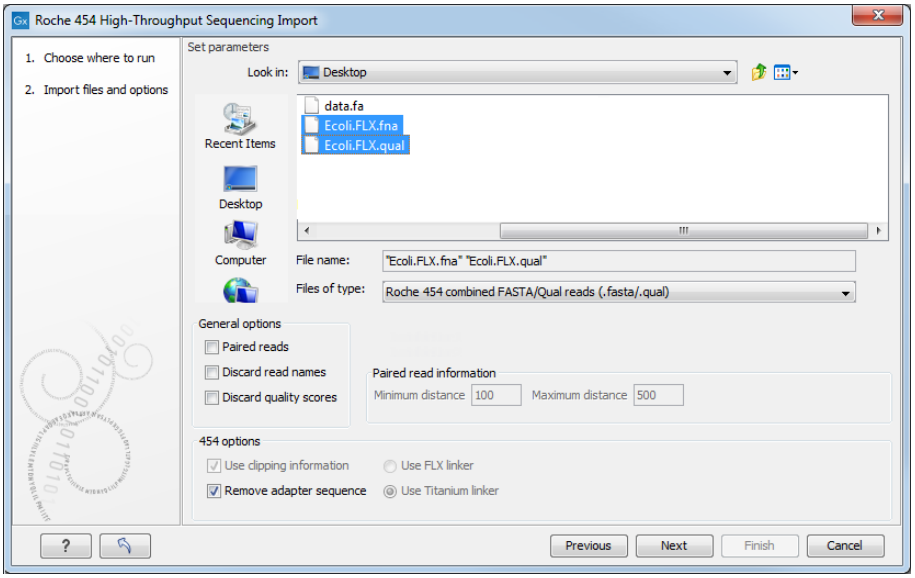


Figure 35.3: *Importing data from Roche 454.*

– Compressed data in gzip format is also supported (.gz).

The **General options** to the left are:

- **Paired reads**. The paired protocol for 454 entails that the forward and reverse reads are separated by a linker sequence. During import of paired data, the linker sequence is removed and the forward and reverse reads are separated and put into the same sequence list (their status as forward and reverse reads is preserved). You can change the linker sequence in the **Preferences** (in the **Edit** menu) under **Data**. Since the linker for the FLX and Titanium versions are different, you can choose the appropriate protocol during import, and in the preferences you can supply a linker for both platforms (see figure 35.4. Note that since the FLX linker is palindromic, it will only be searched on the plus strand, whereas the Titanium linker will be found on both strands. Some of the sequences may not have the linker in the middle of the sequence, and in that case the partial linker sequence is still removed, and the single read is put into a separate sequence list. Thus when you import 454 paired data, you may end up with two sequence lists: one for paired reads and one for single reads. Note that for de novo assembly projects, only the paired list should be used since the single reads list may contain reads where there is still a linker sequence present but only partially due to sequencing errors. Read more about handling paired data in section 6.3.7.

- **Discard read names**. For high-throughput sequencing data, the naming of the individual reads is often irrelevant given the huge amount of reads. This option allows you to discard this option to save disk space.

- **Discard quality scores**. Quality scores are visualized in the mapping view and they are used for SNP detection. If this is not relevant for your work, you can choose to **Discard quality scores**. One of the benefits from discarding quality scores is that you will gain a lot in terms of reduced disk space usage and memory consumption. If you have selected the fna/qual option and choose to discard quality scores, you do not need to select a .qual file.

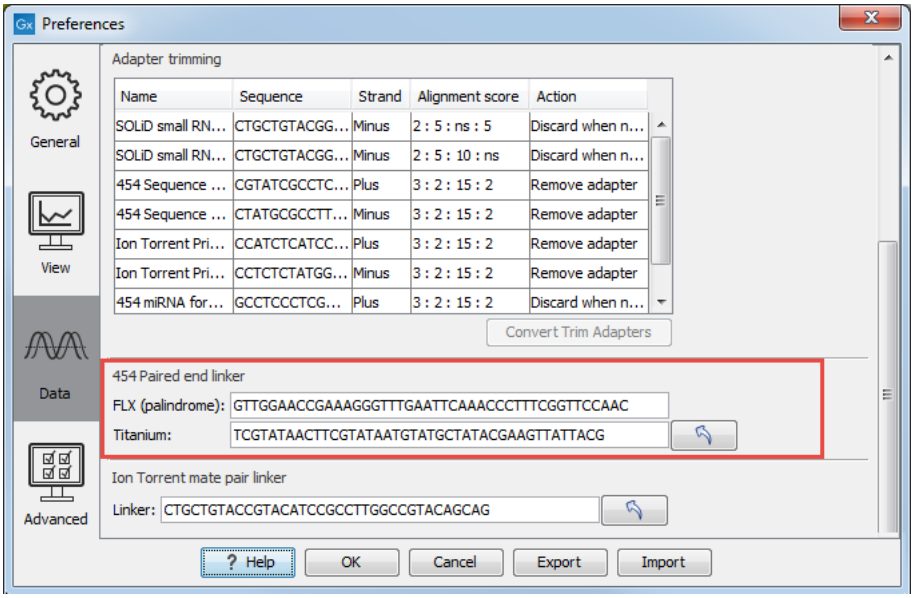


Figure 35.4: *Specifying linkers for 454 import.*

Note! During import, partial adapter sequences are removed (`TCAG` and `ATGC`), and if the full sequencing adapters `GCCTTGCCAGCCCGCTCAG`, `GCCTCCCTCGCGCCATCAG` or their reverse complements are found, they are also removed (including tailing Ns). If you do not wish to remove the adapter sequences (e.g. if they have already been removed by other software), please uncheck the **Remove adapter sequence** option.

Click **Next** to adjust how to handle the results (see section 9.2)). 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 handy for better organizing subsequent analysis results and for batching (see section 9.3).

35.4 Create Combined RNA-Seq Report

Create Combined RNA-Seq Report has been deprecated and will be retired in a future version of the software. Please use the Combine Reports tool (see section 26.4) instead.

Create Combined RNA-Seq Report has been moved to the **Legacy Tools** ( ) folder of the Toolbox, and its name has "(legacy)" appended to it. If you have concerns about the future retirement of this tool, please contact QIAGEN Bioinformatics Support team at ts-bioinformatics@qiagen.com.

With the Create Combined RNA-Seq Report tool, you can generate an overview of several RNA-seq experiments by combining in one document several RNA-Seq reports. The description of the various sections included in the report, as well as the cutoff values that trigger warnings for sub-optimal data are the same as described for RNA-Seq reports (see section 30.2.9). The combined report can be exported in PDF or Excel format.

To start the tool:

Toolbox | **Legacy Tools** ( ) | **Create Combined RNA-Seq Report** ( )

In the wizard that opens (figure 35.5), select several RNA-seq reports and click **Next**. Note that

while it seems possible to select any kind of reports in that dialog, only RNA-seq reports generated by *CLC Genomics Workbench 21.0.1* or higher will be compiled in the combined RNA-seq report.

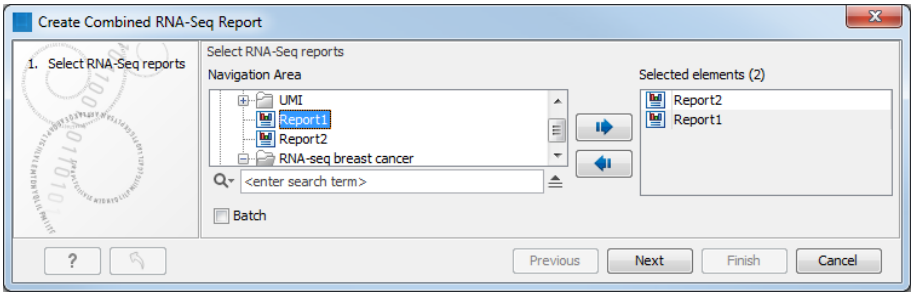


Figure 35.5: *The Create Combined RNA-Seq Report tool.*

In the Result handling window, choose whether you want to **Open** or **Save** the combined report. When saving, specify where you would like to save the combined report in the Navigation Area, and click **Finish**.

The results of the individual reports are compiled in tables, and organized in the same sections than the ones were present in the individual input reports. This layout enables a quick overview of a series of experiments as well as the different troubleshooting options for dealing with sub-optimal data quality (see figure 35.6 for an example of the first four sections of a combined report).

1 Read count statistics

Sample name	Read count	Single, mapped %	Single, unmapped %	Paired, mapped pairs %	Paired, broken pairs %	Paired, not mapped %
Report2	1,000,000	N/A	N/A	39.98	3.69	N/A
Report1	2,758,262	N/A	N/A	93.21	4.13	N/A

For paired data, there are two reads in a pair.

2 Fragment counting statistics

Sample name	Mapped to genes %	Mapped to intergenic %
Report2	67.50	32.50
Report1	98.51	1.49

Default counting scheme ('Fragment counts'): An intact pair is counted as one, broken pairs are ignored.

3 Spike-in quality control

Sample name	Detected spike-ins	R^2	Reads mapped to spike-ins	% of reads mapped to spike-ins	Lower limit of detection (attomoles/µL)
Report2	46/92	0.05	7,336	3.10	0.03
Report1	54/92	0.97	24,667	1.76	7.32

R^2: The sample has poor correlation ($R^2 < 0.8$) between known and measured spike-in concentrations. This may indicate problems with the spike-in protocol, or a more serious problem with the sample.
- Check that the correct spike-in file has been selected.
- Check the integrity of the sample RNA.

Reads mapped to spike-ins: Fewer than 10000 reads mapped to spike-ins.
- Check that the correct spike-in sequences are specified.
- Consider using more spike-in mix in future experiments.

4 Strand specificity

Sample name	Strand specific setting	Forward % of reads mapped	Reverse % of reads mapped	Ignored reads (wrong strand)	Ignored reads % (wrong strand)
Report2	Both	50.50	49.50	0	0.00
Report1	Both	96.01	3.99	1,644	0.06

Strand specific setting: >90% of reads were mapped in the same orientation. Consider re-running the tool with a strand specific setting ("Forward"/" Reverse").

Figure 35.6: *An example combined report with pink highlights where data is of sub-optimal quality.*

35.5 Create Track from Experiment

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 of the Toolbox, and its name has "(legacy)" appended to it. If you have concerns about the future retirement of this tool, please contact QIAGEN Bioinformatics Support team at ts-bioinformatics@qiagen.com.

The **Create Track from Experiment** tool in the *CLC Genomics Workbench* enabled the conversion of experiments to tracks.

You can find the **Create Track from Experiment** tool here:

Toolbox | **Legacy Tools** | **Create Track from Experiment** .

After you start the tool, you are presented with a wizard where you can choose the experiment that you would like to create a track of. The **Create Track from Experiment** tool can be run on experiments with associated genomic information, such as those created using expression

tracks from the **RNA-Seq Analysis** tool.

In the case where the experiment has associated genomic information, the **Create Track from Experiment** tool will automatically infer these and the wizard will jump directly to the filtering step, as shown in figure 35.8.

In the case where the experiment does not have associated genomic information, you will first need to specify how the genomic information should be obtained in the parameters step of the **Create Track from Experiment** tool (figure 35.7).

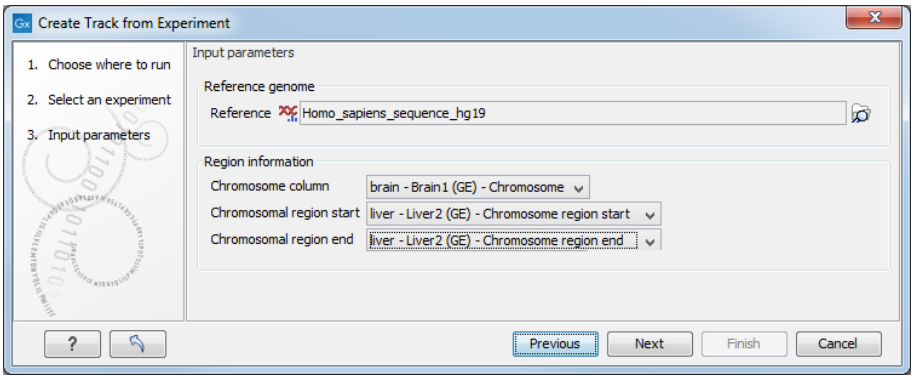


Figure 35.7: *The "Input parameters" step in the Create Track from Experiment tool.*

In the Input parameters step, you must specify the following parameters:

- **Reference genome.** The chosen genome will be used as the reference genome for the resulting track.
- **Chromosome column.** The column containing the chromosome names must be chosen from the drop-down menu.
- **Chromosomal region start.** The column containing the start of the genomic regions must be chosen from the drop-down menu.
- **Chromosomal region end.** The column containing the end of the genomic regions must be chosen from the drop-down menu.

Note! The drop-down menus will only contain the columns that potentially represent the information required by the given parameter. If the experiment does not contain any columns that potentially represent the required genomic information, the drop-down menus may appear empty. In this case, it is not possible to convert the given experiment to a track.

In the filtering step (figure 35.8), you have the following options:

- **Filter based on statistical analysis results** This allows to filter which annotations are transferred to the track on the basis of the statistical analysis. To enable filtering, check the **Filter based on statistical analysis results** checkbox. The filtering option is only available if a statistical analysis has previously been carried out on the experiment, and the drop-down menu will only contain the statistical analyses that are present on the experiment.
- **Statistical analysis** Allows you to choose statistical analysis from the drop-down list. The selection of available statistical analyses depends on which tests have been used when you set up the experiment that you are about to convert to track format.

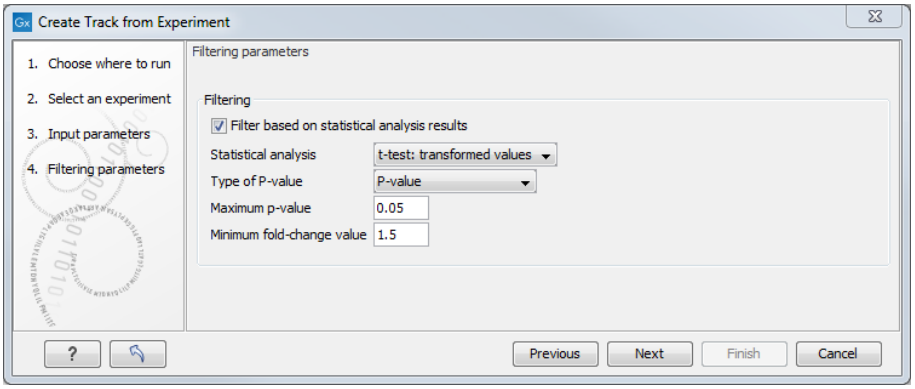


Figure 35.8: *The filtering step in the Create Track from Experiment tool.*

- **Type of p-value** This drop-down menu allows you to select between raw and corrected p-values. Only the types of p-values available for the given statistical analysis will be present in the drop-down menu.
- **Maximum p-value** In this input field, you can enter the maximum allowed p-value, as a number between 0 and 1. If you do not want any filtering based on p-value, enter 1.
- **Minimum fold-change value** You can also specify the minimum allowed fold-change value as a number greater than zero. If you do not want any filtering based on fold-change, enter 0.

You can then select in the drop-down menu which analysis you want to use for filtering.

The fold change values are stored as different columns in the experiment, depending on the statistical analysis performed. The Create Track from Experiment tool will automatically use the fold-change column appropriate for the different statistical analyses:

- Kal's Z-test (see section 31.5.2): Proportions fold change.
- Baggerley's test(see section 31.5.2): Weighted proportions fold change.
- T-test (see section 31.5.3): Fold change.
- ANOVA (see section 31.5.3): Max fold change.
- Empirical analysis of DGE (see section 31.5.1): Fold change.

The resulting track will contain only differentially expressed genes whose p-value is lower than the specified threshold and whose fold-enrichment is above the specified threshold.

If the chosen statistical analysis was performed on several pairs of groups, there will be an output track for each tested pair of groups. For example, if the same statistical analysis has been carried out on 'group 1 vs. group 2' and 'group 1 vs. group 3', then the output will contain two tracks, where one is filtered according to the 'group 1 vs. group 2' analysis results and the other one is filtered according to the 'group 1 vs. group 3' analysis results.

When running the **Create Track from Experiment** tool as part of a workflow, there are a few differences in how the parameters are set (see figure 35.9).

- The **Source of genomic information** parameter determines the behavior of the algorithm if the incoming experiment is *not* coupled to a genome. If the value of this parameter is set to **Require genomic information in experiment**, then the algorithm will expect the incoming experiment to be coupled to a genome, and will fail with an error alerting the user in case the experiment does not fulfill this criterion. If the value of the parameter is set to **Automatic: use genomic information if available**, then the algorithm will still use the genomic information in a genome-coupled experiment. But if this information is not available, the algorithm will attempt to use the information specified by the user in the workflow parameters. *Note:* If the incoming experiment *is* coupled to a genome (as will usually be the case), the value of this parameter makes no difference.

- In a workflow setting, the column titles for the chromosome, region end and region start fields can be specified as texts. These fields may be left empty, if the incoming experiment contains the genomic information. If filling out these fields, note that the format for this text is very strict, and must exactly match the text appearing in the drop-down menu when running the tool from the toolbox. For example, if 'Chromosome' is a sample-specific column, for a sample called 'Liver (GE)' in the 'liver' group in the experiment, then the column name text will be: 'liver - Liver (GE) - Chromosome'.

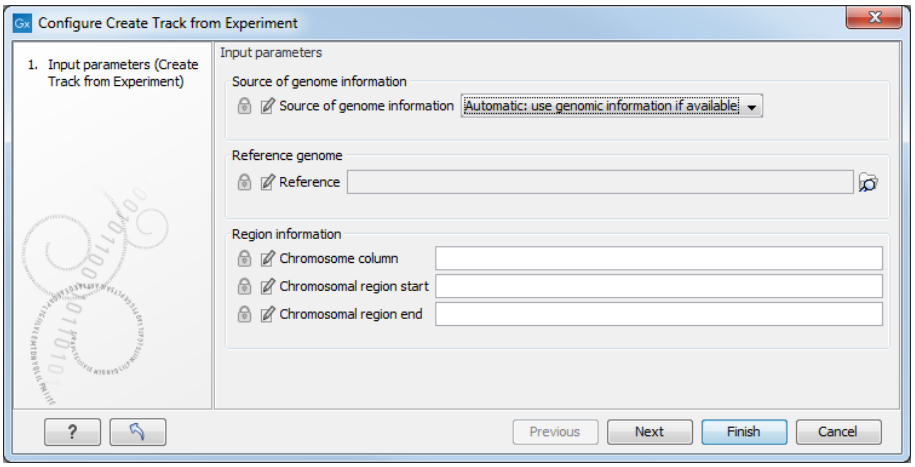


Figure 35.9: *Setting the parameters for the Create Track from Experiment tool in a workflow.*

The **Create Track from Experiment** tool will produce a track or several tracks, if filtering based on analysis results was chosen. The track(s) will contain the following annotations:

- All experiment-specific columns from the experiment
- All user-defined annotations added to the experiment
- All analysis-specific columns from the experiment
- All group-specific columns from the experiment
- Those of the following sample-specific columns when present in the experiment (for each sample): Expression values, Total exon reads, and RPKM.

Two different view options exist: the Track List and the Table View. When opening the annotated output result, the default view is the Track List. It is possible to open both views in split view by

holding down the Ctrl key while clicking on the table icon in the lower left corner of the View Area. The two different views are linked together. This means that when you click once on an entry in the table, the Track List will jump the selected region. With the **Zoom to Selection** ( ) button it is possible to jump to and zoom in on the selected region (figure 35.10).

The results of any statistical test executed on the experiment, including fold-changes and p-values, can be seen in the tooltip when hovering over each region in the annotation track shown in the Track List (figure 35.11).

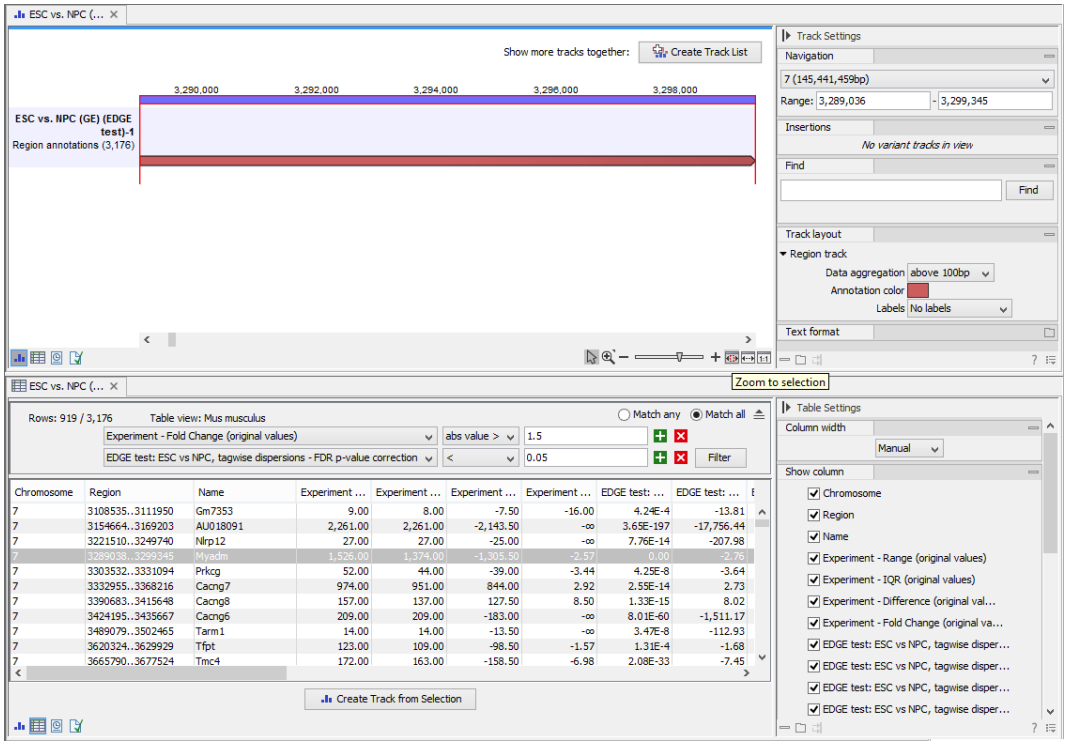


Figure 35.10: *Viewing the track produced by the Create Track from Experiment Tool*

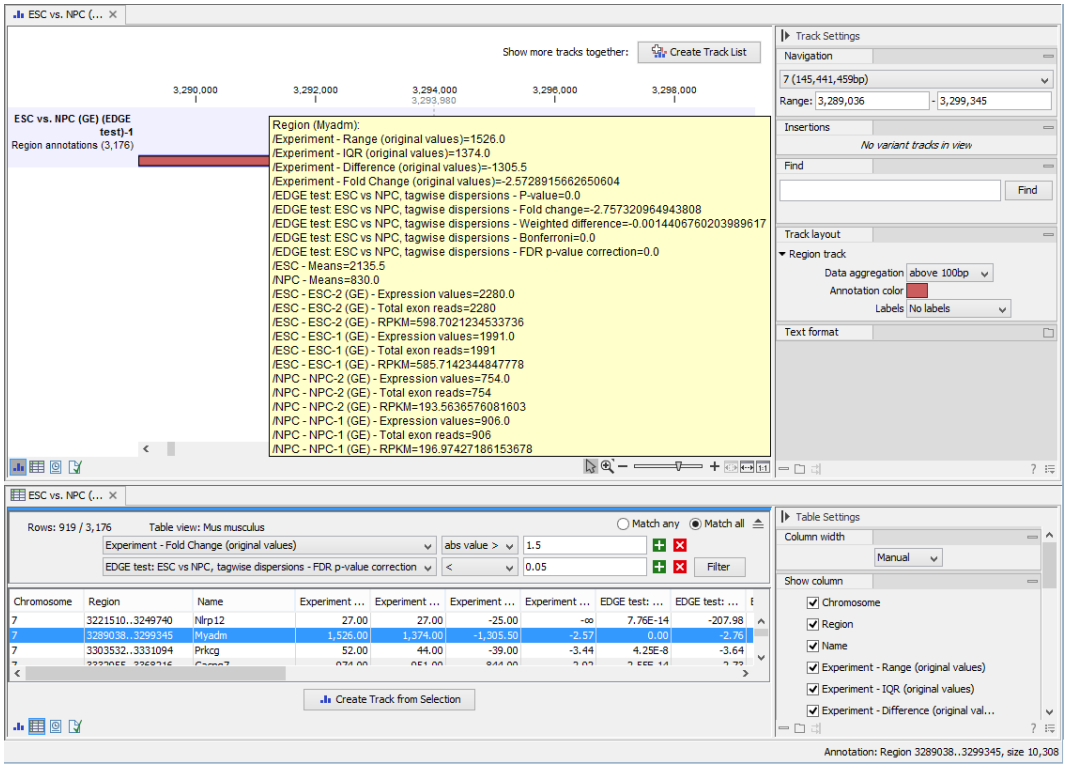


Figure 35.11: *The annotations on the track produced by the Create Track from Experiment Tool*

35.6 Small RNA Analysis

Tools for small RNA Analysis previously available under the **Microarray and Small RNA Analysis** folder of the Toolbox have been deprecated and will be retired in a future version of the software. Small RNA anaysis is now supported by new tools, available under the **RNA-Seq and Small RNA Analysis (**)| **miRNA Analysis** () folder of the Toolbox. Please see section 30.9 for information about the new tools.

The older small RNA analysis tools have been moved to **Legacy Tools** () | **Small RNA Analysis (legacy) (**) folder of the Toolbox, and each tool's name has "(legacy)" appended to it. If you have concerns about the future retirement of these tools, please contact QIAGEN Bioinformatics Support team at ts-bioinformatics@qiagen.com.

The small RNA analysis tools are designed to facilitate trimming of sequencing reads, counting and annotating of the resulting tags using miRBase or other annotation sources and performing expression analysis of the results. The tools are general and flexible enough to accommodate a variety of data sets and applications within small RNA profiling, including the counting and annotation of both microRNAs and other non-coding RNAs from any organism.

The annotation part is designed to make special use of the information in miRBase but more general references can be used as well.

There are generally two approaches to the analysis of microRNAs or other smallRNAs: (1) count the different types of small RNAs in the data and compare them to databases of microRNAs or other smallRNAs, or (2) map the small RNAs to an annotated reference genome and count the numbers of reads mapped to regions which have smallRNAs annotated. The approach taken by *CLC Genomics Workbench* is (1). This approach does not require an annotated genome for mapping - you can use the sequences in miRBase or any other sequence list of smallRNAs of

interest to annotate the small RNAs. In addition, small RNAs that would not have mapped to the genome (for example when lacking a high-quality reference genome, or if the RNAs have not been transcribed from the host genome) can still be measured and their expression be compared. The methods and tools developed for *CLC Genomics Workbench* are inspired by the findings and methods described in Creighton et al., 2009, Wyman et al., 2009, Morin et al., 2008 and Stark et al., 2010.

35.6.1 Extract and count

Extract and Count has been deprecated and will be retired in a future version of the software. Please use the new Quantify miRNA tool described in section 30.9.1 instead.

Extract and Count has been moved to the **Legacy Tools** ( ) folder of the Toolbox, and its name has "(legacy)" appended to it. If you have concerns about the future retirement of this tool, please contact QIAGEN Bioinformatics Support team at ts-bioinformatics@qiagen.com.

When using the Extract and Count tool, the first step in the analysis is to import the data (see section 6.3).

The next step is to extract and count the small RNAs to create a *small RNA sample* that can be used for further analysis (either annotating or analyzing using the expression analysis tools):

Legacy Tools | Small RNA Analysis (legacy) ( ) | Extract and Count ( )

This will open a dialog where you select the sequencing reads that you have imported. Click **Next** when the sequencing data is listed in the right-hand side of the dialog. Note that if you have several samples, they should be processed separately.

In the next dialog (figure 35.12), you specify whether the reads should be trimmed for adapter sequences prior to counting. It is often necessary to trim off remainders of adapter sequences from the reads before counting.

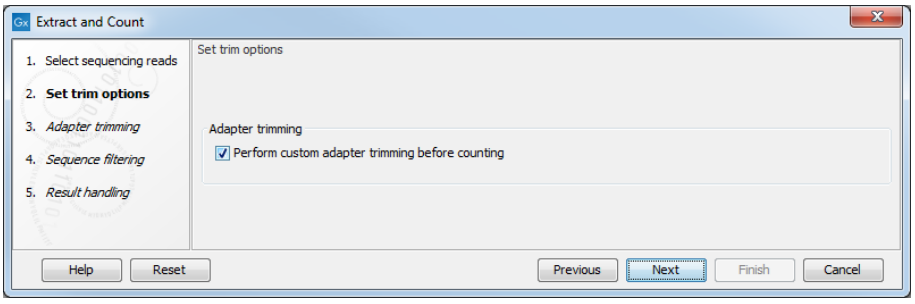


Figure 35.12: *Specifying whether adapter trimming is needed.*

When you click **Next**, you will be able to specify how the trim should be performed as shown in figure 35.13.

The trim options shown in figure 35.13 are the same as described under adapter trim in section 25.2.2. Please refer to this section for more information.

It should be noted that if you expect to see part of adapters in your reads, you would typically choose **Discard when not found** as the action. By doing this, only reads containing the adapter sequence will be counted as small RNAs in the further analysis. If you have a data set where the adapter may be there or not you would choose **Remove adapter**.

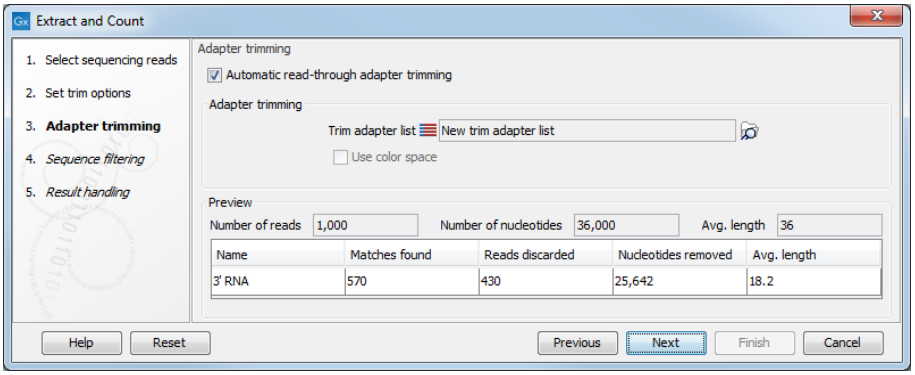


Figure 35.13: *Setting parameters for adapter trim.*

Note that all reads will be trimmed for ambiguity symbols such as N before the adapter trim.

If you have chosen not to trim the reads for adapter sequence, you will see figure 35.14 instead.

Clicking **Next** allows you to specify additional options regarding trimming and counting as shown in figure 35.14.

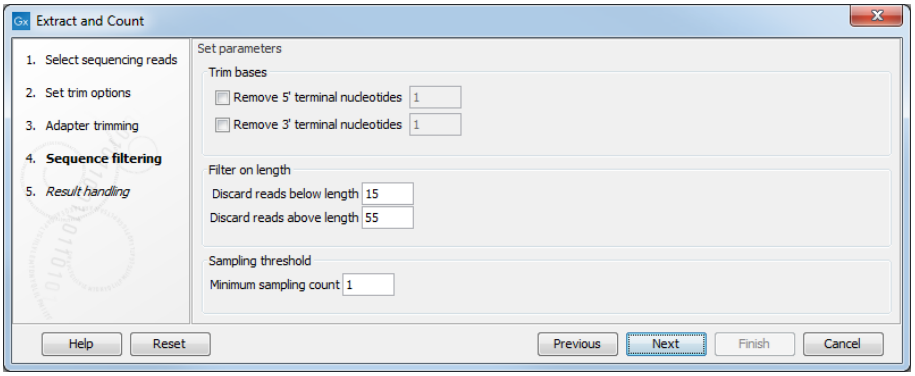


Figure 35.14: *Defining length interval and sampling threshold.*

At the top you can choose to **Trim bases** by specifying a number of bases to be removed from either the 3' or the 5' end of the reads. Below, you can specify the minimum and maximum lengths of the small RNAs to be counted (this is the length after trimming). The minimum length that can be set is 15 and the maximum is 55.

At the bottom, you can specify the **Minimum sampling count**. This is the number of copies of the small RNAs (tags) that are needed in order to include it in the resulting count table (the small RNA sample). The actual counting is very simple and relies on **perfect match** between the reads to be counted together. Note that you can identify variants of the same miRNA when annotating the sample (see below).

This also means that a count threshold of 1 will include a lot of unique tags as a result of sequencing errors. In order to set the threshold right, the following should be considered:

- If the sample is going to *be annotated*, annotations may be found for the tags resulting from sequencing errors. This means that there is no negative effect of including tags with a low count in the output.
- When using *un-annotated sequences* for discovery of novel small RNAs, it may be useful to apply a higher threshold to eliminate the noise from sequencing errors. However, this can

be done at a later stage by filtering the sample and creating a sub-set.

- When multiple samples are compared, it is interesting to know if one tag which is abundant in one sample is also found in another, even at a very low number. In this case, it is useful to include the tags with very low counts, since they may become more trustworthy in combination with information from other samples.
- Setting the count threshold higher will reduce the size of the sample produced which will reduce the memory and disk usage when working with the results.

Clicking **Next** allows you to specify the output of the analysis as shown in 35.15.

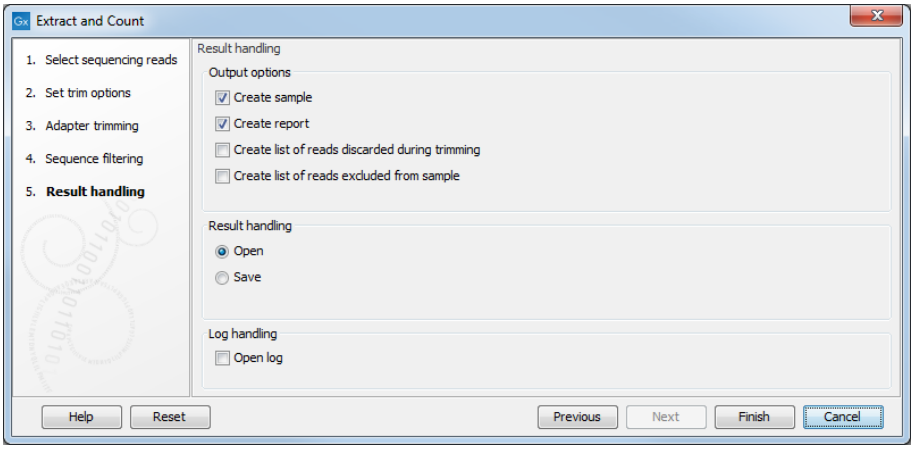


Figure 35.15: *Output options.*

The options are:

Create sample This is the primary result showing all the tags and respective counts (an example is shown in figure 35.16). Each row represents a tag with the actual sequence as the feature ID and columns with **Expression values**, **Length**, and **Count**. Expression values and the actual count are based on 100 % similarity[1]. The sample can be used in further analysis in the "raw" form, or you can annotate it (see below). The tools for working with the data in the sample are described in section 35.6.4. Note that when small RNA samples are used for setting up and experiment, it is always the Expression values that will be used.

Create report This will create a summary report as described below.

Create list of reads discarded during trimming This list contains the reads where no adapter was found (when choosing **Discard when not found** as the action).

Create list of reads excluded from sample This list contains the reads that passed the trimming but failed to meet the sampling thresholds regarding minimum/maximum length and number of copies.

The summary report includes the following information (an example is shown in figure 35.17):

Trim summary Shows the following information for each input file:

[1]Note that you can identify variants of the same miRNA when annotating the sample (see below).

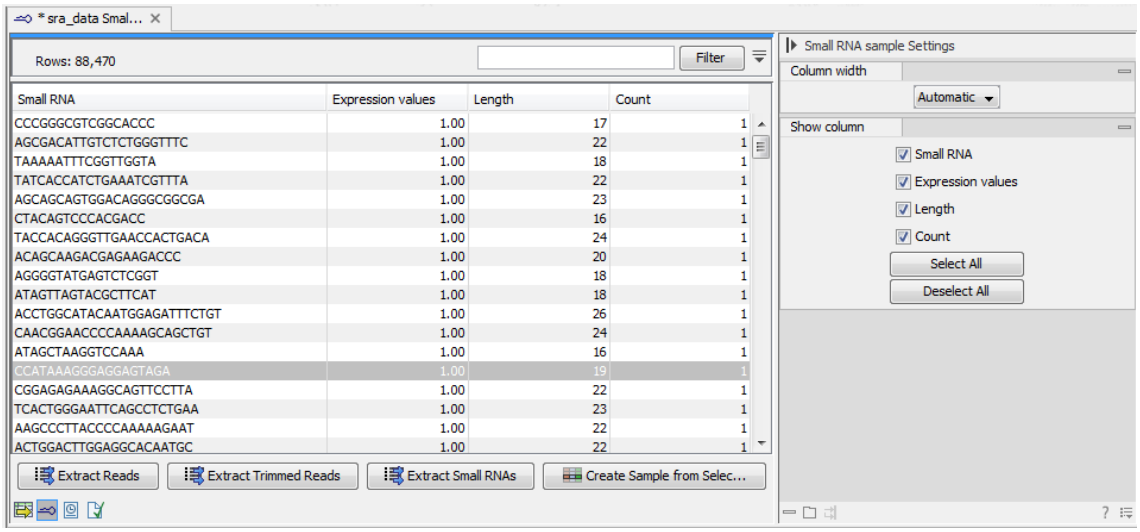


Figure 35.16: *The tags have been extracted and counted.*

- Number of reads in the input.
- Average length of the reads in the input.
- Number of reads after trim. The difference between the number of reads in the input and this number will be the number of reads that are discarded by the trim.
- Percentage of the reads that pass the trim.
- Average length after trim. When analyzing miRNAs, you would expect this number to be around 22. If the number is significantly lower or higher, it could indicate that the trim settings are not right. In this case, check that the trim sequence is correct, that the strand is right, and adjust the alignment scores. Sometimes it is preferable to increase the minimum scores to get rid of low-quality reads. The average length after trim could also be somewhat larger than 22 if your sequenced data contains a mixture of miRNA and other (longer) small RNAs.

Read length before/after trimming Shows the distribution of read lengths before and after trim. The graph shown in figure 35.17 is typical for miRNA sequencing where the read lengths after trim peaks at 22 bp.

Trim settings The trim settings summarized. Note that ambiguity characters will automatically be trimmed.

Detailed trim results This is described in the Trim output section 25.2.6.

Tag counts The number of tags and two plots showing on the x-axis the counts of tags and on the y-axis the number of tags for which this particular count is observed. The plot is in a zoomed version where only the lower part of the y-axis is shown to make it possible to see the numbers of tags higher counts.

35.6.2 Downloading miRBase

Download miRBase has been deprecated and will be retired in a future version of the software. The latest miRBase database can be downloaded via the Reference Data Manager, from under QIAGEN Sets | Reference Data Elements | mirBase, as described in section 30.9.1.

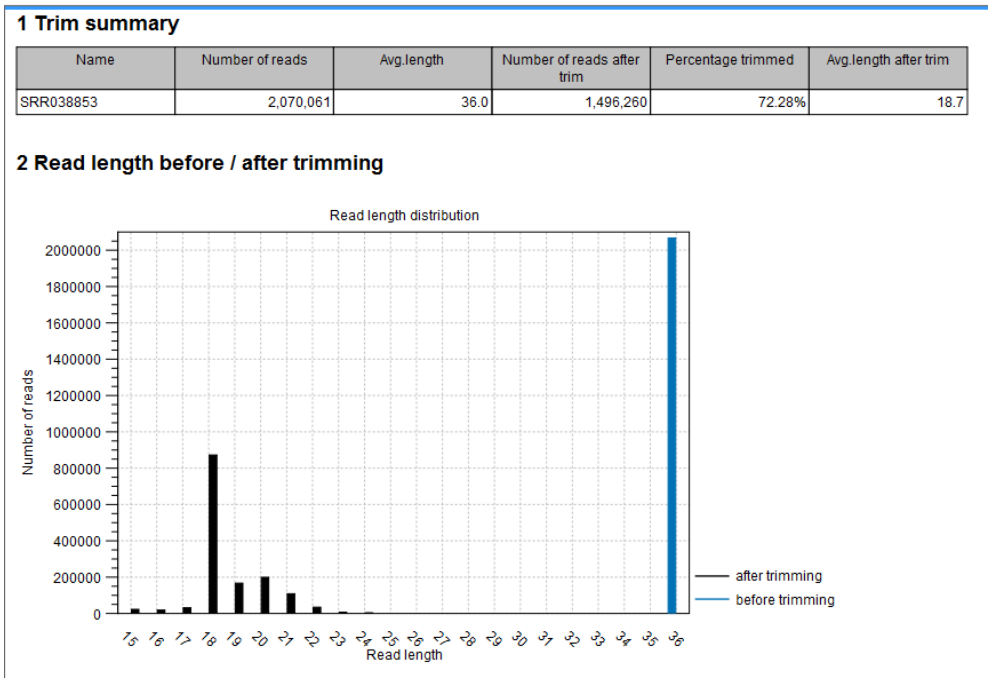


Figure 35.17: *A summary report of the counting.*

Download miRBase has been moved to the **Legacy Tools** ( ) folder of the Toolbox, and its name has "(legacy)" appended to it. If you have concerns about the future retirement of this tool, please contact QIAGEN Bioinformatics Support team at ts-bioinformatics@qiagen.com.

Previously, in order to make use of the additional information about mature regions on the precursor miRNAs in miRBase, you needed to use the integrated tool to download miRBase rather than downloading it from `http://www.mirbase.org/`

Legacy Tools | **Small RNA Analysis** ( ) | **Download miRBase** ( )

This will download a sequence list with all the precursor miRNAs including annotations for mature regions. The list can then be selected when annotating the samples with miRBase (see section 35.6.3).

The downloaded version will always be the latest version (it is downloaded from `ftp://mirbase.org/pub/mirbase/CURRENT/miRNA.dat.gz`). Information on the version number of miRBase is also available in the **History** ( ) of the downloaded sequence list, and when using this for annotation, the annotated samples will also include this information in their **History** ( ).

Importing the miRBase data file You can also import the miRBase data file directly into the Workbench. The file can be downloaded from `ftp://mirbase.org/pub/mirbase/CURRENT/miRNA.dat.gz`.

For the file to be recognized as a miRBase file, you have to 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 I.1.7.

35.6.3 Annotating and merging small RNA samples

Annotate and Merge Counts has been deprecated and will be retired in a future version of the software. Small RNA anaysis is now supported by new tools, available under the **RNA-Seq and Small RNA Analysis** | **miRNA Analysis** folder of the Toolbox. Please see section 30.9 for information on the new tools.

Annotate and Merge Counts has been moved to the **Legacy Tools** ( ) folder of the Toolbox, and its name has "(legacy)" appended to it. If you have concerns about the future retirement of this tool, please contact QIAGEN Bioinformatics Support team at ts-bioinformatics@qiagen.com.

The small RNA sample produced when counting the tags (see section 35.6.1) can be enriched by *CLC Genomics Workbench* by comparing the tag sequences with annotation resources such as miRBase and other small RNA annotation sources. Note that the annotation can also be performed on an experiment, set up from small RNA samples (see section 31.1.1).

Besides adding annotations to known small RNAs in the sample, it is also possible to merge variants of the same small RNA to get a cumulative count. When initially counting the tags, the Workbench requires that the trimmed reads are identical for them to be counted as the same tag. However, you will often see different variants of the same miRNA in a sample, and it is useful to be able to count these together. This is also possible using the tool to annotate and merge samples.

Legacy Tools | **Small RNA Analysis ( )** | **Annotate and Merge Counts ( )**

This will open a dialog where you select the small RNA samples ( ) to be annotated. Note that if you have included several samples, they will be processed separately but summarized in one report providing a good overview of all samples. You can also input **Experiments** ( ) (see section 31.1.1) created from small RNA samples. Click **Next** when the data is listed in the right-hand side of the dialog.

This dialog (figure 35.18) is where you define the annotation resources to be used.

F

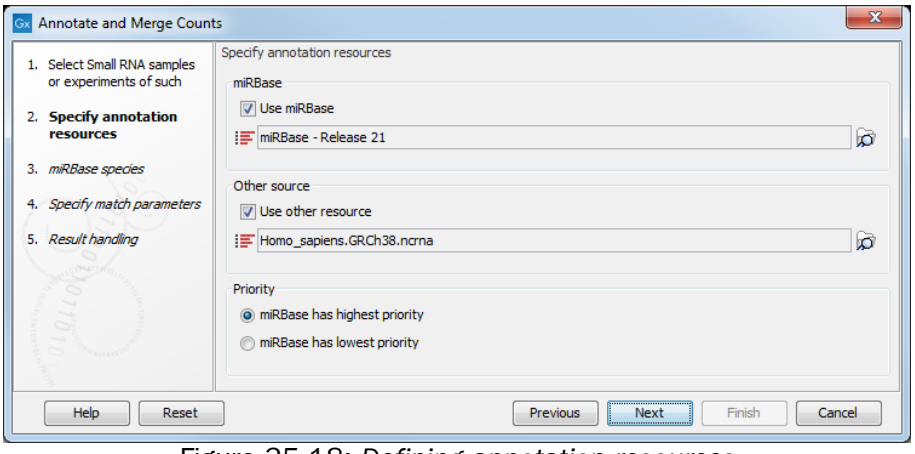


Figure 35.18: *Defining annotation resources.*

There are two ways of providing annotation sources:

- Downloading miRBase using the integrated download tool (explained in section 35.6.2).
- Importing a list of sequences from a fasta file. This could be from Ensembl: `ftp://ftp.`

ensembl.org/pub/release-57/fasta/homo_sapiens/ncrna/Homo_sapiens.GRCh37.57.ncrna.fa.gz

or from ncRNA.org: http://togodb.biosciencedbc.jp/togodb/view/frnadb_summary.

We recommend using the integrated download tool to import miRBase. Although it is possible to import it as a fasta file, the same options with regards to species will not be available if you import from a file.

The downloaded miRBase file contains all precursor sequences from the latest version of miRBase http://www.mirbase.org/ including annotations defining the mature regions (see an example in figure 35.19).

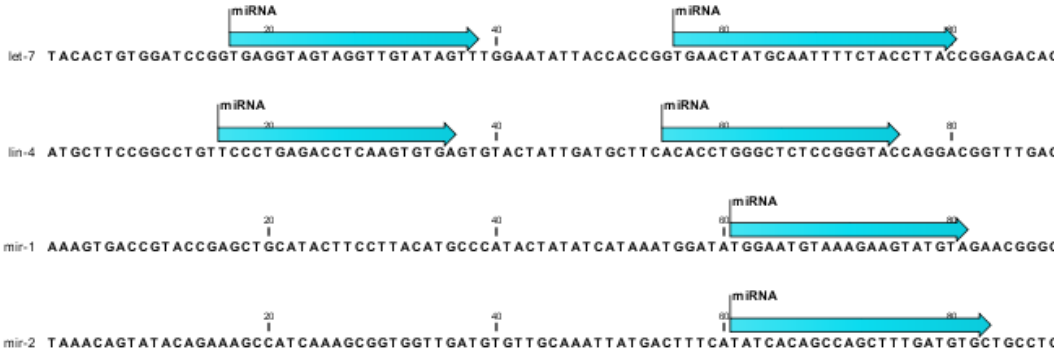


Figure 35.19: *Some of the precursor miRNAs from miRBase have both 3' and 5' mature regions annotated (as the two first in this list).*

This means that it is possible to have a more fine-grained classification of the tags using miRBase compared to a simple fasta file resource containing the full precursor sequence. This is the reason why the miRBase annotation source is specified separately in figure 35.18.

At the bottom of the dialog, you can specify whether miRBase should be prioritized over the additional annotation resource. The prioritization is explained in detail later in this section. To prioritize one over the other can be useful when there is redundant information (e.g. if you have an additional source that also contains all the miRNAs from miRBase and you prefer the miRBase annotations when possible).

When you click **Next**, you will be able to choose which species from miRBase should be used and in which order (see figure 35.20). Note that if you have not selected a miRBase annotation source, you will go directly to the next step shown in figure 35.21.

To the left, you see the list of species in miRBase. This list is dynamically created based on the information in the miRBase file. Using the arrow button ( ) you can add species to the right-hand panel. The order of the species is important since the tags are annotated iteratively based on the order specified here. This means that in the example in figure 35.20, a human miRNA will be preferred over mouse, even if they are identical in sequence (the prioritization is elaborated below). The up and down arrows ( )/ ( ) can be used to change the order of species.

When you click **Next**, you will be able to specify how the alignment of the tags against the annotation sources should be performed (see figure 35.21).

The panel at the top is active only if you have chosen to annotate with miRBase. It is used to

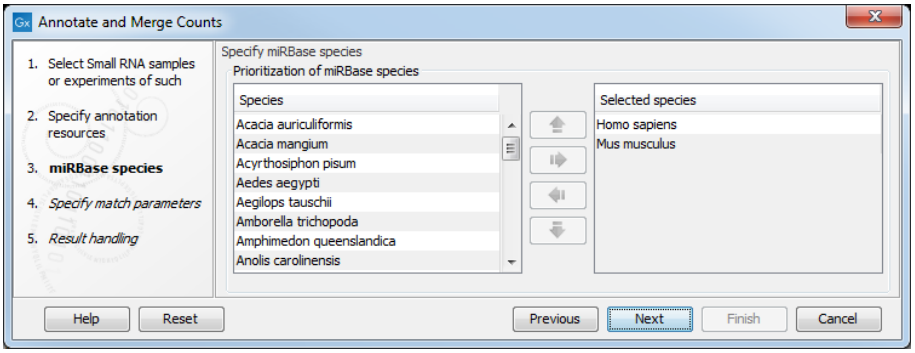


Figure 35.20: *Defining and prioritizing species in miRBase.*

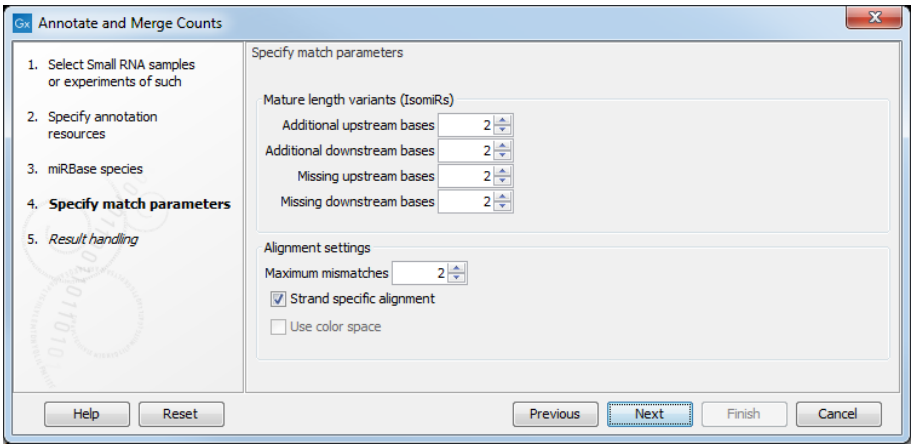


Figure 35.21: *Setting parameters for aligning.*

define the requirements to the alignment of a read for it to be counted as a 3' or 5' mature region tag:

Additional upstream bases This defines how many bases the tag is allowed to extend the annotated mature region at the 5' end and still be categorized as mature.

Additional downstream bases This defines how many bases the tag is allowed to extend the annotated mature region at the 3' end and still be categorized as mature.

Missing upstream bases This defines how many bases the tag is allowed to miss at the 5' end compared to the annotated mature region and still be categorized as mature.

Missing downstream bases This defines how many bases the tag is allowed to miss at the 3' end compared to the annotated mature region and still be categorized as mature.

At the bottom of the dialog you can specify the **Maximum mismatches** (default value is 2). Finally, you can choose whether the tags should be aligned against both strands of the reference or only the positive strand. Usually it is only necessary to align against the positive strand.

At this point, a more elaborate explanation of the annotation algorithm is needed. The short read mapping algorithm in the *CLC Genomics Workbench* is used to map all the tags to the reference sequences which comprise the full precursor sequences from miRBase and the sequence lists chosen as additional resources. The mapping is done in several rounds: the first round is done requiring a perfect match, the second allowing one mismatch, the third allowing two mismatches

etc. No gaps are allowed. The number of rounds depend on the number of mismatches allowed (by default 2, which means 3 rounds of read mapping as in figure 35.21).

After each round of mapping, the tags that are mapped will be removed from the list of tags that continue to the next round. This means that a tag mapping with perfect match in the first round will not be considered for the subsequent one-mismatch round of mapping.

Note: If there are mismatches in the read, there will be a limitation on how short reads can be mapped:

- A minimum read length of 17 nucleotides is required when the read contains one mismatch.
- A minimum read length of 21 nucleotides is required when the read contains two mismatches.
- A minimum read length of 25 nucleotides is required when the read contains three mismatches.

Following the mapping, the tags are classified into the following categories according to where they match.

- Mature 5': mature miRNA that is located closer (or equally close) to the 5' end than to the 3' end.
- Mature 3': mature miRNA that is located closer to the 3' end.
- exact: the tag matches exactly to the annotated mature 3' region.
- super: the observed tag is longer than the annotated mature region.
- sub: the observed tag is shorter than the annotated mature region.
- sub/super: the observed tag extends the annotation in one end and is shorter at the other end.
- Precursor: the tag matches on a miRBase sequence, but not within the extended annotated mature region(s). These are defined by the "mature length variants (IsomiRs)" parameters in the "specify match parameters" wizard step (by default these parameters are set to 2 which means that reads that are at most 2 bases too long or too short relative to the annotated mature region are all considered mature hits).
- Other: for hits in the other resources (the information about resource is also shown in the output).

All these categories except *Other* refer to hits in miRBase. The miRBase sequences may have up to two mature micro RNAs annotated.

An example of an alignment is shown in figure 35.22 using the same alignment settings as in figure 35.21.

The two tags at the top are both classified as *mature 5' super* because they cover and extend beyond the annotated mature 5' RNA. The third tag is identical to the annotated mature 5'.

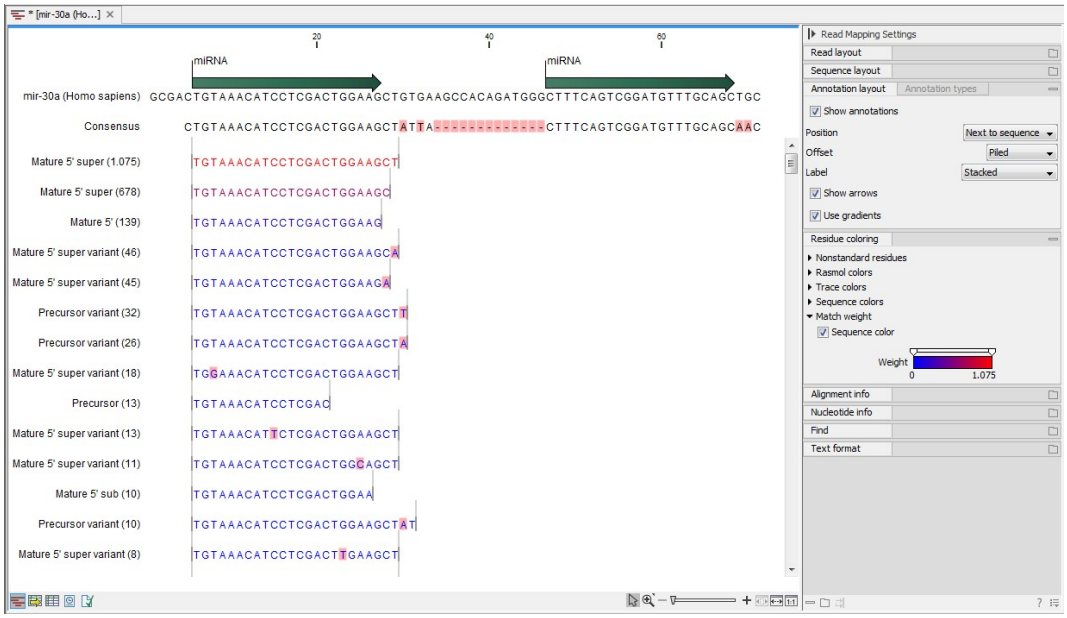


Figure 35.22: *Alignment of length variants of mir-30a.*

If a tag has several hits, the list above is used for prioritization. This means that for example, a *Mature 5' sub* is preferred over a *Mature 3' exact*. Note that if miRBase was chosen as lowest priority (figure 35.18), the *Other* category will be at the top of the list. All tags mapping to a miRBase reference without qualifying to any of the mature 5' and mature 3' types will be typed as *Other*. Also note that if a tag has several hits to references *with the same priority* (when the tag matches the mature regions of two different miRBase sequences) it will be annotated with all these sequences. In the report we refer to these tags as 'ambiguously annotated'.

In case you have selected more than one species for miRBase annotation (e.g., Homo Sapiens and Mus Musculus), adding annotations follows these rules:

1. If a tag has hits with the same priority for both species, the annotation for the top-prioritized species will be added.
2. Read category priority is stronger than species category priority: If a read is a higher priority match for a mouse miRBase sequence than it is for a human miRBase sequence the annotation for the mouse will be used.

Clicking **Next** allows you to specify the output of the analysis as shown in 35.23.

Create unannotated sample All the tags where no hit was found in the annotation source are included in the unannotated sample. This sample can be used for investigating novel miRNAs, see section 35.6.5. No extra information is added, so this is just a subset of the input sample.

Create annotated sample This will create a sample as described in section 35.6.4. In this sample, the following columns have been added to the counts.

Name This is the name of the annotation sequence in the annotation source. For miRBase, it will be the names of the miRNAs (e.g. *let-7g* or *mir-147*), and for other source, it will be the name of the sequence.

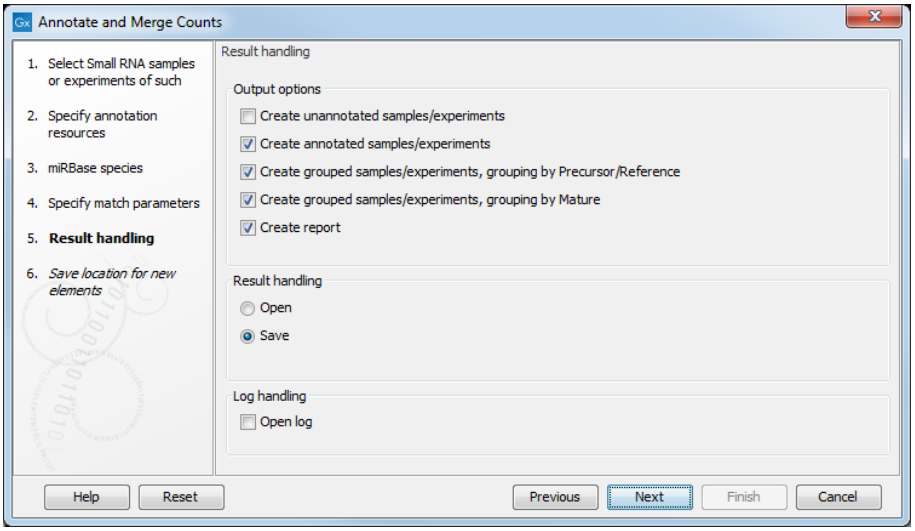


Figure 35.23: *Output options.*

Resource This is the source of the annotation, either miRBase (in which case the species name will be shown) or other sources (e.g. Homo_sapiens.GRCh37.57.ncrna).

Match type The match type can be exact or variant (with mismatches) of the following types:

- Mature 5'
- Mature 5' super
- Mature 5' sub
- Mature 5' sub/super
- Mature 3'
- Mature 3' super
- Mature 3' sub
- Mature 3' sub/super
- Other

Mismatches The number of mismatches.

Note that if a tag has two equally prioritized hits, they will be shown with // between the names. This could be for example two precursor sequences sharing the same mature sequence (also see the sample grouped on mature below).

Create grouped sample, grouping by Precursor/Reference This will create a sample as described in section 35.6.4. All variants of the same reference sequence will be merged to create one expression value for all.

Expression values. The expression value can be changed at the bottom of the table. The default is to use the counts in the mature 5' column.

Name. The name of the reference. For miRBase this will then be the name of the precursor.

Resource. The name of the resource that the reference comes from.

Exact mature 5'. The number of exact mature 5' reads.

Mature 5'. The number of all mature 5' reads including sub, super and variants.

Unique exact mature 5'. In cases where one tag has several hits (as denoted by the // in the ungrouped annotated sample as described above), the counts are distributed evenly across the references. The difference between *Exact mature 5'* and *Unique exact mature 5'* is that the latter only includes reads that are unique to this reference.

Unique mature 5'. Same as above but for all mature 5's, including sub, super and variants.

Exact mature 3'. Same as above, but for mature 3'.

Mature 3'. Same as above, but for mature 3'.

Unique exact mature '3. Same as above, but for mature 3'.

Unique mature '3. Same as above, but for mature 3'.

Exact other. Exact matches in miRBase sequences, but outside annotated mature regions.

Other. All matches in miRBase sequences, but outside annotated mature regions, including variants.

Total. The total number of tags mapped and classified to the precursor/reference sequence.

Note that, for non-miRBase sequences, the counts are collected in the 'Mature 5'' columns: 'Exact mature 5'' (number reads that map to the sequence without mismatches), 'Mature 5' (number reads that map to the sequence, including those with mismatches), 'Unique exact mature 5' (number reads that map uniquely to the sequence without mismatches) and 'Unique mature 5' (number reads that map uniquely to the sequence, including those with mismatches).

Create grouped sample, grouping by Mature This will create a sample as described in section 35.6.4. This is also a grouped sample, but in addition to grouping based on the same reference sequence, the tags in this sample are grouped on the same mature 5'. This means that two precursor variants of the same mature 5' miRNA are merged. For precursor miRNAs that have both a 5' and a 3', we group on both 5' mature and 3' mature. For these miRNAs you will see two rows in the grouped on mature output table, one with the 5' mature sequence as feature ID and one with the 3' mature sequence as feature ID. The 'Match type' column indicates whether the feature ID is 5' or 3' mature. Note that it is only possible to create this sample when using miRBase as annotation resource (because the Workbench has a special interpretation of the miRBase annotations for mature as described previously). To find identical mature 5' miRNAs, the Workbench compares all the mature 5' sequences and when they are identical, they are merged. The names of the precursor sequences merged are all shown in the table.

Expression values. The expression value can be changed at the bottom of the table. The default is to use the counts in the mature 5' column.

Name. The name of the reference. When several precursor sequences have been merged, all the names will be shown separated by //.

Resource. The species of the reference.

Exact mature 5'. The number of exact mature 5' reads.

Mature 5'. The number of all mature 5' reads including sub, super and variants.

Unique exact mature 5'. In cases where one tag has several hits (as denoted by the // in the ungrouped annotated sample as described above), the counts are distributed evenly across the references. The difference between *Exact mature 5'* and *Unique exact mature 5'* is that the latter only includes reads that are unique to one of the precursor sequences that are represented under this mature 5' sequence.

Unique mature 5'. Same as above but for all mature 5's, including sub, super and variants.

Create report The summary report includes the following information (an example is shown in figure 35.24):

1 Summary

Name	Small RNAs	Annotated	Percentage	Ambiguously annotated	Percentage	Reads	Annotated	Percentage	Ambiguously annotated	Percentage
sra_data Small RNA sample	88.470	32.125	36,3%	9.788	11,1%	1.720.280	1.510.721	87,8%	971.902	56,5%

2 Resources

Resource	Sequences in resource	Sequences found	Percentage found
miRBase (Homo sapiens)	1.600	599	37,4%
miRBase (Mus musculus)	855	84	9,8%
Homo_sapiens.GRCh37.57.ncrna	12.887	3.655	28,4%

3 Reads

Annotation	Count	Percentage	Perfect matches	%	1 mismatch	%	2 mismatches	%
Annotated	1.510.721	87,8%	1.212.258	80,2%	247.484	16,4%	50.979	3,4%
- with miRBase	1.467.902	97,2%	1.188.128	80,9%	234.583	16,0%	45.191	3,1%
- Homo sapiens	1.456.045	99,2%	1.182.700	81,2%	230.445	15,8%	42.900	2,9%
- Mus musculus	11.857	0,8%	5.428	45,8%	4.138	34,9%	2.291	19,3%
- with Homo_sapiens. GRCh37.57. ncrna	42.819	2,8%	24.130	56,4%	12.901	30,1%	5.788	13,5%
Unannotated	209.559	12,2%						
Total	1.720.280	100,0%						

4 Small RNAs

Annotation	Count	Percentage
Annotated	32.125	36,3%
- with miRBase	21.490	66,9%
- Homo sapiens	20.259	94,3%
- Mus musculus	1.231	5,7%
- with Homo_sapiens.GRCh37.57.ncrna	10.635	33,1%
Unannotated	56.345	63,7%
Total	88.470	100,0%

Figure 35.24: *A summary report of the annotation.*

Summary Shows the following information for each input sample:

- Number of small RNAs(tags) in the input.

- Number of annotated tags (number and percentage).
- Number of ambiguously annotated tags (number and percentage).
- Number of reads in the sample (one tag can represent several reads)
- Number of annotated reads (number and percentage).
- Number of ambiguously annotated reads (number and percentage).

Resources Shows how many matches were found in each resource:

- Number of sequences in the resource.
- Number of sequences where a match was found (i.e. this sequence has been observed at least once in the sequencing data).

Reads Shows the number of reads that fall into different categories (there is one table per input sample). On the left hand side are the annotation resources. For each resource, the count and percentage of reads in that category are shown. Note that the percentage are relative to the overall categories (e.g. the miRBase reads are a percentage of all the *annotated* reads, not all reads). This is information is shown for each mismatch level.

Small RNAs Similar numbers as for the reads but this time for each small RNA tag and without mismatch differentiation.

Read count proportions A histogram showing, for each interval of read counts, the proportion of annotated (respectively, unannotated) small RNAs with a read count in that interval. Annotated small RNAs may be expected to be associated with higher counts, since the most abundant small RNAs are likely to be known already.

Annotations (miRBase) Shows an overview table for classifications of the number of reads that fall in the miRBase categories for each species selected.

Annotations (Other) Shows an overview table with read numbers for total, exact match and mutant variants for each of the other annotation resources.

An example of the results table is shown at the bottom of figure 35.25.

In the upper part of the figure, you can see two tables showing the imported miRBase file (opened twice for illustration purpose). This example has been included to illustrate the logic behind the order in which the resources are listed. If you look at the results table found at the bottom of the figure, you will see that in the **Resource** column, Mus musculus is listed before Homo sapiens in the first row and in the second row Homo sapiens is listed before Mus musculus. The rationale behind this can be seen from the two other tables if you look at the accession numbers. If you filter only to include Homo sapiens and Mus musculus RNAs, and sort on the "name" column, you can see that for mir-130b (Mus musculus/Homo sapiens), the miRBase number of the mouse RNA is smaller than the miRBase number of the human RNA. Conversely, for mir-495 (Homo sapiens/Mus musculus), the number is smaller in the human case.

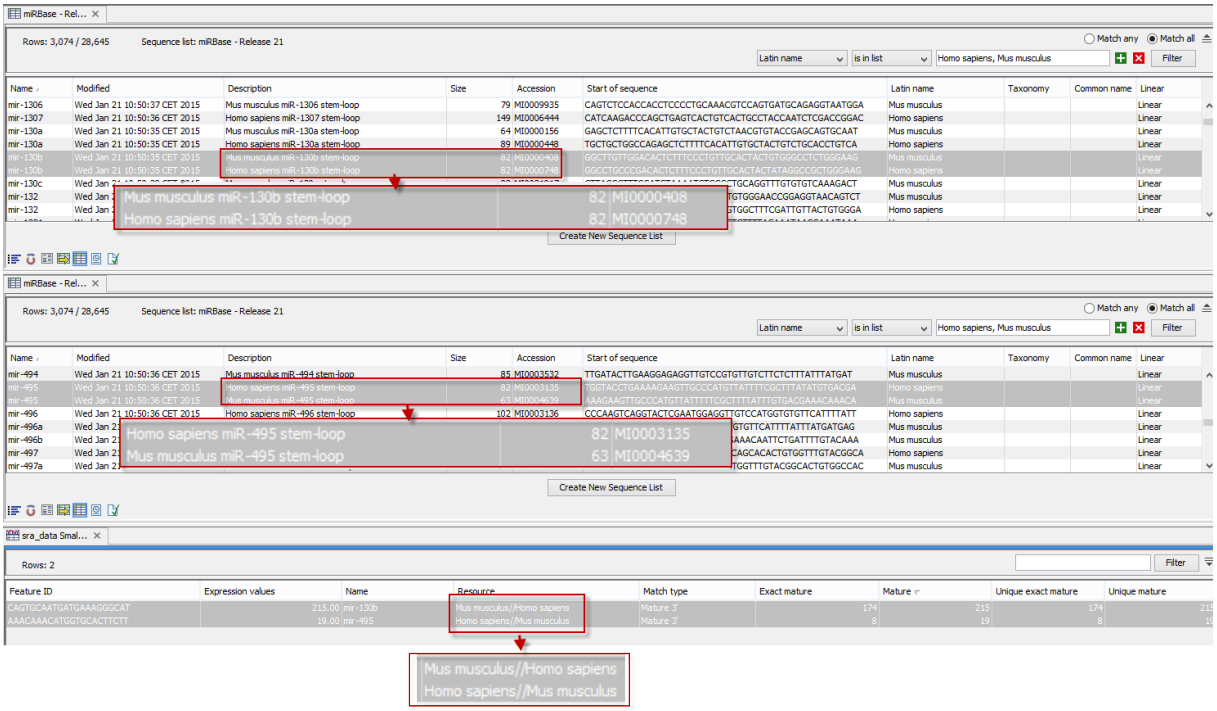


Figure 35.25: *The results table (at the bottom of the figure) shown in split view with the imported miRBase data shown twice to illustrate that the order in which the resource is listed is determined by the miRBase accession number. Due to the very small fonts in the table, the most important information has been magnified (red box).*

35.6.4 Working with the small RNA sample

Generally speaking, the small RNA sample comes in two variants:

- The *un-grouped* sample, either as it comes directly from the **Extract and Count** ( ) or when it has been annotated. In this sample, there is one row per tag, and the feature ID is the tag sequence.
- The *grouped* sample created using the **Annotate and Merge Counts** ( ) tool. In this sample, each row represents several tags grouped by a common Mature or Precursor miRNA or other reference.

Below, these two kinds of samples are described in further detail. Note that for both samples, filtering and sorting can be applied, see section 3.2.

The ungrouped sample

An example of an ungrouped annotated sample is shown in figure 35.26.

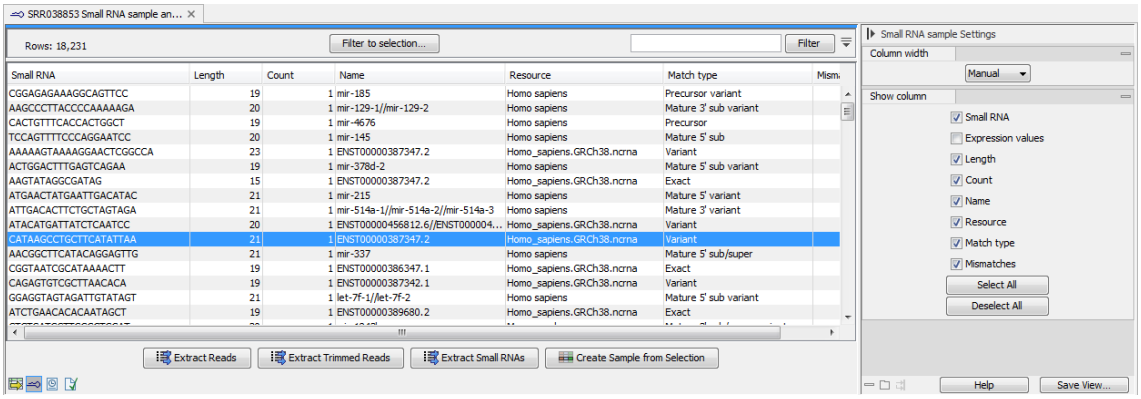


Figure 35.26: *An ungrouped annotated sample.*

By selecting one or more rows in the table, the buttons at the bottom of the view can be used to extract sequences from the table:

Extract Reads ( ) This will extract the original sequencing reads that contributed to this tag. Figure 35.27 shows an example of such a read. The reads include trim annotations (for use when inspecting and double-checking the results of trimming). Note that if these reads are used for read mapping, the trimmed part of the read will automatically be removed. If all rows in the sample are selected and extracted, the sequence list would be the same as the input except for the reads that did not meet the adapter trim settings and the sampling thresholds (tag length and number of copies).

Extract Trimmed Reads ( ) The same as above, except that the trimmed part has been removed.

Extract Small RNAs ( ) This will extract only one copy of each tag.

Note that for all these, you will be able to determine whether a list of DNA or RNA sequences should be produced (when working within the *CLC Genomics Workbench* environment, this only effects the RNA folding tools).

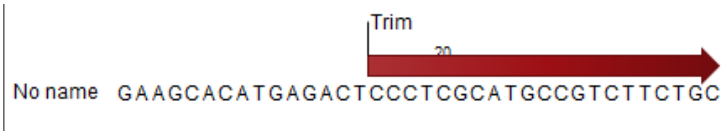


Figure 35.27: *Extracting reads from a sample.*

The button **Create Sample from Selection** ( ) can be used to create a new sample based on the tags that are selected. This can be useful in combination with filtering and sorting.

The grouped sample

An example of a grouped annotated sample is shown in figure 35.28.

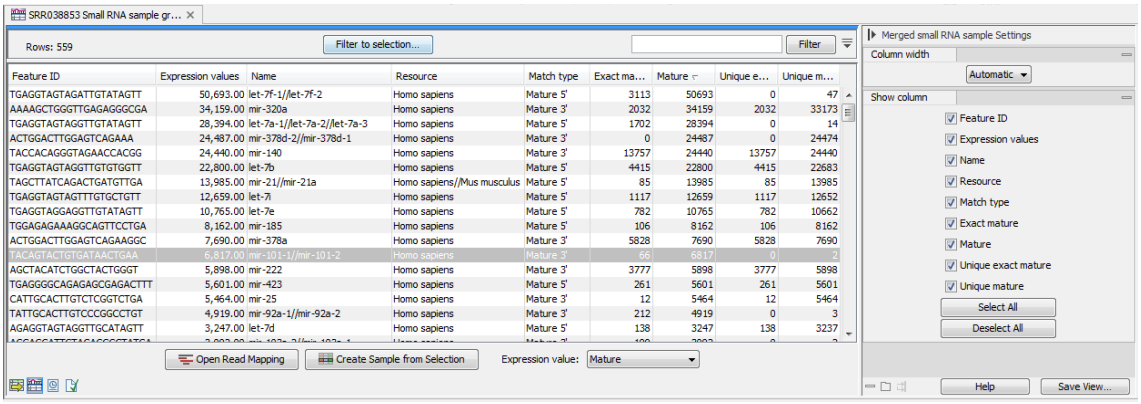


Figure 35.28: *A sample grouped on mature 5' miRNAs.*

The contents of the table are explained in section 35.6.3. In this section, we focus on the tools available for working with the sample.

By selecting one or more rows in the table, the buttons at the bottom of the view become active:

Open Read Mapping ( ) This will open a view showing the annotation reference sequence at the top and the tags aligned to it as shown in figure 35.29. The names of the tags indicate their status compared with the reference (e.g. Mature 5', Mature super 5', Precursor). This categorization is based on the choices you make when annotating. You can also see the annotations when using miRBase as the annotation source. In this example both the mature 5' and the mature 3' are annotated, and you can see that both are found in the sample. In the **Side Panel** to the right you can see the **Match weight** group under **Residue coloring** which is used to color the tags according to their relative abundance. The weight is also shown next to the name of the tag. The left side color is used for tags with low counts and the right side color is used for tags with high counts, relative to the total counts of this annotation reference. The sliders just above the gradient color box can be dragged to highlight relevant levels of abundance. The colors can be changed by clicking the box. This will show a list of gradients to choose from.

Create Sample from Selection ( ) This is used to create a new sample based on the tags that are selected. This can be useful in combination with filtering and sorting.

35.6.5 Exploring novel miRNAs

One way of doing this would be to identify interesting tags based on their counts (typically you would be interested in pursuing tags with not too low counts in order to avoid wasting efforts on tags based on reads with sequencing errors), **Extract Small RNAs** ( ) and use this list of tags as input to **Map Reads to Reference** ( ) using the genome as reference. You could then examine where the reads match, and for reads that map in otherwise unannotated regions you could select a region around the match and create a subsequence from this. The subsequence could be folded and examined to see whether the secondary structure was in agreement with the expected hairpin-type structure for miRNAs.

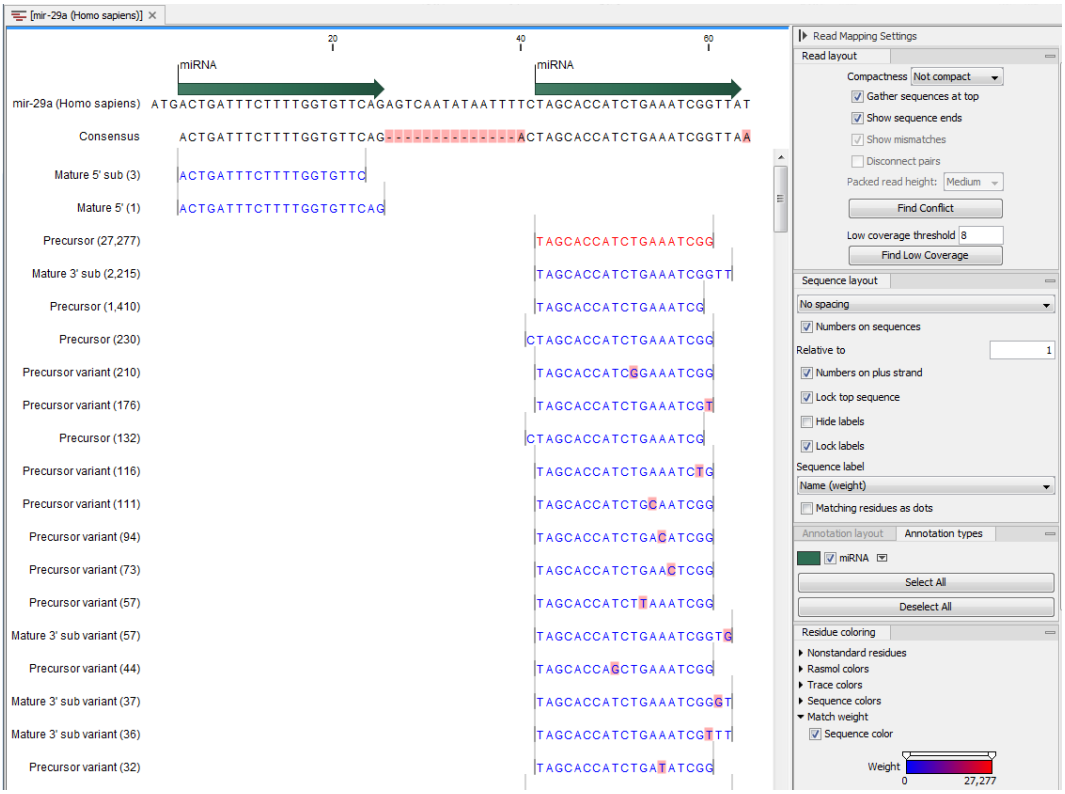


Figure 35.29: *Aligning all the variants of this miRNA from miRBase, providing a visual overview of the distribution of tags along the precursor sequence.*

35.7 Batch launching workflows with multiple inputs

This section describes legacy functionality for launching workflows with multiple inputs, where the contents of more than one input should change in each batch. This functionality is now supported through standard launch mechanisms. Please see section **??** for information about the new functionality.

Previously, the requirements for launching workflows with multiple inputs in batch mode were:

- The workflow must be installed on the Workbench, meaning that the workflow is accessible from the Toolbox (as opposed to workflows accessible from the Navigation Area).
- The workflow is characterized by more than one input file, and all input elements are unique per batch. You cannot reuse a common input element (such as control reads for example), unless it has been saved under different names in the Navigation Area.
- An Excel format file (.xlsx/.xls) must be provided, with at least 3 different columns:
  - **Unique ID** The first column must contain either the exact name of the data elements to be used as inputs, or partial name information such that data elements being entered into the analysis can be uniquely identified and matched with the information contained in the spreadsheet.
  - **Grouping** A second column must specify which data elements should be analyzed together in a given batch unit: this would be the ID of a single individual when comparing different tissues from the same individual (one individual per batch); or a family name when identifying variants existing within one family (one family per batch).

– **Type** The third column must specify the type for each data element: the values in this column distinguish tissue samples from controls, or inform about the disease status of a family member (affected/non-affected/proband) when identifying disease causing variants.

(Figure 35.30) shows an example of a spreadsheet used in the case of tissue comparison. Note that the "grouping" and "type" are context specific, and will depend on the analysis performed, i.e., on the tools that constitute the workflow.

Unique ID = sample ID, exact of partial name of the reads file to ensure a unique match between reads and metadata.	Grouping = Identical values will be analyzed together in one batch unit, for example here Patient ID.	Type = value that defines which tissue is the control tissue and which is the sample tissue to be compared to the control.
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 35.30: *Example of a spreadhseet necessary to run a workflow in batch, where the workflow intend to compare two tissue samples.*

To launch a workflow with multiple input elements in batch mode, right click on the name of the workflow in the Toolbox and select the option "Run in Batch Mode..." (figure 35.31).

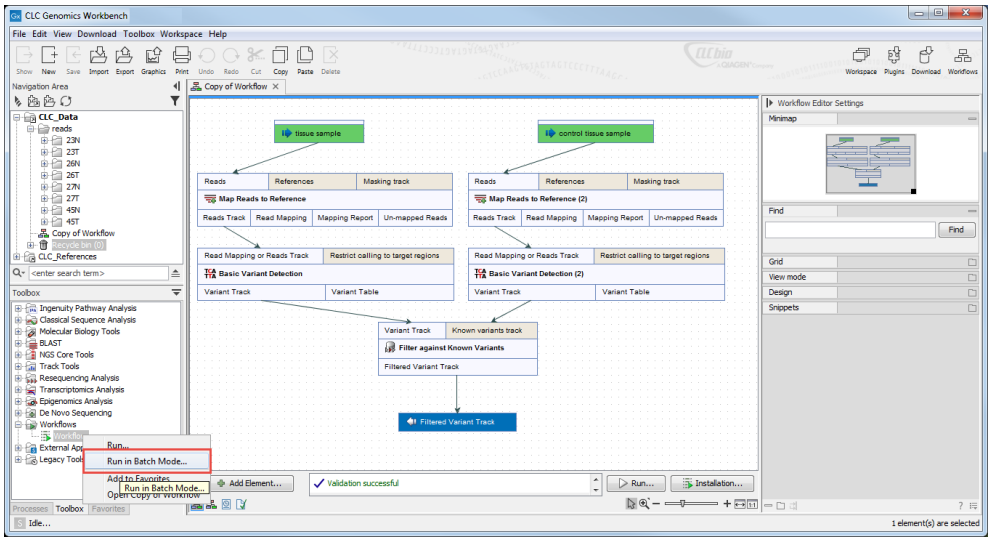


Figure 35.31: *The option to "Run in Batch Mode..." appears in the context menu when you right click on the name of an installed workflow that has multiple input elements in the Toolbox panel.*

A wizard opens and in the first window, you need to specify:

- An Excel file containing the information about the data to be analyzed (figure 35.32). Note that this file dos not need to be saved in your Navigation Area. When it has been selected, the table found in the lower part of the wizard will show recapitulate the content of the Excel sheet. The location of the data for this analysis is not yet specified, so a red, no-entry sign is visible in the header of the first column.
- The location of the reads: click on the Navigation button next to the "Location of data" field and specify the folder(s) that contain(s) the data, as shown in figure 35.33.

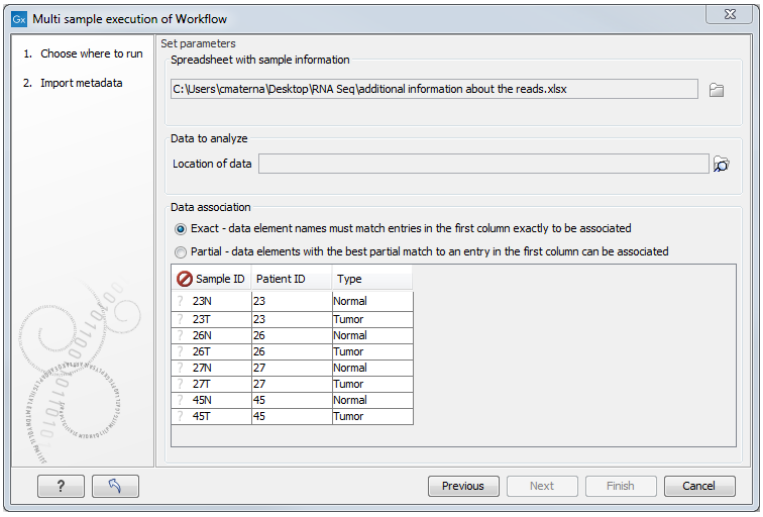


Figure 35.32: *Select the information about the data to be analyzed and the folder holding the data to analyze. An example of an Excel sheet with the relevant information is shown.*

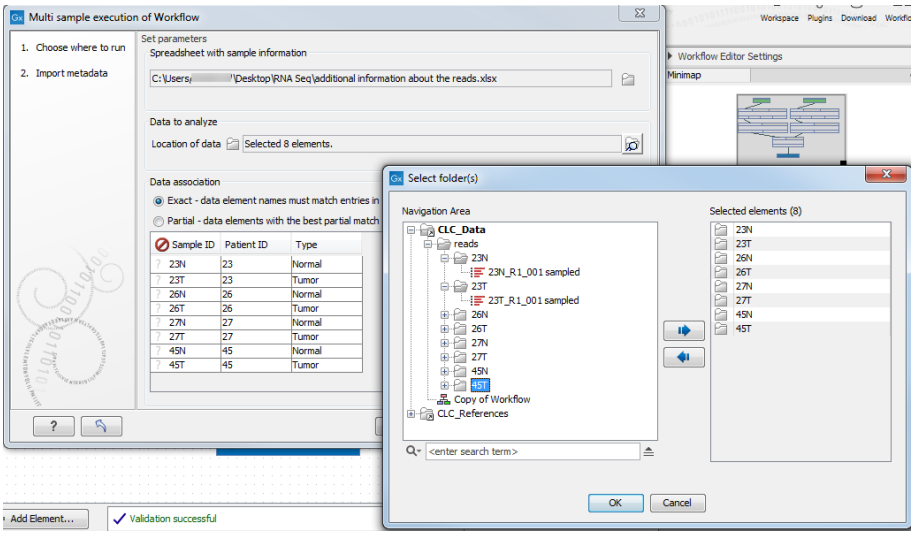


Figure 35.33: *Select the folder(s) that contain(s) the data to be analyzed.*

Data elements within the selected folders are considered for the analysis. Subfolders and their contents are not considered unless the subfolder is also selected. Individual data elements cannot be selected.

- Select the appropriate matching scheme - exact or partial. The matching rules applied are the same as those used for metadata association: "Exact" means that data element names must exactly match an entry in the first column of the Excel file; "Partial" matching allows for data elements names partially matching an entry in the first column. "Exact" is selected by default.

An icon with a green check mark (✓) appears in the table preview next to rows where a data element corresponding to a row of the Excel sheet was uniquely identified. If no match can be made to a given row of the Excel sheet, a question mark (?) is displayed.

Graphical symbols are also presented in the header of the first column of the preview pane to give information about the overall status of the matching of rows in the Excel sheet with data

elements in the Workbench:

- When no data elements match information in the Excel sheet, a red, no entry symbol ( ) is displayed. In this situation, the button labeled **Next** is not enabled. This is the expected state before any data elements have been selected.
- A yellow exclamation mark ( ) indicates that some, but not all rows in the Excel sheet have been matched to a data element in the selected folder(s).
- A green checkmark ( ) indicates that all rows in the Excel sheet have been matched to a data element in the selected folder(s).

In figure 35.34, the green check mark symbol in the header of the first column in the preview pane indicates that data elements were identified for each of the rows in the Excel sheet. You can click on the button labeled "Next".

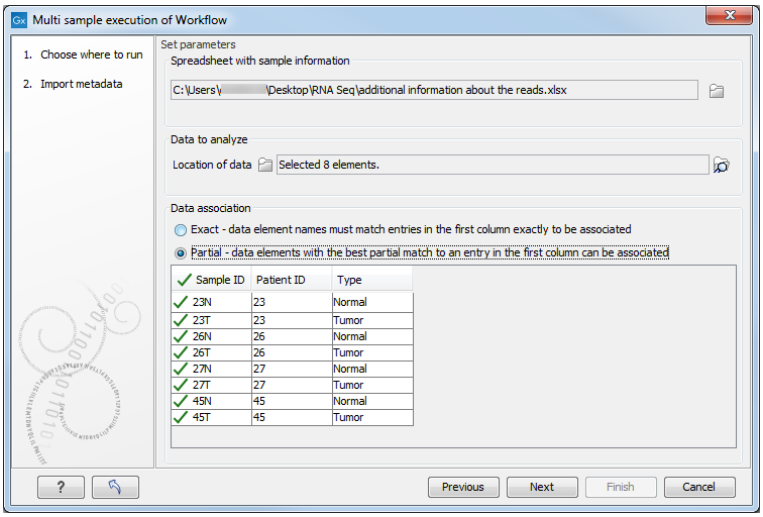


Figure 35.34: *View of the Data Association table after all samples were successfully associated.*

The next wizard window is called "Select grouping parameters and analysis inputs".

- In the **Group by** drop down menu, select the name of the column containing information that specifies which samples should be analyzed together.
- In the **Type** drop down menu, select the name of the column containing information that can be mapped to the workflow input type of each data element.

In the same window you will need to further specify the inputs of the workflow. What needs to be specified here is dependant on the workflow itself.

An example is shown in figure 35.35. **Group by** is set to a column specifying "Patient ID", because each workflow run will analyze a sample pair. **Type** is set to the "Type" column, because the workflow inputs are either tumor or normal tissues. The sample columns section maps data elements to the different workflow inputs, in this case "Tissue sample" is set to "Tumor", and "Control tissue sample" to "Normal".

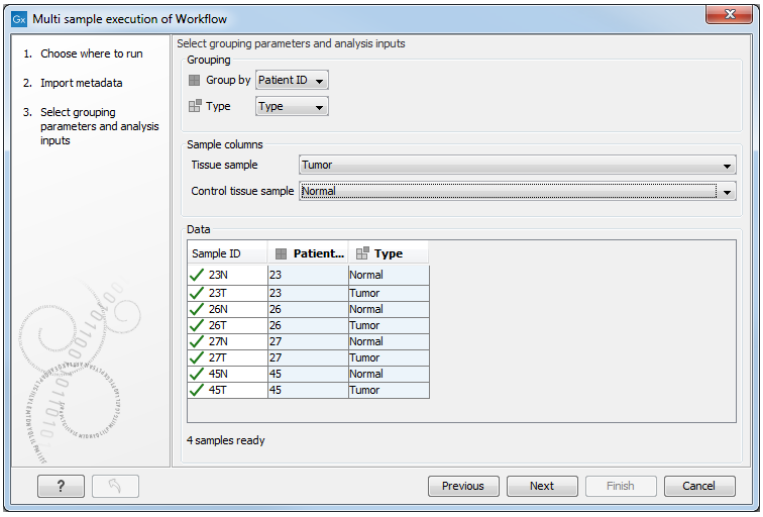


Figure 35.35: *Grouping samples.*

The rest of the wizard is dependant of the tools included in the workflow. Fill in the appropriate information and save the results of your workflow in a folder you can create in the Navigation Area.

As in a regular batching mode, you can use the progress bar to see how the job is progressing (figure 35.36): a process called "Batch Process" indicates how many batches have been completed, while the ones situated above show the analysis progress of a particular batch unit.

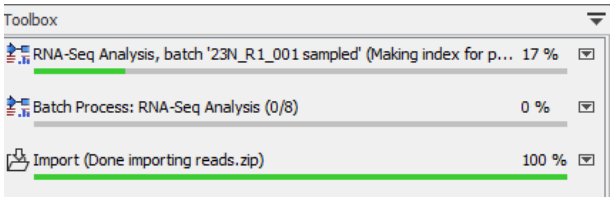


Figure 35.36: *Check on the progress of your workflow being run in batch mode using the Processes tab below the Toolbox.*

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 and Merge Counts (legacy)
- 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
- Compare Sample Variant Tracks (legacy)
- Copy Number Variant Detection
- Create Alignment
- Create RRBS-fragment Track

- Demultiplex Reads
- De Novo Assembly
- Differential Expression
- Differential Expression in Two Groups

- Extract and Count (legacy)

- 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 Reads to Contigs
- Map Reads to Reference
- Maximum Likelihood Phylogeny
- Merge Annotation Tracks
- Model Testing

- 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
- Remove Reference Variants (legacy)
- RNA-Seq Analysis

- Trim Reads
- Trio 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 http://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 http://www.rcsb.org/pdb/.
- **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 Sagarsso Sea project. This does overlap with nucleotide nr.
- **tsa_nt.** Transcriptome Shotgun Assembly database.
- **prokaryotic_16S_ribosomal_RNA.** 16S ribomsal 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[clcdefault] = 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).

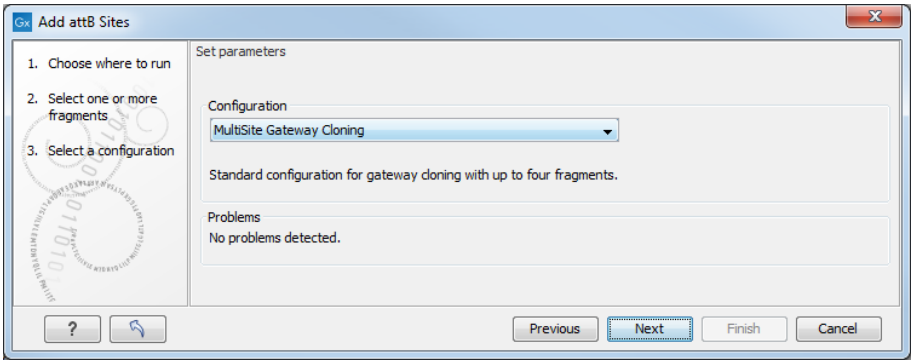


Figure F.1: *Selecting between different gateway cloning configurations.*

Appendix G

IUPAC codes for amino acids

(Single-letter codes based on International Union of Pure and Applied Chemistry)

The information is gathered from: http://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: http://www.iupac.org and http://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

Note that high-throughput sequencing data formats from Illumina, IonTorrent, 454 and also high-throughput fasta and trace files are imported using a special import as described in section 6.3. These data can also be exported in fastq format (using NCBI/Sanger Phred quality scores).

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	.fsa/.fasta	X	X	Simple format, name & description
FASTQ	.fastq	X	X	Simple format, name & description
GenBank	.gbk/.gb/.gp/.gbff	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
Vector NTI archives*	.ma4/.pa4/.oa4	X		Archives in rich format
Vector NTI Database*		X		Special import full database

*Vector NTI import functionality comes as standard within the CLC Main Workbench and can be installed as a plugin via the Plugins Manager of the CLC Genomics Workbench (read more in section 1.5).

When exporting in fasta format, it is possible to remove sequence ends covered by annotations of type "Trim" (read more in section 19.2).

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	Compressed version of SAM. See details in section 6.3.8
CLC	.clc	X	X	Rich format including all information
CLC Assembly File	.cas	X		Output from the CLC Assembly Cell
SAM	.sam	X	X	Sequence Alignment/Map. See details in section 6.3.8
Mapping coverage	.tsv		X	Detailed per-base info on coverage (see below)

Note about BAM export Index files can be created as part of BAM exports.

Note about AGP format Both sequence lists and contigs with reads mapped can be used. Based on annotations of type **Scaffold** (which are automatically added when running the *de novo* assembly with the scaffold option), the contigs are broken up before exported as fasta. The agp file produced holds information about how the contigs relate to each other.

Export of coverage information from sequence alignments

Coverage information from read mappings can be exported in a tabular format using the **Mapping coverage** export. The output contains information on the number of nucleotides aligned to positions in 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 A's
6	Number of C's
7	Number of G's
8	Number of T's
9	Number of N's
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.

I.1.3 Alignment formats

File type	Suffix	Import	Export	Description
Aligned fasta	.fa	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

Read about technical details of these data formats in section K.

File type	Suffix	Import	Export	Description
Affymetrix CHP	.chp/.psi	X		Expression values and annotations
Affymetrix pivot/metric	.txt/.csv	X		Gene-level expression values
Affymetrix NetAffx	.csv	X		Annotations
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 soft sample/series	.txt/.csv	X		Expression values
Illumina	.txt	X		Expression values and annotations
Table CSV	.csv		X	Samples and experiments
Tab delimited	.txt		X	Samples and experiments

I.1.6 Annotation and variant formats

Please note that all of the annotation and variant formats can be imported as tracks (see section 6.2). GFF, GVF and GTF formats can also be imported as annotations on a standard (i.e., non-track) sequence or sequence list using functionality provided by the Annotate with GFF plugin (https://digitalinsights.qiagen.com/plugins/annotate-with-gff-file/).

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
VCF	.vcf	X	X	See section 6.2.2 and section 6.6.7
GFF	.gff	X		To import as annotation track, see section 6.2.
GVF	.gvf	X	X	Special version of GFF for variant data. See GFF entry above.
GTF	.gtf	X	X	Special version of GFF for gene annotation data. See GFF entry above.
GFF3	.gff3	X	X	To import and export as annotation track, see section 6.2.1.
COSMIC variation database	.tsv	X		Special format for COSMIC data
BED	.bed	X	X	See section 6.2
Wiggle	.wig	X	X	See section 6.2
UCSC variant database table dump	.txt	X		See section 6.2

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 == chromsome_26

For chromsome 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 the Annotate with GFF plugin. There, your reference names in the 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 30.9.1.

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
uaugcaauuu ucuaccuuac 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	.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 table I.1.5 **Expression data formats** 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 Workbench and view these in the **Navigation Area**, but it is only the above mentioned formats whose *contents* can be shown in the 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 6.7 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 export format specification

SAM Specification The workbench aims to import and export SAM and BAM files according to the v1.4-r962 version of the SAM specification (see http://samtools.sourceforge.net/SAM1.pdf). This appendix describes how the workbench exports SAM and BAM files along with known limitations.

SAM and BAM Export - General notes The SAM exporter writes unsorted SAM and BAM files.

If the reference name contains spaces, the spaces are removed. Each occurrence of '=' (equals sign) and '@' (at sign) in a reference name is replaced by an '_' (underscore).

The SAM importer and exporter support the ID, SM, PI and PL read group tags. All other read group tags are ignored.

The BAM exporter can also output additional annotations added by tools provided by plugins, and where that is the case, further details are provided in the plugin manual.

SAM Alignment Section A few remarks on the exported alignment section:

- 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'.
- CLC software does not support or record mapping quality for read mappings. To fulfill the requirement in the BAM format specifications that a read mapping quality is recorded for all mapped reads, the values 0 and 60 are used when mappings are exported from the Workbench. 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 BAM 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.

J.1 Flags

The workbench's use of the alignment flags is shown in the following table and subsequent examples.

Bit	SAM description	Usage in Workbench
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[1].
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 workbench.

[1]The representation of a particular read with more than one location in a mapping is not supported in the software and thus cannot be imported.

Description of the example	Bits	Flag	Illustration
The first mate of a non-broken paired read	`0x1`, `0x2`, `0x20`, `0x40`	**99**	See figure J.1
The second mate of a non-broken paired read	`0x1`, `0x2`, `0x10`, `0x80`	**147**	See figure J.2
A single, forward read (or paired read, where only one mate of the pair is mapped)	No set bits	**0**	see figure J.3
A single, reversed read (or paired read, where only one mate of the pair is mapped)	`0x10`	**16**	See figure J.4
The first, forward segment from a broken pair with forward mate	`0x1`, `0x40`	**65**	See figure J.5
The second, forward segment from broken pair with reversed mate	`0x1`, `0x20`, `0x80`	**161**	See figure J.6
The first, reversed segment from broken pair with forward mate	`0x1`, `0x10`, `0x40`	**81**	See figure J.7
The second, reversed segment from broken pair with reversed mate	`0x1`, `0x10`, `0x20`, `0x80`	**177**	See figure J.8

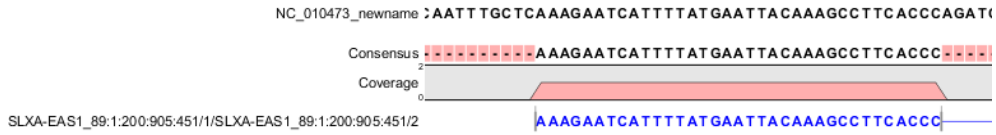


Figure J.1: *The read is paired, both reads are mapped and the mate of this read is reversed*

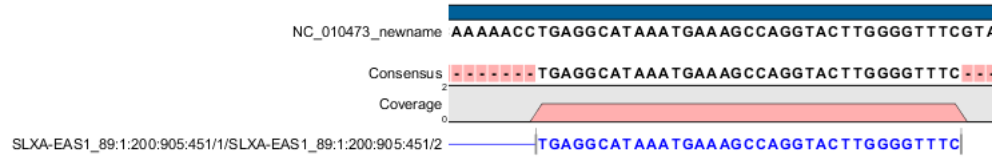


Figure J.2: *The read is paired, both mates are mapped, and this segment is reversed*

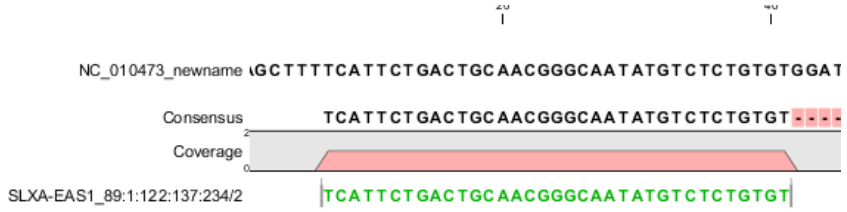


Figure J.3: *A single, forward read, or a paired read where the mate is not mapped*

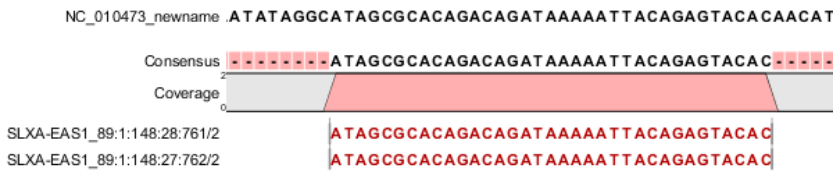


Figure J.4: *The read is a single, reversed read, or a paired read where the mate is not mapped*

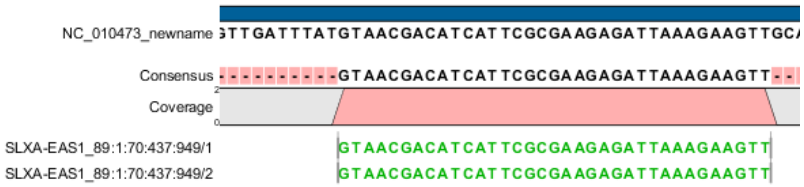


Figure J.5: *These forward reads are paired. They map to the same place, so the pair is broken*

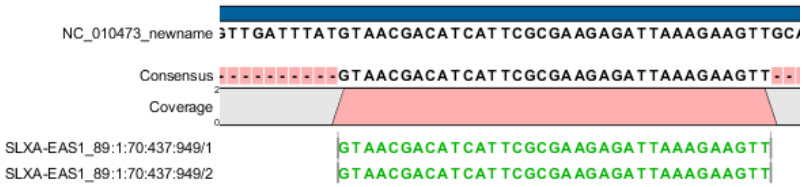


Figure J.6: *Forward read that is part of a broken read where the mate is reversed*

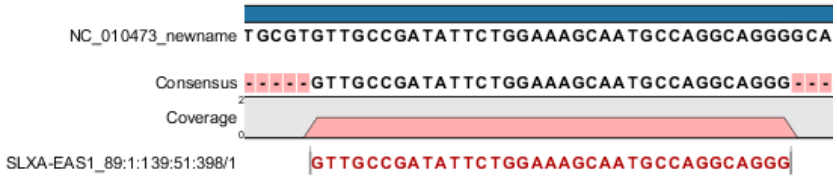


Figure J.7: *Reversed read that is part of a broken pair, where the mate is forward*

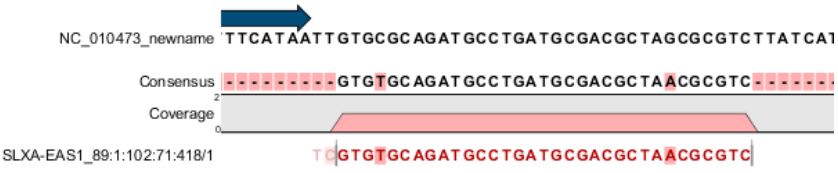


Figure J.8: *Reversed read that is part of a broken pair, where the mate is also reversed.*

Appendix K

Gene expression annotation files and microarray data formats

The 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 section K.5).

The 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 workbench allows import of txt files exported from R containing processed Affymetrix CEL-file data (see see section K.2).

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 section K.5).

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 `http://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
```

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`,

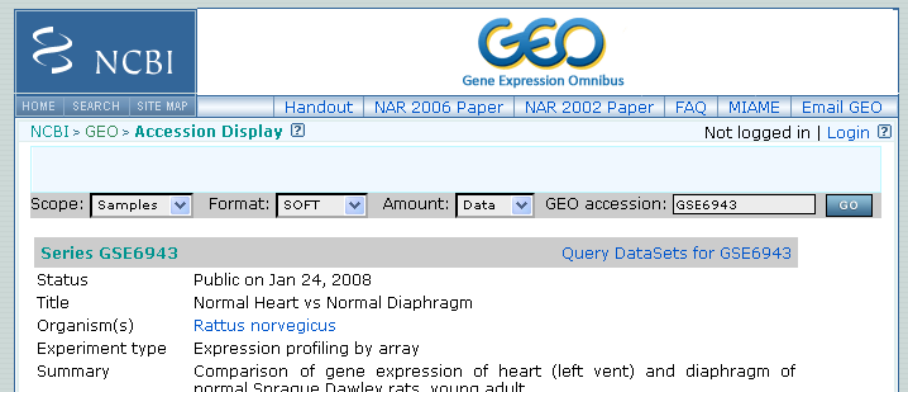


Figure K.1: *Selecting Samples, SOFT and Data before clicking go will give you the format supported by the* **CLC Genomics Workbench**.

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 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 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 probe expression levels to 'evals-txt'
```

The exported txt file (evals.txt) can be imported into the workbench using the Generic expression data table format importer (see section K.5; 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 metrix 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 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 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 31.1.3.

Download a small example annotation file here which includes header information:
https://resources.qiagenbioinformatics.com/madata/AffymetrixNetAffxAnnotationFil
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 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 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/IlluminaBeadStudioWithAnnotati
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/IlluminaBeadStudioMultipleSamp
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 workbench recognizes the 18 samples and their columns.

Illumina annotation files The 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 31.1.3.

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 31.1.3. 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 http://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 identfiers 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 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, ProbeID, 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
Ensemble 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	Trancript assignments
mrna_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 `http://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 29.3.1)	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 29.3.3)	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 29.3.2)	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 29.1.4)	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 29.1.1)	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 29.2.1)	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 24.8.4)	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.

[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.

[Creighton et al., 2009] Creighton, C. J., Reid, J. G., and Gunaratne, P. H. (2009). Expression profiling of microrna by deep sequencing. *Brief Bioinform*, 10(5):490–497.

[Cronn et al., 2008] Cronn, R., Liston, A., Parks, M., Gernandt, D. S., Shen, R., and Mockler, T. (2008). Multiplex sequencing of plant chloroplast genomes using solexa sequencing-by-synthesis technology. *Nucleic Acids Res*, 36(19):e122.

[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 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., Wünning, 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., Shoresh, N., Sidow, A., Slattery, 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 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-rota: 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*, 10:4288–4297.

[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.

[Morin et al., 2008] Morin, R. D., O'Connor, M. D., Griffith, M., Kuchenbauer, F., Delaney, A., Prabhu, A.-L., Zhao, Y., McDonald, H., Zeng, T., Hirst, M., Eaves, C. J., and Marra, M. A. (2008). Application of massively parallel sequencing to microrna profiling and discovery in human embryonic stem cells. *Genome Res*, 18(4):610–621.

[Morrison, 1968] Morrison, D. R. (1968). Patricia – practical algorithm to retrieve information coded in alphanumeric. *J. ACM*, 15(4):514–534.

[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.

[Parkhomchuk et al., 2009] Parkhomchuk, D., Borodina, T., Amstislavskiy, V., Banaru, M., Hallen, L., Krobitsch, 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.

[Robinson and Smyth, 2007] Robinson, M. D. and Smyth, G. K. (2007). Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics*, 23(21):2881–2887.

[Robinson and Smyth, 2008] Robinson, M. D. and Smyth, G. K. (2008). Small-sample estimation of negative binomial dispersion, with applications to sage data. *Biostatistics*, 9(2):321–332.

[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.

[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.

[Stark et al., 2010] Stark, M. S., Tyagi, S., Nancarrow, D. J., Boyle, G. M., Cook, A. L., Whiteman, D. C., Parsons, P. G., Schmidt, C., Sturm, R. A., and Hayward, N. K. (2010). Characterization of the melanoma mirnaome by deep sequencing. *PLoS One*, 5(3):e9685.

[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.*

[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.

[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.

[Wyman et al., 2009] Wyman, S. K., Parkin, R. K., Mitchell, P. S., Fritz, B. R., O'Briant, K., Godwin, A. K., Urban, N., Drescher, C. W., Knudsen, B. S., and Tewari, M. (2009). Repertoire of microrna s in epithelial ovarian cancer as determined by next generation sequencing of small rna cdna libraries. *PLoS One*, 4(4):e5311.

[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.

[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 Mathemetical 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.