


# CLC **Genomics** Workbench

USER MANUAL

Manual for
# *CLC Genomics Workbench 8.0*
Windows, Mac OS X and Linux

February 25, 2015

**This software is for research purposes only.**

CLC bio, a QIAGEN Company
Silkeborgvej 2
Prismet
DK-8000 Aarhus C
Denmark

# Contents


**I Introduction** **12**

**1 Introduction to *CLC Genomics Workbench*** **13**

1.1 Contact information . . . . . 15
1.2 Download and installation . . . . . 15
1.3 System requirements . . . . . 18
1.4 Workbench Licenses . . . . . 19
1.5 About CLC Workbenches . . . . . 34
1.6 When the program is installed: Getting started . . . . . 36
1.7 Plugins . . . . . 37
1.8 Network configuration . . . . . 40
1.9 The format of the user manual . . . . . 40
1.10 Latest improvements . . . . . 41

**II Core Functionalities** **42**

**2 User interface** **43**

2.1 View Area . . . . . 44
2.2 Zoom and selection in View Area . . . . . 52
2.3 Toolbox and Status Bar . . . . . 54
2.4 Workspace . . . . . 57
2.5 List of shortcuts . . . . . 58

**3 Data management and search** **61**

3.1 Navigation Area . . . . . 62
3.2 Customized attributes on data locations . . . . . 70

3.3 Filling in values . . . 72
3.4 Local search . . . 74

**4 User preferences and settings 82**
4.1 General preferences . . . 82
4.2 Default view preferences . . . 84
4.3 Data preferences . . . 86
4.4 Advanced preferences . . . 87
4.5 Export/import of preferences . . . 87
4.6 View settings for the Side Panel . . . 88

**5 Printing 91**
5.1 Selecting which part of the view to print . . . 92
5.2 Page setup . . . 93
5.3 Print preview . . . 94

**6 Import/export of data and graphics 95**
6.1 Standard import . . . 96
6.2 Import tracks . . . 98
6.3 Import high-throughput sequencing data . . . 100
6.4 Data export . . . 120
6.5 Export graphics to files . . . 128
6.6 Export graph data points to a file . . . 133
6.7 Copy/paste view output . . . 134

**7 History log 136**
7.1 Element history . . . 136

**8 Batching and result handling 139**
8.1 Batch processing . . . 139
8.2 How to handle results of analyses . . . 142
8.3 Working with tables . . . 144

**9 Workflows 148**
9.1 Creating a workflow . . . 149

9.2 Distributing and installing workflows . . . 165
9.3 Executing a workflow . . . 173
9.4 Open copy of installed workflow . . . 174

**III Basic sequence analysis 176**

**10 Viewing and editing sequences 177**

10.1 View sequence . . . 177
10.2 Circular DNA . . . 186
10.3 Working with annotations . . . 189
10.4 Element information . . . 197
10.5 View as text . . . 198
10.6 Sequence Lists . . . 198

**11 Data download 202**

11.1 GenBank search . . . 202
11.2 UniProt (Swiss-Prot/TrEMBL) search . . . 206
11.3 Search for structures at NCBI . . . 208
11.4 Download reference genome data . . . 212
11.5 Sequence web info . . . 215

**12 BLAST search 217**

12.1 Running BLAST searches . . . 218
12.2 Output from BLAST searches . . . 226
12.3 Local BLAST databases . . . 232
12.4 Manage BLAST databases . . . 235
12.5 Bioinformatics explained: BLAST . . . 236

**13 3D Molecule Viewer 245**

13.1 Importing molecule structure files . . . 246
13.2 Viewing molecular structures in 3D . . . 250
13.3 Customizing the visualization . . . 252
13.4 Snapshots of the molecule visualization . . . 260
13.5 Tools for linking sequence and structure . . . 260

13.6 Protein structure alignment . . . . . 263

**14 General sequence analyses** **268**

14.1 Extract Annotations . . . . . 268
14.2 Extract sequences . . . . . 270
14.3 Shuffle sequence . . . . . 272
14.4 Dot plots . . . . . 274
14.5 Local complexity plot . . . . . 284
14.6 Sequence statistics . . . . . 285
14.7 Join sequences . . . . . 291
14.8 Pattern discovery . . . . . 292
14.9 Motif Search . . . . . 294
14.10 Create motif list . . . . . 300

**15 Nucleotide analyses** **302**

15.1 Convert DNA to RNA . . . . . 302
15.2 Convert RNA to DNA . . . . . 303
15.3 Reverse complements of sequences . . . . . 304
15.4 Reverse sequence . . . . . 304
15.5 Translation of DNA or RNA to protein . . . . . 305
15.6 Find open reading frames . . . . . 307

**16 Protein analyses** **310**

16.1 Signal peptide prediction . . . . . 311
16.2 Protein charge . . . . . 317
16.3 Transmembrane helix prediction . . . . . 318
16.4 Antigenicity . . . . . 319
16.5 Hydrophobicity . . . . . 321
16.6 Pfam domain search . . . . . 326
16.7 Secondary structure prediction . . . . . 328
16.8 Protein report . . . . . 329
16.9 Reverse translation from protein into DNA . . . . . 332
16.10 Proteolytic cleavage detection . . . . . 335

**17 Primers** **341**

17.1 Primer design - an introduction . . . 342
17.2 Setting parameters for primers and probes . . . 344
17.3 Graphical display of primer information . . . 347
17.4 Output from primer design . . . 348
17.5 Standard PCR . . . 349
17.6 Nested PCR . . . 353
17.7 TaqMan . . . 355
17.8 Sequencing primers . . . 357
17.9 Alignment-based primer and probe design . . . 358
17.10 Analyze primer properties . . . 363
17.11 Find binding sites and create fragments . . . 364
17.12 Order primers . . . 368

**18 Sequencing data analyses** **370**

18.1 Importing and viewing trace data . . . 371
18.2 Trim sequences . . . 372
18.3 Assemble sequences . . . 375
18.4 Sort sequences by name . . . 377
18.5 Assemble sequences to reference . . . 380
18.6 Add sequences to an existing contig . . . 383
18.7 View and edit read mappings . . . 383
18.8 Reassemble contig . . . 393
18.9 Secondary peak calling . . . 394

**19 Cloning and cutting** **396**

19.1 Molecular cloning . . . 397
19.2 Gateway cloning . . . 407
19.3 Restriction site analysis . . . 416
19.4 Gel electrophoresis . . . 430
19.5 Restriction enzyme lists . . . 432

**20 Sequence alignment** **436**

20.1 Create an alignment . . . 437
20.2 View alignments . . . 442
20.3 Edit alignments . . . 446
20.4 Join alignments . . . 449
20.5 Pairwise comparison . . . 450
20.6 Bioinformatics explained: Multiple alignments . . . 453

**21 Phylogenetic trees** **455**
21.1 Phylogenetic tree features . . . 455
21.2 Create Trees . . . 456
21.3 Tree Settings . . . 470
21.4 Metadata and phylogenetic trees . . . 480

**22 RNA structure** **486**
22.1 RNA secondary structure prediction . . . 487
22.2 View and edit secondary structures . . . 493
22.3 Evaluate structure hypothesis . . . 501
22.4 Structure scanning plot . . . 502
22.5 Bioinformatics explained: RNA structure prediction by minimum free energy minimization . . . 506

**IV High-throughput sequencing** **512**

**23 Trimming, multiplexing and sequencing quality control** **513**
23.1 Trim Sequences . . . 513
23.2 Demultiplex reads . . . 524
23.3 Sequencing data quality control . . . 533
23.4 Merge overlapping pairs . . . 537

**24 Tracks** **542**
24.1 Track lists . . . 544
24.2 Retrieving reference data tracks . . . 555
24.3 Merging tracks . . . 556
24.4 Converting data to tracks and back . . . 557

24.5 Annotate and filter tracks . . . 559
24.6 Creating graph tracks . . . 564

**25 Read mapping** **571**
25.1 Map Reads to Reference . . . 572
25.2 Mapping output options . . . 581
25.3 Mapping reports . . . 582
25.4 Color space . . . 590
25.5 Mapping result . . . 594
25.6 Local realignment . . . 601
25.7 Merge mapping results . . . 609
25.8 Extract consensus sequence . . . 610
25.9 Coverage analysis . . . 613

**26 Sample Reads** **616**
26.1 What is Sample Reads? . . . 616
26.2 How to run Sample Reads . . . 616

**27 Resequencing** **618**
27.1 Create Statistics for Target Regions . . . 620
27.2 Variant Detectors - overview . . . 628
27.3 Basic Variant Detection . . . 633
27.4 Fixed Ploidy Variant Detection . . . 633
27.5 Low Frequency Variant Detection . . . 635
27.6 Variant Detectors - error model estimation . . . 636
27.7 Variant Detectors - filters . . . 637
27.8 Variant Detectors - the outputs . . . 644
27.9 The Fixed Ploidy and Low Frequency variant callers: detailed descriptions . . . 649
27.10 InDels and Structural Variants . . . 658
27.11 Variant data . . . 672
27.12 Detailed information about overlapping paired reads . . . 677
27.13 Annotate and filter variants . . . 678
27.14 Comparing variants . . . 684

27.15 Predicting functional consequences . . . . . 691

**28 Transcriptomics** **712**

28.1 RNA-Seq analysis . . . . . 713
28.2 Small RNA analysis . . . . . 730
28.3 Expression profiling by tags . . . . . 749
28.4 Experimental design . . . . . 760
28.5 Working with tracks and experiments . . . . . 774
28.6 Transformation and normalization . . . . . 780
28.7 Quality control . . . . . 784
28.8 Statistical analysis - identifying differential expression . . . . . 798
28.9 Feature clustering . . . . . 809
28.10 Annotation tests . . . . . 817
28.11 General plots . . . . . 823

**29 De novo sequencing** **830**

29.1 *De novo* assembly . . . . . 830
29.2 Map reads to contigs . . . . . 846

**30 Epigenomics** **850**

30.1 ChIP-Seq Analysis . . . . . 850
30.2 Annotate with nearby gene information . . . . . 857

**31 Legacy tools** **859**

31.1 Quality-based variant detection . . . . . 860
31.2 Probabilistic variant detection . . . . . 866
31.3 ChIP-Seq Analysis (legacy) . . . . . 875

**V Appendix** **883**

**A Comparison of workbenches** **884**

**B Use of multi-core computers** **889**

**C Graph preferences** **891**

**D BLAST databases** **893**
D.1 Peptide sequence databases . . . 893
D.2 Nucleotide sequence databases . . . 893
D.3 Adding more databases . . . 894

**E Proteolytic cleavage enzymes** **896**

**F Restriction enzymes database configuration** **898**

**G Technical information about modifying Gateway cloning sites** **899**

**H IUPAC codes for amino acids** **901**

**I IUPAC codes for nucleotides** **902**

**J Formats for import and export** **903**
J.1 List of bioinformatic data formats . . . 903
J.2 List of graphics data formats . . . 911

**K SAM/BAM export format specification** **912**
K.1 Flags . . . 913

**L Gene expression annotation files and microarray data formats** **916**
L.1 GEO (Gene Expression Omnibus) . . . 916
L.2 Affymetrix GeneChip . . . 919
L.3 Illumina BeadChip . . . 920
L.4 Gene ontology annotation files . . . 922
L.5 Generic expression and annotation data file formats . . . 922

**M Custom codon frequency tables** **926**

**N Comparison of track comparison tools** **927**

**Bibliography** **929**

**VI Index** **939**

# Part I

# Introduction

# Chapter 1

# Introduction to *CLC Genomics Workbench*

## Contents


**1.1 Contact information . . . 15**
**1.2 Download and installation . . . 15**
1.2.1 Program download . . . 15
1.2.2 Installation on Microsoft Windows . . . 15
1.2.3 Installation on Mac OS X . . . 16
1.2.4 Installation on Linux with an installer . . . 17
1.2.5 Installation on Linux with an RPM-package . . . 18
**1.3 System requirements . . . 18**
1.3.1 Limitations on maximum number of cores . . . 19
**1.4 Workbench Licenses . . . 19**
1.4.1 Request an evaluation license . . . 20
1.4.2 Download a license using a license order ID . . . 23
1.4.3 Import a license from a file . . . 25
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 . . . 32
1.4.7 Limited mode . . . 34
**1.5 About CLC Workbenches . . . 34**
1.5.1 New program feature request . . . 35
1.5.2 Getting help . . . 35
1.5.3 CLC Sequence Viewer vs. Workbenches . . . 36
**1.6 When the program is installed: Getting started . . . 36**
1.6.1 Quick start . . . 36
1.6.2 Import of example data . . . 37
**1.7 Plugins . . . 37**
1.7.1 Installing plugins . . . 37
1.7.2 Uninstalling plugins . . . 38
1.7.3 Updating plugins . . . 39
1.7.4 Resources . . . 40
**1.8 Network configuration . . . 40**

**1.9 The format of the user manual . . . . . 40**
1.9.1 Text formats . . . . . 41
**1.10 Latest improvements . . . . . 41**

Welcome to *CLC Genomics Workbench* — 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

The *CLC Genomics Workbench* is developed by:

CLC bio, a QIAGEN Company
Silkeborgvej 2
Prismet
8000 Aarhus C
Denmark

http://www.clcbio.com

VAT no.: DK 28 30 50 87

Telephone: 45 70 22 32 44
Fax: +45 86 20 12 22

E-mail: info@clcbio.com

If you have questions or comments regarding the program, you can contact us through the support team as described here: http://www.clcsupport.com/clcgenomicsworkbench/current/index.php?manual=Getting_help.html.

## 1.2 Download and installation

The *CLC Genomics Workbench* is developed for Windows, Mac OS X and Linux. The software for either platform can be downloaded from http://www.clcbio.com/download.

### 1.2.1 Program download

The program is available for download on http://www.clcbio.com/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

Depending on your operating system and your Internet browser, you are taken through some download options.

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]

### 1.2.2 Installation on Microsoft Windows

Starting the installation process is done in one of the following ways:

[1] You must be connected to the Internet throughout the installation process.

*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 Mac OS X

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_7_5_1_64.sh.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:

```
# clcgenomicswb7
```

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

```
# ./clcgenomicswb7
```

### 1.2.5 Installation on Linux with an RPM-package

Navigate to the directory containing the rpm-package and install it using the rpm-tool by running a command similar to:

```
# rpm -ivh CLCGenomicsWorkbench_7_5_1_64.sh.rpm
```

Installation of RPM-packages usually requires root-privileges.

When the installation process is finished the program can be executed by running the command:

```
# clcgenomicswb7
```

## 1.3 System requirements

- Windows Vista, Windows 7, Windows 8 or Windows Server 2008
- Mac OS X 10.7 or later.
- Linux: RHEL 5.0 or later. SUSE 10.2 or later. Fedora 6 or later.
- 2 GB RAM required
- 4 GB RAM recommended
- 1024 x 768 display required
- 1600 x 1200 display recommended
- Intel or AMD CPU required

  **Special requirements for the 3D Molecule Viewer**

  - **System 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.
  - **System 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.
    - A 64-bit workbench version is recommended for working with large complexes.

- **Special 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. Systems with less memory than specified below will benefit from installing the legacy read mapper plugin (see http://www.clcbio.com/plugins). This is slower than the standard mapper but adjusts to the amount of memory available.

  - **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://www.clcbio.com/white-paper for examples of the memory usage of various data sets.

- 64 bit computer and operating system required to use more than 2 GB RAM

### 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 therm "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 http://www.clcbio.com/desktop-applications/licensing/).

## 1.4 Workbench Licenses

When you have installed the *CLC Genomics Workbench*, and start it for the first time, you will meet the license assistant, shown in figure 1.1. The **License Manager** can also be accessed from the menu bar in the Workbench:

**Help** | **License Manager**

This can be useful if you wish to use a different license or want to view information about the license(s) the Workbench is currently using. The **License Manager** is described in detail in section 1.4.5 and can be seen in figure 1.23.

To install a license, you must be running the program in administrative mode [2].

The following options are available. They are described in detail in the sections that follow.

- **Request an evaluation license**. Request a fully functional, time-limited license (see below).

[2] How to do this differs for different operating systems. To run the program in administrator mode on Windows Vista, or 7, right-click the program shortcut and choose "Run as Administrator.

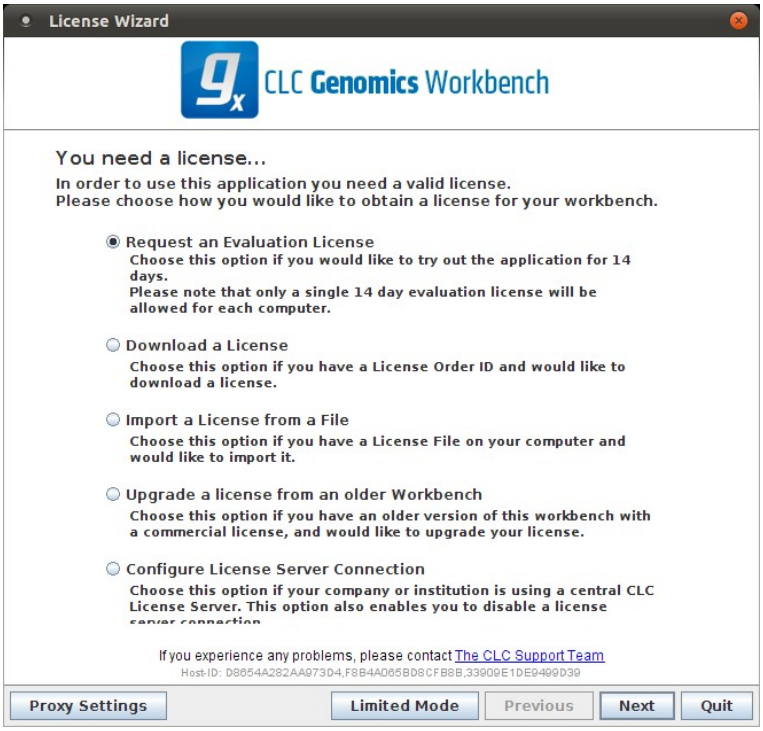



Figure 1.1: *The license assistant showing you the options for getting started.*

- **Download a license**. Use the license order ID received 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 web-based licensing system.
- **Upgrade license**. 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 click on button labeled **Next**.

To use the Download option in the License Manager, your machine must be able to access the external network. If this is not the case, please see section 1.4.6.

If for some reason you don't have a license order ID or access to a license, you can click the **Limited Mode** button (see section 1.4.7).

### 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 user is entitled to 14 days demo of *CLC Genomics Workbench*.

If you are unable to complete your assessment in the available time, please send an email to sales@clcbio.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.

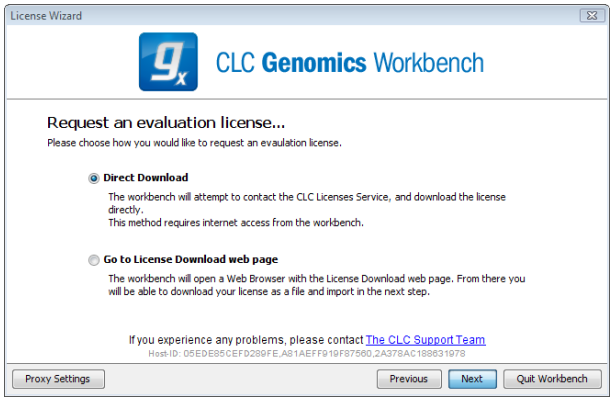



Figure 1.2: *Choosing between direct download or going to the license download web page.*

In this dialog, there are two options:

- **Direct download**. Download the license directly from CLC bio. This method requires that the Workbench has access to the external network.
- **Go to license download web page**. In a browser window, show the license download web page, which can be used to download a license file. This option is suitable in situations where, for example, you are working behind a proxy, so that the Workbench does not have direct access to the CLC Licenses Service.

If you select the option to download a license directly and it turns out that the Workbench does not have direct access to the external network, (because of a firewall, proxy server etc.), you can click **Previous** button to try the other method.

After selection on 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**, the dialog shown in figure 1.3 appears.

A progress for getting the license is shown, and when the license is downloaded, you will be able to click **Next**.

### Go to license download web page

After choosing the *Go to license download web page* option and clicking on the button labeled **Next**, the license download web page appears in a browser window, as shown in 1.4.

Click the **Request Evaluation License** button. You can then save the license on your system.

Back in the Workbench window, you will now see the dialog shown in 1.5.

Click the **Choose License File** button and browse to find the license file you saved. When you have selected the file, click on the button labeled **Next**.

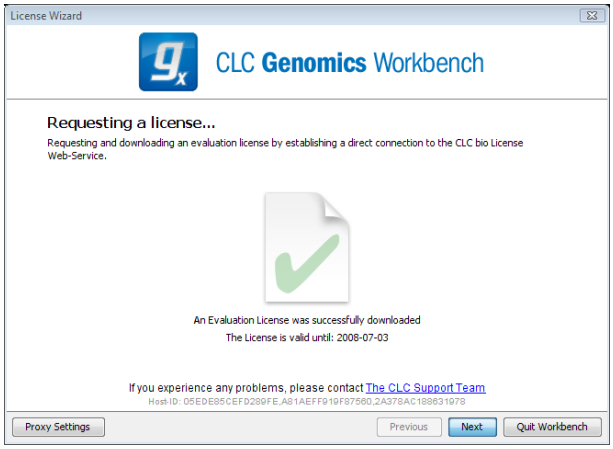



Figure 1.3: *A license has been downloaded.*

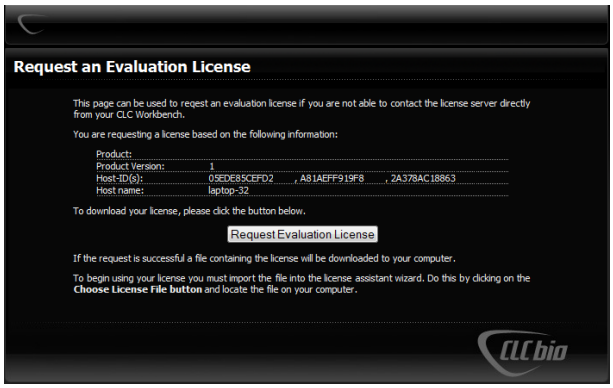



Figure 1.4: *The license download web page.*

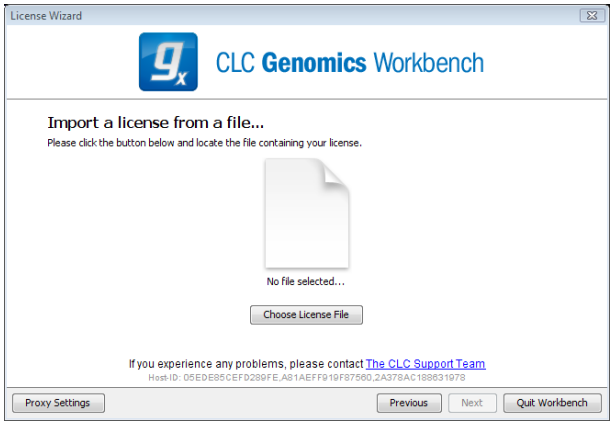



Figure 1.5: *Importing the license file downloaded from the web page.*

### Accepting the license agreement

Part of the installation of the license involves checking and accepting the end user license agreement (EULA). You should now see the a window like that in figure 1.6.

Please read the EULA text carefully before clicking in the box next to the text **I accept these terms** to accept, and then clicking on the button labeled **Finish**.

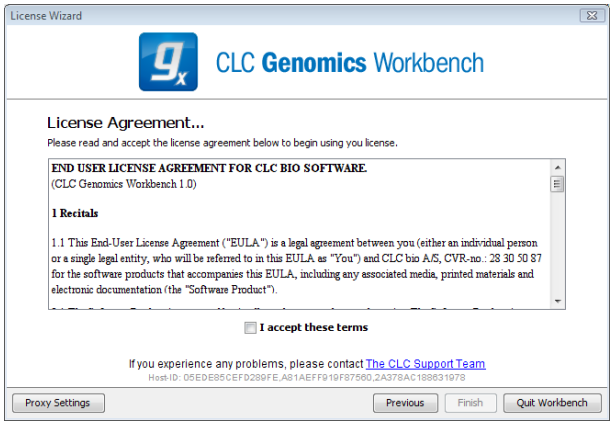



Figure 1.6: *Read the license agreement carefully.*

### 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 **Next** button, you will see the dialog shown in 1.7. Enter your license order ID into the text field under the title License Order-ID. (The ID can be pasted into the box after copying it and then using menus or key combinations like Ctrl+V on some system or ⌘ + V on Mac).

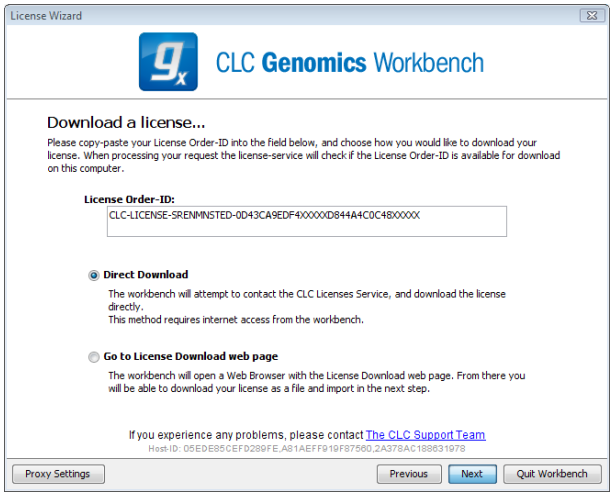



Figure 1.7: *Enter a license order ID for the software.*

In this dialog, there are two options:

- **Direct download**. Download the license directly from CLC bio. This method requires that the Workbench has access to the external network.
- **Go to license download web page**. In a browser window, show the license download web page, which can be used to download a license file. This option is suitable in situations where, for example, you are working behind a proxy, so that the Workbench does not have direct access to the CLC Licenses Service.

If you select the option to download a license directly and it turns out that the Workbench does not have direct access to the external network, (because of a firewall, proxy server etc.), you can click **Previous** button to try the other method.

After selection on 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**, the dialog shown in figure 1.8 appears.

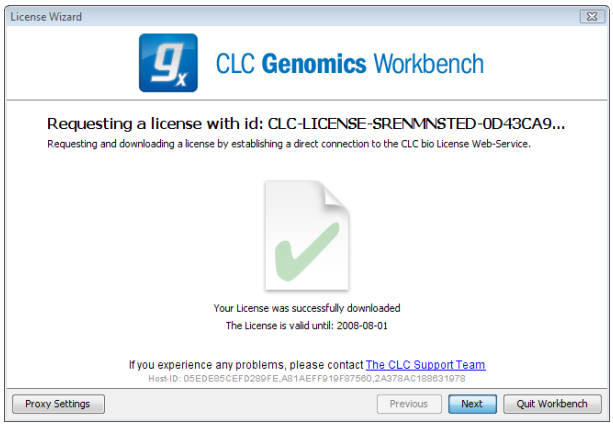



Figure 1.8: *A license has been downloaded.*

A progress for getting the license is shown, and when the license is downloaded, you will be able to click **Next**.

### Go to license download web page

After choosing the *Go to license download web page* option and clicking on the button labeled **Next**, the license download web page appears in a browser window, as shown in 1.9.

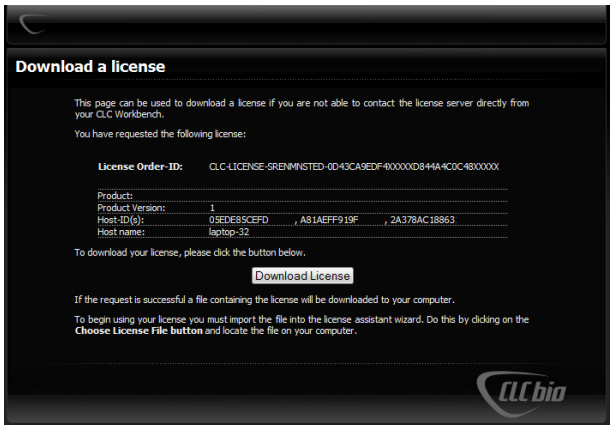



Figure 1.9: *The license download web page.*

Click the **Request Evaluation License** button. You can then save the license on your system.

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

Click the **Choose License File** button and browse to find the license file you saved. When you have selected the file, click on the button labeled **Next**.

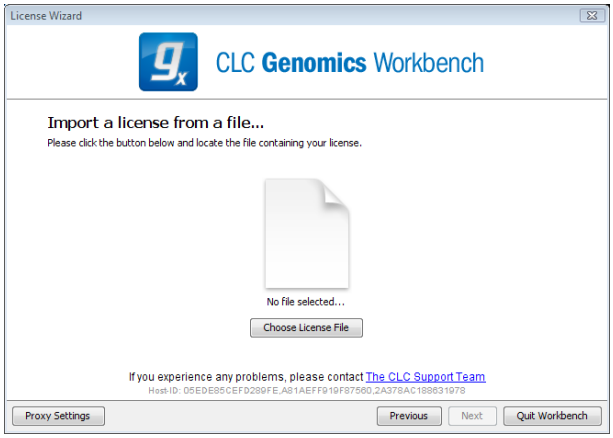



Figure 1.10: *Importing the license file downloaded from the web page.*

#### Accepting the license agreement

Part of the installation of the license involves checking and accepting the end user license agreement (EULA). You should now see the a window like that in figure 1.11.

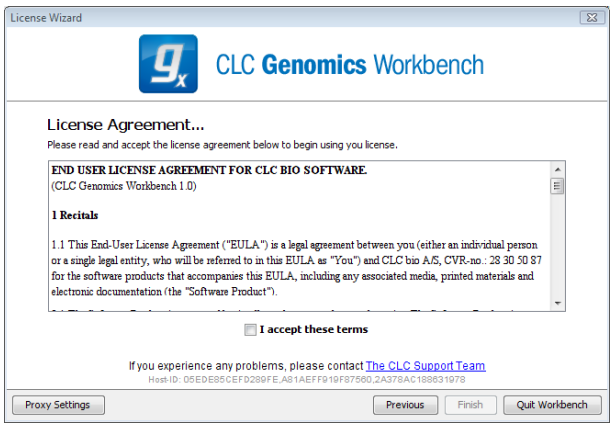



Figure 1.11: *Read the license agreement carefully.*

Please read the EULA text carefully before clicking in the box next to the text **I accept these terms** to accept, and then clicking on the button labeled **Finish**.

### 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 1.12.

Click the **Choose License File** button and browse to find the license file. When you have selected the file, click on the **Next** button.

#### Accepting the license agreement

Part of the installation of the license involves checking and accepting the end user license agreement (EULA). You should now see the a window like that in figure 1.13.

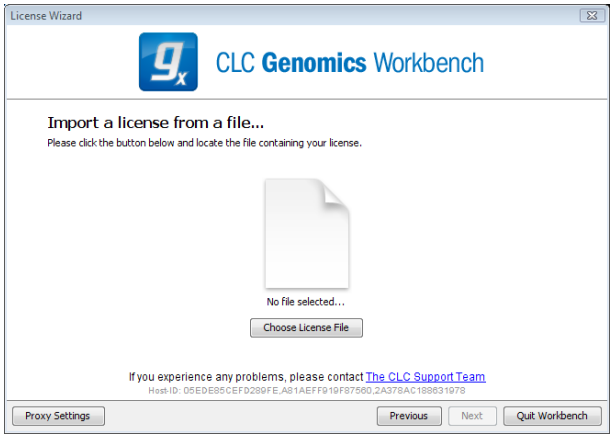



Figure 1.12: *Selecting a license file .*

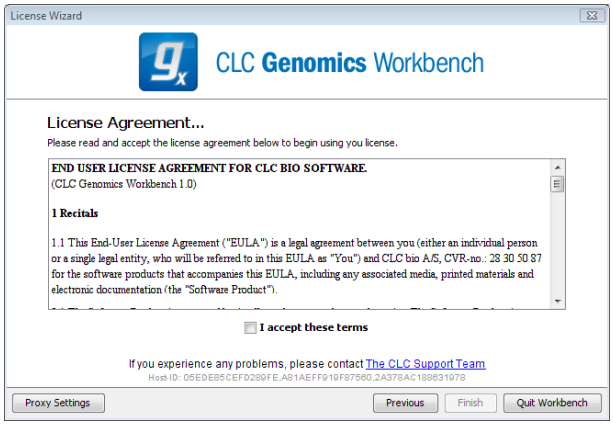



Figure 1.13: *Read the license agreement carefully.*

Please read the EULA text carefully before clicking in the box next to the text **I accept these terms** to accept, and then clicking on the button labeled **Finish**.

### 1.4.4 Upgrade license

This option is used when you already have used a previous version of *CLC Genomics Workbench*, and you are entitled to upgrade to a new major version. The Workbench will need direct access to the external network to use this option.

When you click on the **Next** button, the Workbench will search for a previous installation of *CLC Genomics Workbench*. It will then locate the old license.

If the Workbench finds an existing license file, the next dialog will look like figure 1.14.

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

If the Workbench cannot connect to the external network directly, please see the section on downloading a license for non-networked machines. You will need your license order ID for this.

Your license must be covered by our Maintenance, Upgrades and Support (MUS) program to be eligible to upgrade your license. If the license is covered for upgrades and there are any problems with this, please contact licenses@clcbio.com.

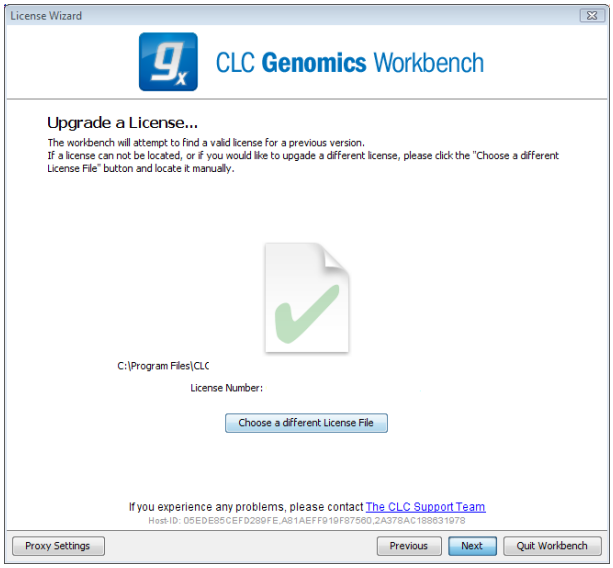



Figure 1.14: *An license from an older installation is found.*

In this dialog, there are two options:

- **Direct download**. Download the license directly from CLC bio. This method requires that the Workbench has access to the external network.
- **Go to license download web page**. In a browser window, show the license download web page, which can be used to download a license file. This option is suitable in situations where, for example, you are working behind a proxy, so that the Workbench does not have direct access to the CLC Licenses Service.

If you select the option to download a license directly and it turns out that the Workbench does not have direct access to the external network, (because of a firewall, proxy server etc.), you can click **Previous** button to try the other method.

After selection on 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**, the dialog shown in figure 1.15 appears.

A progress for getting the license is shown, and when the license is downloaded, you will be able to click **Next**.

### Go to license download web page

After choosing the *Go to license download web page* option and clicking on the button labeled **Next**, the license download web page appears in a browser window, as shown in 1.16.

Click the **Request Evaluation License** button. You can then save the license on your system.

Back in the Workbench window, you will now see the dialog shown in 1.17.

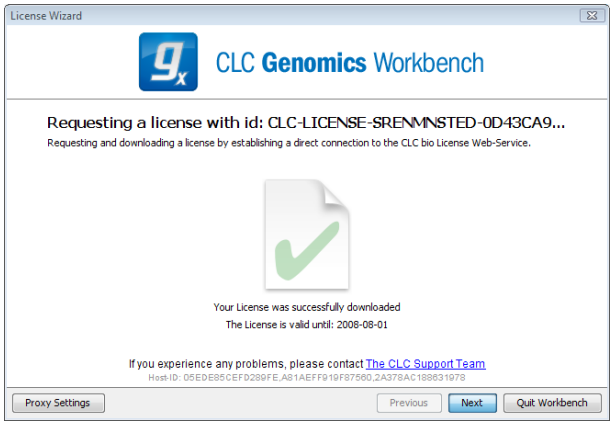



Figure 1.15: *A license has been downloaded.*

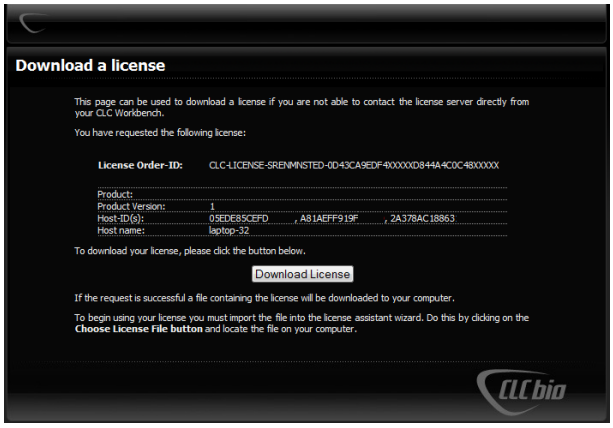



Figure 1.16: *The license download web page.*

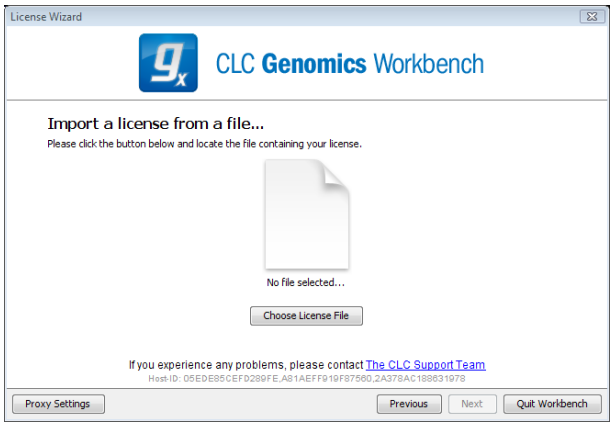



Figure 1.17: *Importing the license file downloaded from the web page.*

Click the **Choose License File** button and browse to find the license file you saved. When you have selected the file, click on the button labeled **Next**.

### Accepting the license agreement

Part of the installation of the license involves checking and accepting the end user license agreement (EULA). You should now see the a window like that in figure 1.18.

Please read the EULA text carefully before clicking in the box next to the text **I accept these**

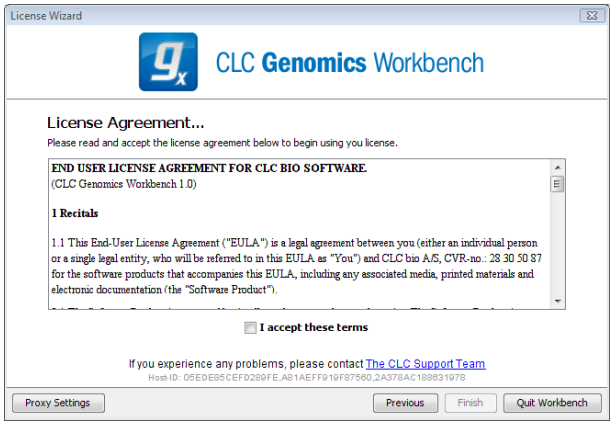



Figure 1.18: *Read the license agreement carefully.*

**terms** to accept, and then clicking on the button labeled **Finish**.

### 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 do this, select this option and click on the **Next** button. A dialog like that shown in figure 1.19 then appears. Here, you configure how to connect to the CLC License Server.

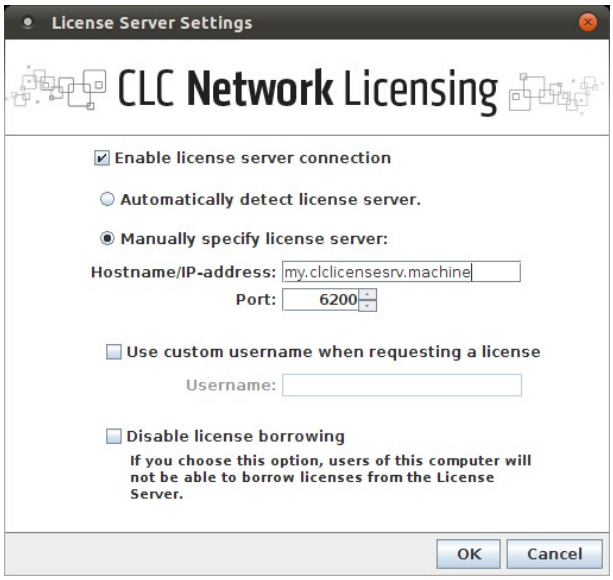



Figure 1.19: *Connecting to a CLC License Server.*

- **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[3].

[3]Automatic server discovery sends UDP broadcasts from the Workbench on a fixed port, 6200. Available license servers respond to the broadcast. The Workbench then uses TCP communication for to get a license, assuming one

- **Manually specify license server**. If there are technical limitations such that the CLC License Server cannot be detected automatically, use this option to provides details of machine the CLC License Server software is on, and the port used by the software to receive requests. After selecting this option, please enter:
  - **Host name**. The address for the machine the CLC Licenser Server software is running on.
  - **Port**. The port used by the CLC License Server to receive requests.
- **Disable license borrowing on this computer**. If you do not want users of the computer to borrow a license from the set of licenses available, then (see section 1.4.5), select this option.

### Borrowing a license

A network license can only be used when you are connected to the a license server. If you wish to use the *CLC Genomics Workbench* when you are not connected to the CLC License Server, you can *borrow* an available license for a period of time. During this time, there will be one less network license available on the for other users. The Workbench must have a connection to the CLC License Server at the point in time when you wish to borrow a license.

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 1.20.
3. Use the checkboxes at the right hand sideof the table in the License overview section of the window to select the license(s) that you wish to borrow.
4. Select the length of time you wish to borrow the license(s).
5. Click on the button labeled **Borrow Licenses**.
6. Close the License Manager when you are done.

You can now go offline and work with the *CLC Genomics Workbench*. When the time period you borrowed the license for has elapsed, the network license you borrowed is made available again for other users to access. To continue using the *CLC Genomics Workbench* with a license, you will need to connect the Workbench to the network again so it can contact the CLC Licene Server to obtain one.

**Note!** Your CLC License Server administrator can choose to disable to the option allowing the borrowing of licenses. If this has been done, you will not be able to borrow a network license using your Workbench.

---

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 manually instead.

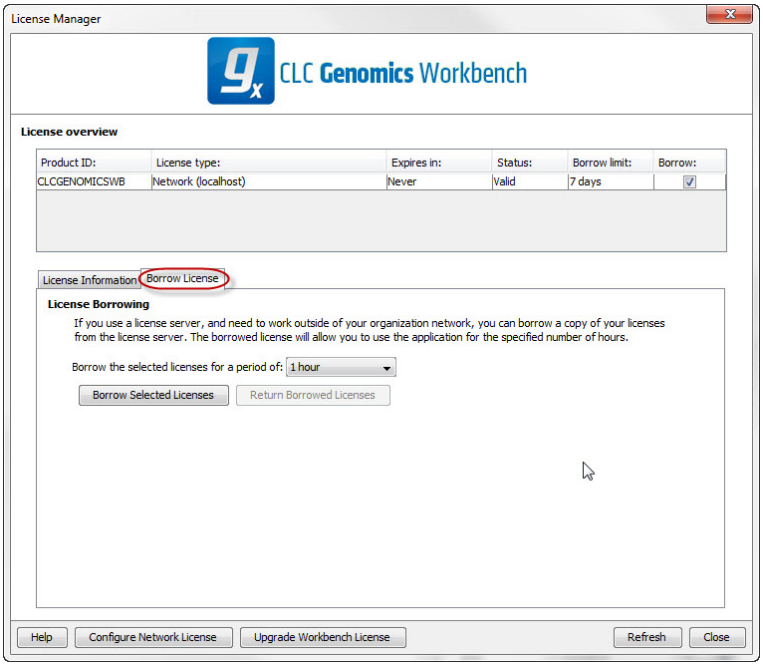



Figure 1.20: *Borrow a license.*

### Common issues when using a network license

**No license available at the moment** If all the network licenses or *CLC Genomics Workbench*are in use, you will see a dialog like that shown in figure 1.21 when you start up the Workbench.

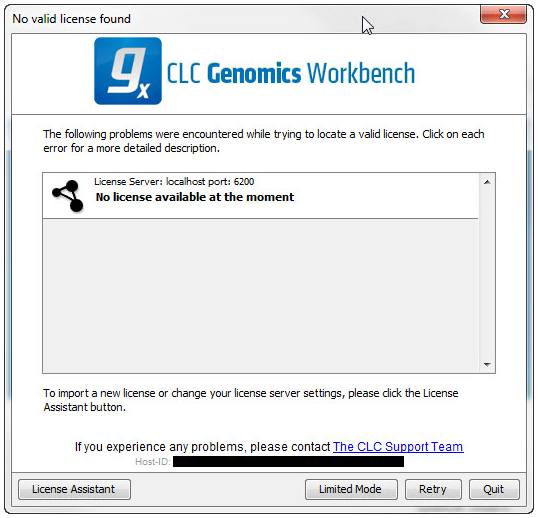



Figure 1.21: *This window appears when there are no available network licenses for the software you are running.*

This means others are using the network licenses. You will need to wait for them to return their licenses before you can continue to work with a fully functional copy of software. If this is a frequent issue, you may wish to discuss this with your CLC License Server administrator.

Clicking on the **Limited Mode** button in the dialog allows you to start the Workbench with a subset of features available. This includes the ability your access CLC data.

**Lost connection to the CLC License Server** If the Workbench connection to the CLC License Server is lost, you will see a dialog as shown in figure 1.22.

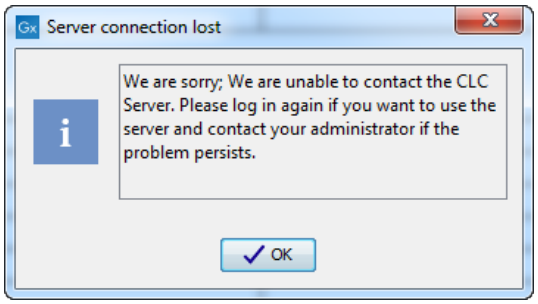



Figure 1.22: *This message appears if the Workbench is unable to establish a connection to a CLC License server.*

If you have chosen the option to **Automatically detect license server** and you have not succeeded in connecting to the 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 possible at your site, you will need to manually configure the CLC License Server settings using the License Manager, 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, to make sure that the CLC License Server is running and that your Workbench can 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 1.23.

This dialog can be used to:

- See information about the license (e.g. what kind of license, when it expires)
- Configure how to connect to a license server (**Configure License Server** the button at the lower left corner). Clicking this button will display a dialog similar to figure 1.19.
- Upgrade from an evaluation license by clicking the **Upgrade license** button. This will display the dialog shown in figure 1.1.
- Export license information to a text file.
- Borrow a license

If you wish to switch away from using a network license, click on the button to **Configure License Server** 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:

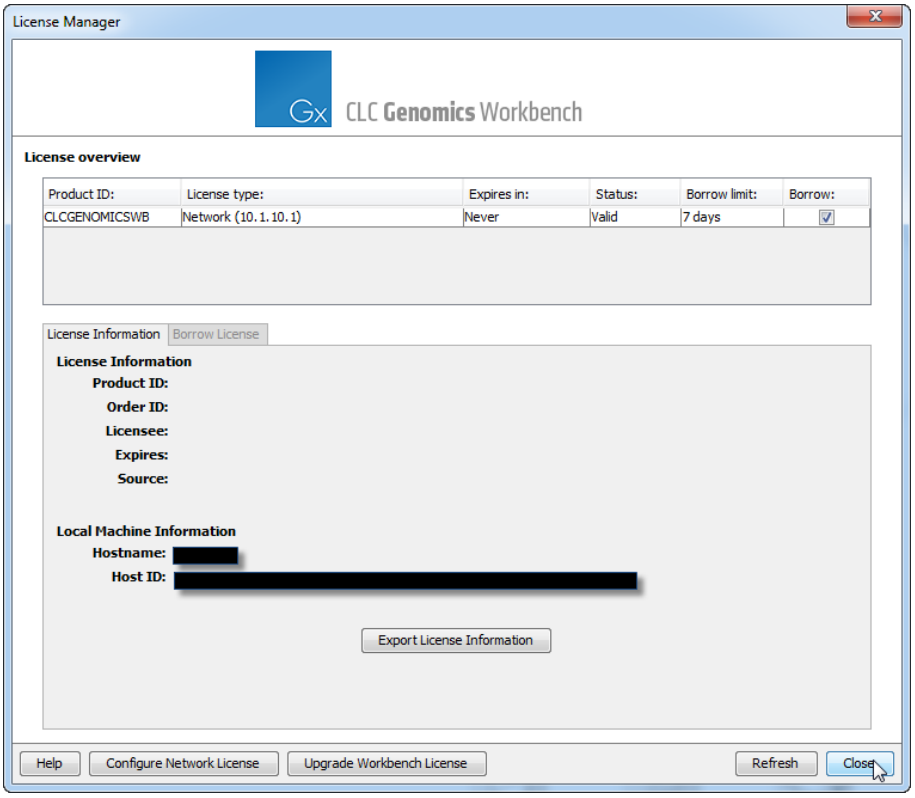



Figure 1.23: *The license manager.*

- 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 the machine reported at the bottom of the License Manager window in grey text.
- 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 relevant network license download web page:
- For Workbenches released from January 2013 and later, (e.g. the Genomics Workbench version 6.0 or higher, and the Main Workbench, version 6.8 or higher), please go to:

  `https://secure.clcbio.com/LmxWSv3/GetLicenseFile`

  For earlier Workbenches, including any DNA, Protein or RNA Workbench, please go to:

  `http://licensing.clcbio.com/LmxWSv1/GetLicenseFile`

  It is vital that you choose the license download page appropriate to the version of the software you plan to run.
- Paste in your license order ID and the host ID that you noted down in the relevant boxes on the webpage.
- Click '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 'choose license file' to browse the location of 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 Help menu and choosing License Manager.

- Click on the **Next** button and go through the remaining steps of the license manager wizard.

### 1.4.7 Limited mode

We have created the limited mode to prevent a situation where you are unable to access your data because you do not have a license. When you run in limited mode, a lot of the tools in the Workbench are not available, but you still have access to your data (also when stored in a *CLC Bioinformatics Database*) . When running in limited mode, the functionality is equivalent to the *CLC Sequence Viewer* (see section A). To get out of the limited mode and run the Workbench normally, restart the Workbench. When you restart the Workbench will try to find a proper license and if it does, it will start up normally. If it can't find a license, you will again have the option of running in limited mode.

## 1.5 About CLC Workbenches

In November 2005 CLC bio released 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. However, the *CLC Protein Workbench* includes a range of more advanced analyses.

In March 2006, *CLC DNA Workbench* (formerly *CLC Gene Workbench*) and *CLC Main Workbench* were 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 it has additional advanced features. *CLC Main Workbench* holds all basic and advanced features of the *CLC Workbenches*.

In June 2007, *CLC RNA Workbench* was 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, the *CLC Free Workbench* changed name to *CLC Sequence Viewer*.

In June 2008, the first version of the *CLC Genomics Workbench* was 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.

For an overview of which features all the applications include, see http://www.clcbio.com/features.

In December 2006, CLC bio released 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* were discontinued. All customers with a valid license for any of these products were offered an upgrade to the *CLC Main Workbench*.

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

### 1.5.1 New program feature request

The CLC team is continuously improving the *CLC Genomics Workbench* with our users' 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. To contact us via the Workbench, please go to the menu option:

**Help** | **Contact Support**

### 1.5.2 Getting help

If you encounter a problem or need help understanding how the *CLC Genomics Workbench* works, and the license you are using is covered by our Maintenance, Upgrades and Support (MUS) program (https://www.clcbio.com/support/maintenance-support-program/), you can contact our customer support via the Workbench by going to the menu option:

**Help** | **Contact Support**

This will open a dialog to enter your contact information and a text field for entering the question or problem you have.

You can also attach small datasets, if this helps explain the problem or you believe it will help in troubleshooting the problem.

When you send a support request this way, it will include technical information about your installation that usually helps when troubleshooting. It also includes your license information so that you do not have to look this up yourself. Our support staff will reply to you by email.

Further information about Maintenance, Upgrades and Support (MUS) program can be found online at https://www.clcbio.com/support/maintenance-support-program/.

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.

#### 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.

### 1.5.3 CLC Sequence Viewer vs. Workbenches

The *CLC Sequence Viewer* is a user friendly application offering basic bioinformatics analyses. The *CLC Sequence Viewer* can be used to view outputs from many analyses of the CLC commercial workbenches, with notable exceptions being workflows and track-based data, which can only be viewed using our commercial Workbench offerings.

Track-based outputs can be viewed using the *CLC Genomics Workbench* and *CLC Main Workbench*, while workflows can be viewed in all commercial CLC workbenches, including the *CLC Main Workbench*, *CLC Genomics Workbench* and the *CLC Drug Discovery Workbench*.

The CLC Workbenches and the *CLC Sequence Viewer* are developed for Windows, Mac and Linux platforms. Data can be exported/imported between the different platforms in the same easy way as when exporting/importing between two computers with e.g. Windows.

## 1.6 When the program is installed: Getting started

*CLC Genomics Workbench* includes an extensive **Help** function, which can be found in the **Help** menu of the program's **Menu bar**. The **Help** can also be shown by pressing F1. The help topics are sorted in a table of contents and the topics can be searched.

Tutorials describing hands-on examples of how to use the individual tools and features of the *CLC Genomics Workbench* can be found at `http://www.clcbio.com/support/tutorials/`. We also recommend our **Online presentations** where a product specialist from CLC bio 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://www.clcbio.tv/`.

### 1.6.1 Quick start

When the program opens for the first time, the background of the workspace is visible. In the background are three quick start shortcuts, which will help you getting started. These can be seen in figure 1.24.




Figure 1.24: *Quick start short cuts, available in the background of the workspace.*

The function of the quick start shortcuts is explained here:

- **Import data.** Opens the **Import** dialog, which you let you browse for, and import data from your file system.
- **New sequence.** Opens a dialog which allows you to enter your own sequence.

- **Read tutorials.** Opens the tutorials menu with a number of tutorials. These are also available from the **Help** menu in the **Menu bar**.

### 1.6.2 Import of example data

It might be easier to understand the logic of the program by trying to do simple operations on existing data. Therefore *CLC Genomics Workbench* includes an example data set.

When downloading *CLC Genomics Workbench* you are asked if you would like to import the example data set. If you accept, the data is downloaded automatically and saved in the program. If you didn't download the data, or for some other reason need to download the data again, you have two options:

You can click **Import Example Data** ( ) in the **Help** menu of the program. This imports the data automatically. You can also go to `http://www.clcbio.com/download` and download the example data from there.

If you download the file from the website, you need to import it into the program. See chapter 6 for more about importing data.

## 1.7 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 `http://www.clcbio.com/plugins` for a full list of plugins with descriptions of their functionalities.

### 1.7.1 Installing plugins

Plugins are installed using the plugin manager[4]:

**Help in the Menu Bar** | **Plugins and Resources... ( )**

or **Plugins ( ) in the Toolbar**

The plugin manager has three 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 CLC bio's server.
- **Manage Resources.** This is an overview of resources that are installed.

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 1.25).

Clicking a plugin will display additional information at the right side of the dialog. This will also display a button: **Download and Install**.

[4] In order to install plugins on Windows, the Workbench must be run in administrator mode: Right-click the program shortcut and choose "Run as Administrator". Then follow the procedure described below.

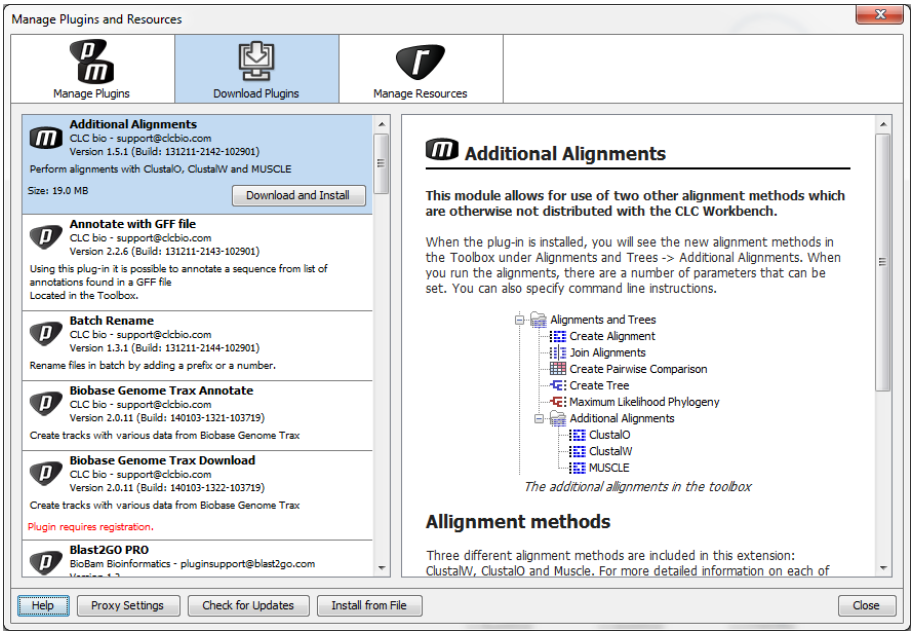



Figure 1.25: *The plugins that are available for download.*

Click the plugin and press **Download and Install**. A dialog displaying progress is now shown, and the plugin is downloaded and installed.

If the plugin is not shown on the server, and you have it on your computer (e.g. if you have downloaded it from our web-site), you can install it by clicking the **Install from File** button at the bottom of the dialog. This will open a dialog where you can browse for the plugin. The plugin file should be a file of the type ".cpa".

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

### 1.7.2 Uninstalling plugins

Plugins are uninstalled using the plugin manager:

**Help in the Menu Bar** | **Plugins and Resources... (** **)**

or **Plugins (** **) in the Toolbar**

This will open the dialog shown in figure 1.26.

The installed plugins are shown in this dialog. To uninstall:

**Click the plugin** | **Uninstall**

If you do not wish to completely uninstall the plugin but you don't 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.

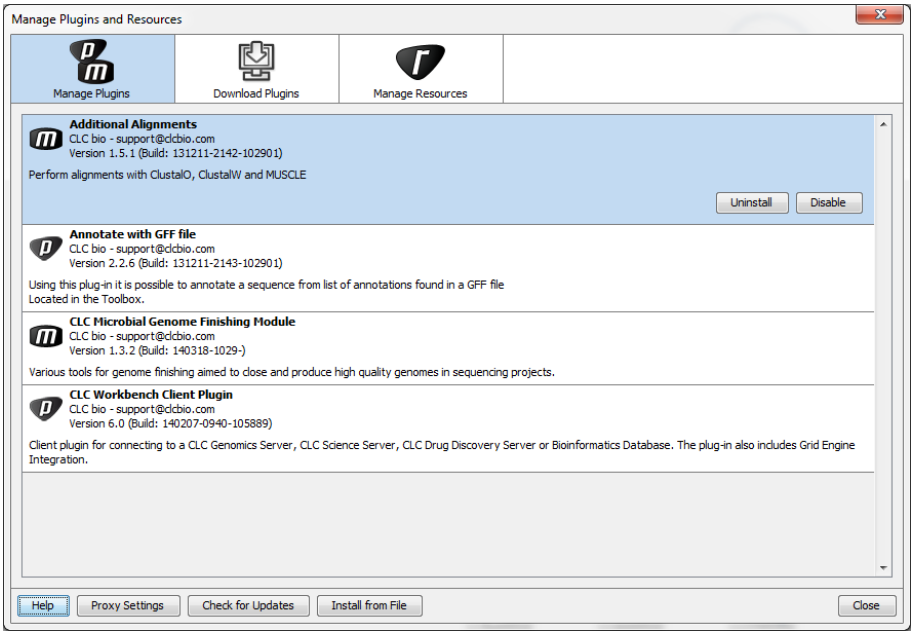



Figure 1.26: *The plugin manager with plugins installed.*

### 1.7.3 Updating plugins

If a new version of a plugin is available, you will get a notification during start-up as shown in figure 1.27.

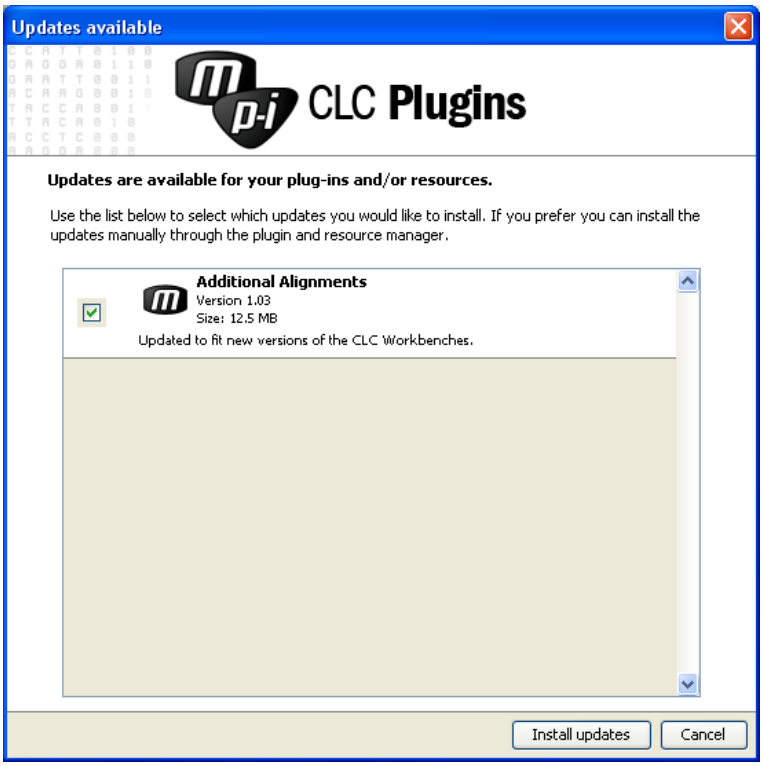



Figure 1.27: *Plugin updates.*

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

manager (see figure 1.26).

### 1.7.4 Resources

Resources can be manually installed and un-installed the same way as plugins.

## 1.8 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 (e.g. searching GenBank).

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

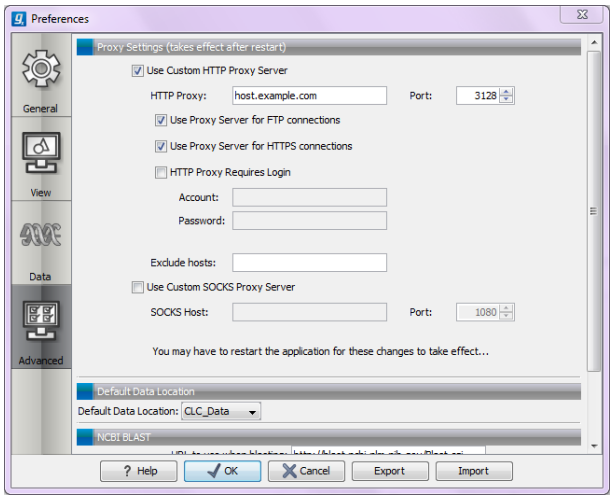



Figure 1.28: *Adjusting proxy preferences.*

To configure your proxy settings, open *CLC Genomics Workbench*, and go to the **Advanced**-tab of the **Preferences** dialog (figure 1.28) and enter the appropriate information. The **Preferences** dialog is opened from the **Edit** menu.

You have the choice between an HTTP-proxy and a SOCKS-proxy. *CLC Genomics 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.

## 1.9 The format of the user manual

This user manual offers support to Windows, Mac OS X and Linux users. The software is very similar on these operating systems. In areas where differences exist, these will be described separately. However, the term "right-click" is used throughout the manual, but some Mac users may have to use Ctrl+click in order to perform a "right-click" (if they have a single-button mouse).

The most recent version of the user manuals can be downloaded from http://www.clcbio.com/usermanuals.

The user manual consists of four parts.

- The **first part** includes the introduction to the *CLC Genomics Workbench*.
- The **second part** describes in detail how to operate all the program's basic functionalities.
- The **third part** digs deeper into some of the molecular modeling and bioinformatic features of the program. In this part, you will also find our "Bioinformatics explained" sections. These sections elaborate on the algorithms and analyses of *CLC Genomics Workbench* and provide more general knowledge of molecular modeling and bioinformatic concepts.
- The **fourth part** is the Appendix and Index.

Each chapter includes a short table of contents.

### 1.9.1 Text formats

In order to produce a clearly laid-out content in this manual, different formats are applied:

- A feature in the program is in bold starting with capital letters. ( Example: **Navigation Area**)
- An explanation of how a particular function is activated, is illustrated by "|" and bold. (E.g.: **select the element** | **Edit** | **Rename**)

## 1.10 Latest improvements

*CLC Genomics Workbench* is under constant development and improvement. A detailed list that includes a description of new features, improvements, bugfixes, and changes for the current version of *CLC Genomics Workbench* can be found at:

http://www.clcbio.com/products/latest-improvements/.

# Part II

# Core Functionalities

# Chapter 2

# User interface

**Contents**


**2.1 View Area . . . 44**
2.1.1 Open view . . . 45
2.1.2 Show element in another view . . . 45
2.1.3 Close views . . . 46
2.1.4 Save changes in a view . . . 46
2.1.5 Undo/Redo . . . 47
2.1.6 Arrange views in View Area . . . 47
2.1.7 Moving a view to a different screen . . . 50
2.1.8 Side Panel . . . 50
**2.2 Zoom and selection in View Area . . . 52**
2.2.1 Zoom in . . . 53
2.2.2 Zoom out . . . 53
2.2.3 Selecting, panning and zooming . . . 54
**2.3 Toolbox and Status Bar . . . 54**
2.3.1 Processes . . . 54
2.3.2 Toolbox . . . 55
2.3.3 Status Bar . . . 57
**2.4 Workspace . . . 57**
2.4.1 Create Workspace . . . 57
2.4.2 Select Workspace . . . 58
2.4.3 Delete Workspace . . . 58
**2.5 List of shortcuts . . . 58**


This chapter provides an overview of the different areas in the user interface of *CLC Genomics Workbench*. As can be seen from figure 2.1 this includes a **Navigation Area**, **View Area**, **Menu Bar**, **Toolbar**, **Status Bar** and **Toolbox**.

A description of the **Navigation Area** is tightly connected to the data management features of *CLC Genomics Workbench* and can be found in section 3.1.

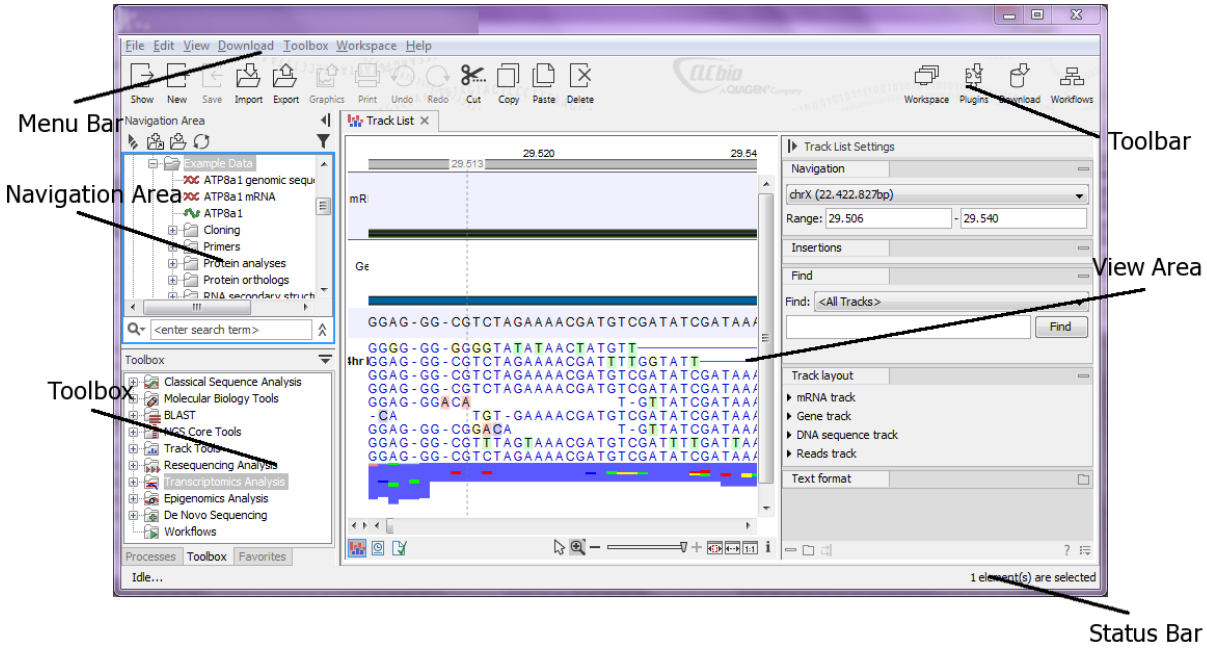



Figure 2.1: *The user interface consists of the Menu Bar, Toolbar, Status Bar, Navigation Area, Toolbox, and View Area.*

## 2.1 View Area

The **View Area** is the right-hand 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**.

This is illustrated in figure 2.2.

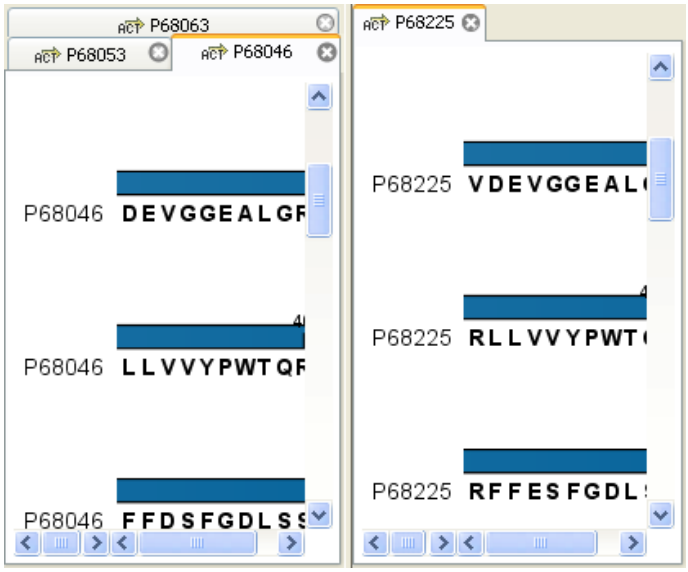



Figure 2.2: *A View Area can enclose several views, each view is indicated with a tab (see right view, which shows protein P68225). Furthermore, several views can be shown at the same time (in this example, four views are displayed).*

The tab concept is central to working with *CLC Genomics Workbench*, because several operations can be performed by dragging the tab of a view, and extended right-click menus can be activated

from the tabs.

This chapter deals with the handling of views inside a **View Area**. Furthermore, it deals with rearranging the views.

Section 2.2 deals with the zooming and selecting functions.

### 2.1.1 Open view

Opening a view can be done in a number of ways:

**double-click an element in the Navigation Area**

or **select an element in the Navigation Area** | **File** | **Show** | **Select the desired way to view the element**

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

Opening a view while another view is already open, will show the new view in front of the other view. The view that was already open can be brought to front by clicking its tab.

**Note!** If you right-click an open tab of any element, click **Show**, and then choose a different view of the same element, this new view is automatically opened in a split-view, allowing you to see both views.

See section 3.1.5 for instructions on how to open a view using drag and drop.

### 2.1.2 Show element in another view

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

In the following example, you want to see a sequence in a circular view. If the sequence is already open in a view, you can change the view to a circular view:

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

The buttons used for switching views are shown in figure 2.3).



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 e.g. a circular view or a history view.*

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 10.5).

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 ( )**

### 2.1.3 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 of the View | Close**

or **select the view | Ctrl + W**

or **hold down the Ctrl-button | Click the tab of the view while the button is pressed**

By right-clicking a tab, the following close options exist. See figure 2.4

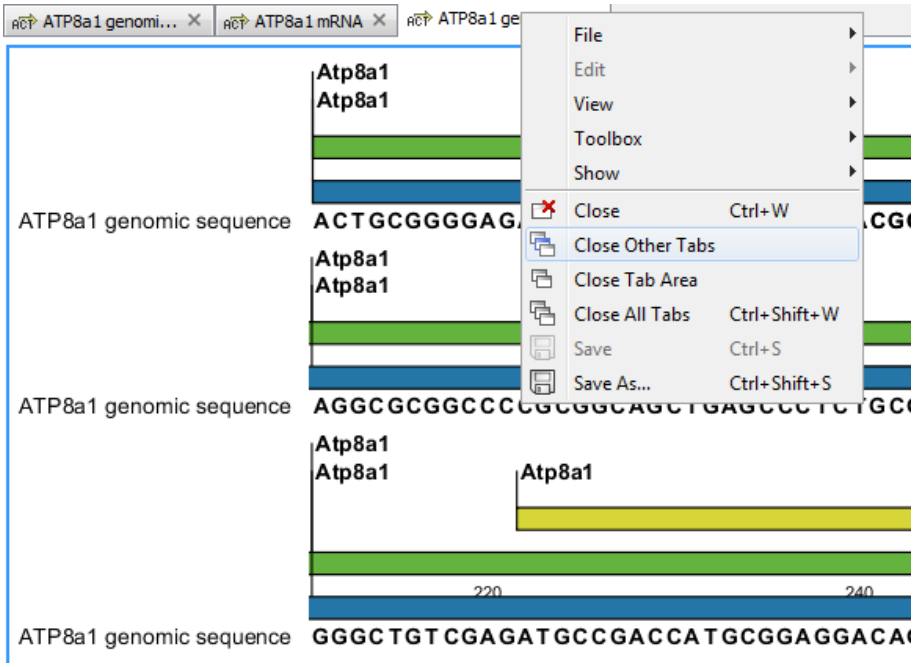



Figure 2.4: *By right-clicking a tab, several close options are available.*

- **Close.** See above.
- **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.
- **Close All Tabs.** Closes all tabs, in all tab areas. Leaves an empty workspace.

### 2.1.4 Save changes in a view

When changes to an element are made in a view, the text on the tab appears *bold and italic* (on Mac it is indicated by an * before the name of the tab). This indicates that the changes are not saved. The **Save** function may be activated in two ways:

**Click the tab of the view you want to save | Save ( ) in the toolbar.**

or **Click the tab of the view you want to save | Ctrl + S (⌘ + S on Mac)**

If you close a tab of a view containing an element that has been changed since you opened it, you are asked if you want to save.

When saving an element from a new view that has not been opened from the **Navigation Area** (e.g. when opening a sequence from a list of search hits), a save dialog appears (figure 2.5).

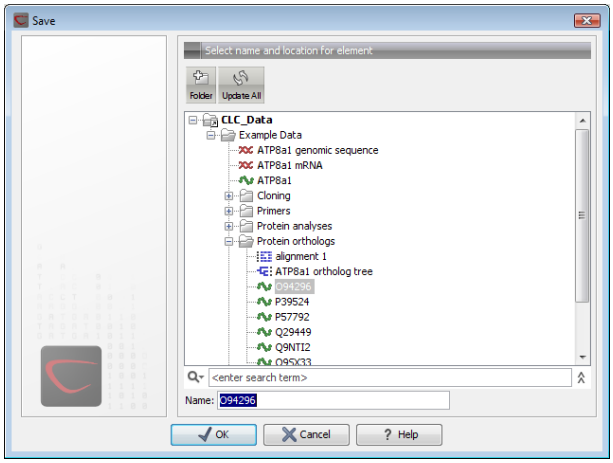



Figure 2.5: *Save dialog.*

In the dialog you select the folder in which you want to save the element.

After naming the element, press **OK**

### 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 **Edit | Undo ( )**

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 **Edit | Redo ( )**

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 the **Toolbox** by clicking the hide icon ( ) at the top of the **Navigation Area**.

**Views** are arranged in the **View Area** by their tabs. The order of the **views** can be changed using drag and drop. E.g. drag the tab of one view onto the tab of a another. The tab of the first view is

now placed at the right side of the other tab.

If a tab is dragged into a view, an area of the view is made gray (see fig. 2.6) illustrating that the view will be placed in this part of the **View Area**.

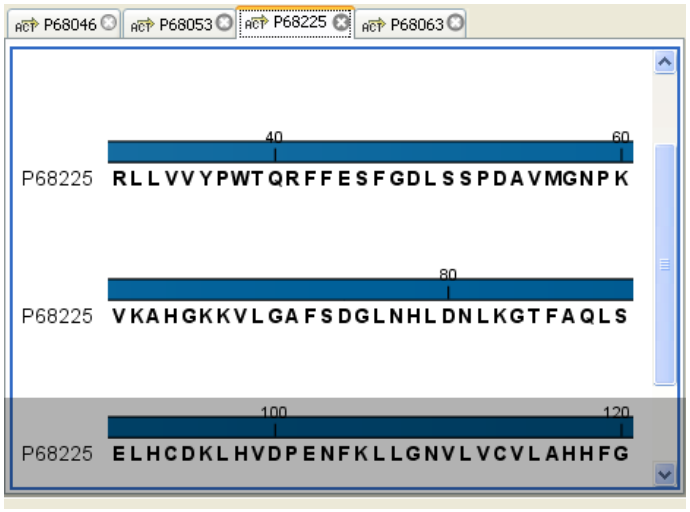



Figure 2.6: *When dragging a view, a gray area indicates where the view will be shown.*

The results of this action is illustrated in figure 2.7.

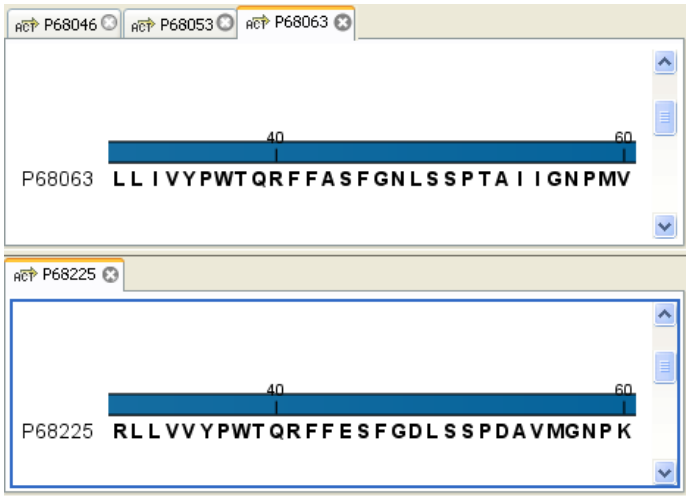



Figure 2.7: *A horizontal split-screen. The two views split the View Area.*

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. (See figure 2.8). When the split is made vertically, the new view opens to the right of the existing view.

Splitting the **View Area** can be undone by dragging e.g. the tab of the bottom view to the tab of the top view. This is marked by a gray area on the top of the view.

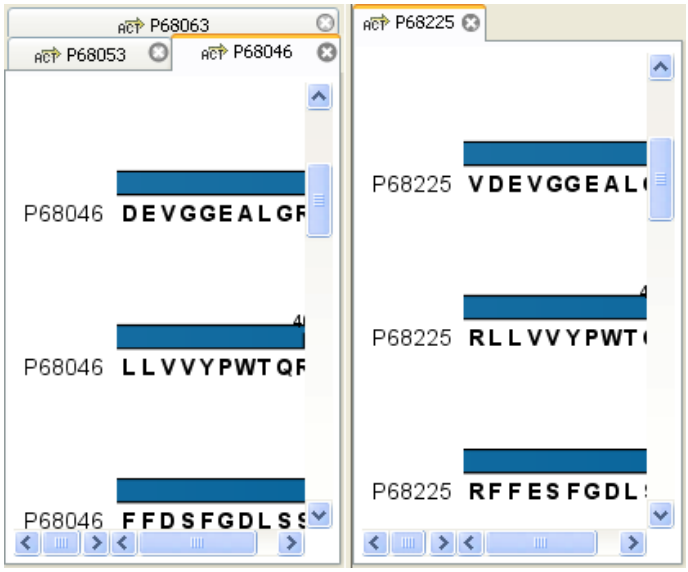



Figure 2.8: *A vertical split-screen.*

### Maximize/Restore size of view

The **Maximize/Restore View** function allows you to see a view in maximized mode, meaning a mode where no other **views** nor the **Navigation Area** is shown.

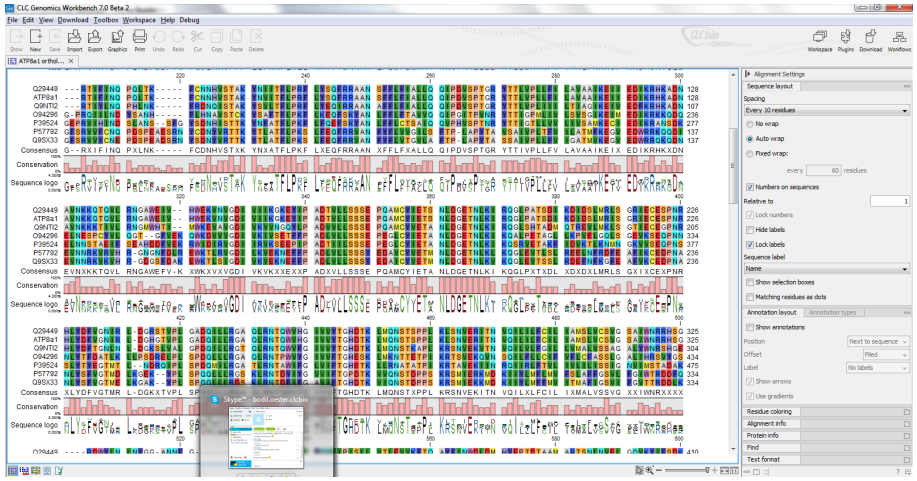


Figure 2.9: *A maximized view. The function hides the Navigation Area and the Toolbox.*

Maximizing a view can be done in the following ways:

**select view** | **Ctrl + M**

or **select view** | **View** | **Maximize/restore View (▭)**

or **select view** | **right-click the tab** | **View** | **Maximize/restore View (▭)**

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

The following restores the size of the view:

**Ctrl + M**

or **View** | **Maximize/restore View (▭)**

or **double-click title of view**

Please note that you can also hide **Navigation Area** and the **Toolbox** by clicking the hide icon ( ) at the top of the **Navigation Area**

### 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 2.10, where the main Workbench window shows a table of open reading frames, and the screen to the right is used to display the sequence and annotations.

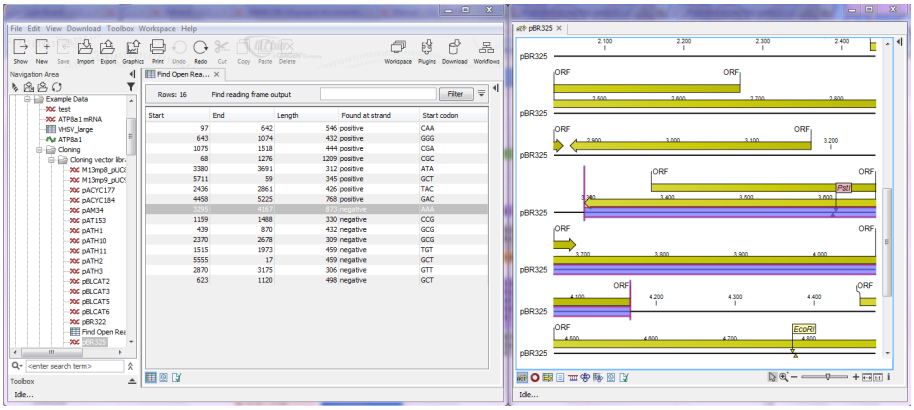


Figure 2.10: *Showing the table on one screen while the sequence is displayed on another screen. Clicking the table of open reading frames causes the view on the other screen to follow the selection. Note that the screen resolution in this figure is kept low in order to include it in the manual; in a real scenario, the resolution will be much higher.*

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.

You can also split the view area in the detached windows as described in section 2.1.6.

### 2.1.8 Side Panel

The **Side Panel** allows you to change the way the contents of a view are 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 2.11 shows the default **Side Panel** for a protein sequence. It is organized into *palettes*.

In this example, there is one 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.

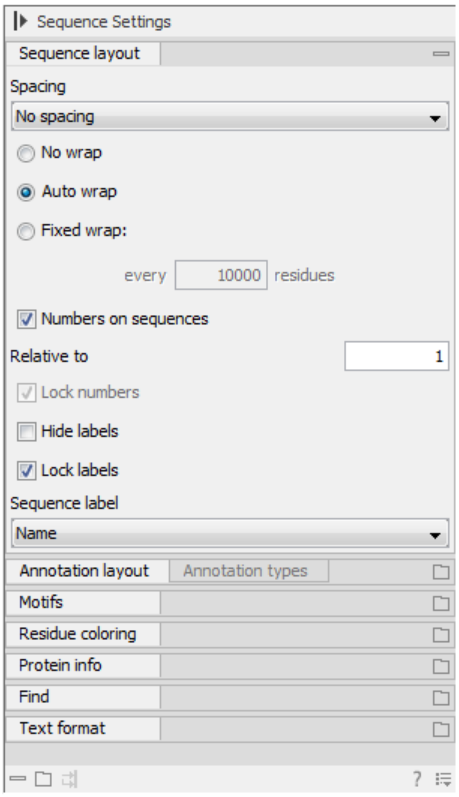



Figure 2.11: *The default view of the Side Panel when opening a protein sequence.*

In addition, they can be moved away from the **Side Panel** and placed anywhere on the screen as shown in figure2.12.

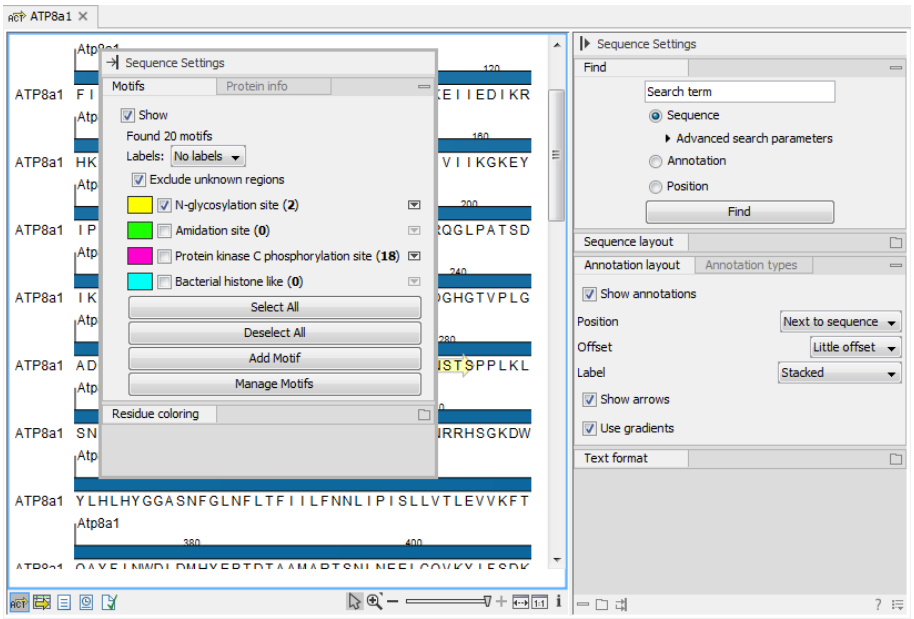



Figure 2.12: *Palettes can be organized in the Side Panel as you like or placed anywhere on the screen.*

In this example, the **Motifs** palette has been placed on top of the sequence view together with the **Protein info** and the **Residue coloring** palettes. 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 2.13) there are a number of icons used to:

- Expand all settings ( ).
- Collapse 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. Read more in section 4.6

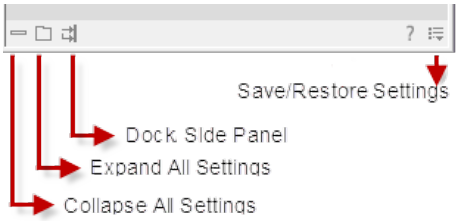



Figure 2.13: *Controlling the Side Panel at the bottom*

**Note!** Changes made to the **Side Panel**, including the organization of palettes will not be saved when you save the view. See how to save the changes in section 4.6

## 2.2 Zoom and selection in View Area

All views except tabular and text views support zooming. Figure 2.14 shows the zoom tools, located at the bottom right corner of the view.

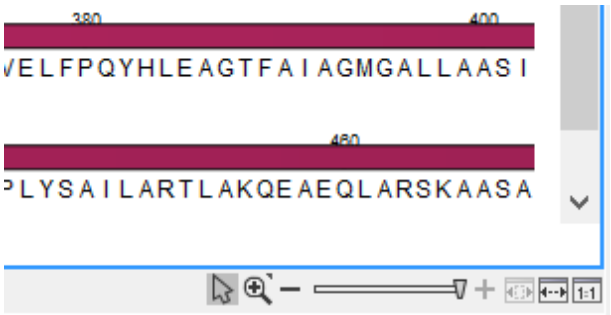



Figure 2.14: *The zoom tools are located at the bottom right corner of the view.*

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 13.2.

### 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 all the data, 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 2.15:

- **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 2.15: *Additional mouse modes can be found in the zoom tools.*

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 Status Bar

The **Toolbox** is placed in the left side of the user interface of *CLC Genomics Workbench* below the **Navigation Area**.

The **Toolbox** shows a **Processes tab**, **Favorites tab** and a **Toolbox tab**.

The **Toolbox** can be hidden, so that the **Navigation Area** is enlarged and thereby displays more elements:

**View** | **Show/Hide Toolbox** | **Show/Hide Toolbox**

You can also click the **Hide Toolbox** ( ) button.

### 2.3.1 Processes

By clicking the **Processes** tab, the **Toolbox** displays previous and running processes, e.g. an NCBI search or a calculation of an alignment. The running processes can be stopped, paused, and resumed by clicking the small icon ( ) next to the process (see figure 2.16).

Running and paused processes are not deleted.

Besides the options to stop, pause and resume processes, there are some extra options for *a selected number* of the tools running from the Toolbox:

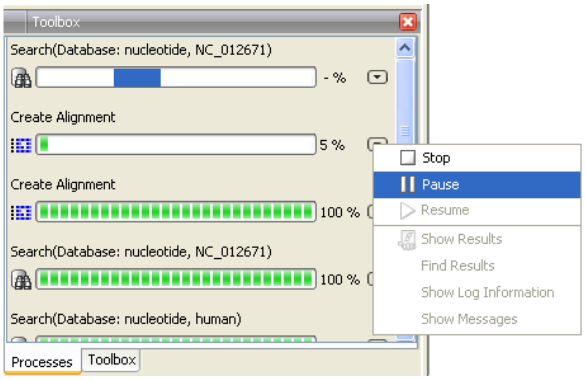



Figure 2.16: *A database search and an alignment calculation are running. Clicking the small icon next to the process allow you to stop, pause and resume processes.*

- **Show results**. If you have chosen to save the results (see section 8.2), you will be able to open the results directly from the process by clicking this option.
- **Find results**. If you have chosen to save the results (see section 8.2), you will be able to high-light the results in the Navigation Area.
- **Show Log Information**. This will display a log file showing progress of the process. The log file can also be shown by clicking **Show Log** in the "handle results" dialog where you choose between saving and opening the results.
- **Show Messages**. Some analyses will give you a message when processing your data. The messages are the black dialogs shown in the lower left corner of the Workbench that disappear after a few seconds. You can reiterate the messages that have been shown by clicking this option.

The terminated processes can be removed by:

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

If you close the program while there are running processes, a dialog will ask if you are sure that you want to close the program. Closing the program will stop the process, and it cannot be restarted when you open the program again.

### 2.3.2 Toolbox

The content of the **Toolbox** tab in the **Toolbox** corresponds to **Toolbox** in the **Menu Bar**.

The tools in the toolbox can be accessed by double-clicking or by dragging elements from the **Navigation Area** to an item in the **Toolbox**.

#### Quick access to tools

To enable quick launch of tools in *CLC Genomics Workbench*, press Ctrl + Shift + T (⌘ + Shift + T on Mac) to show the quick launch dialog (see figure 2.17).

When the dialog is opened, you can start typing search text in the text field at the top. This will bring up the list of tools that match this text either in the name, description or location in the

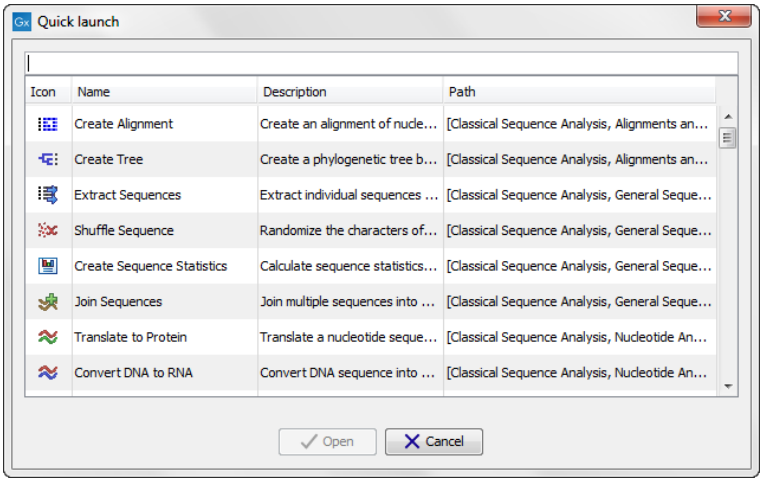



Figure 2.17: *Quick access to all tools in* **CLC Genomics Workbench**.

Toolbox. In the example shown in figure 2.18, typing `create` shows a list of tools involving the word "create", and the arrow keys or mouse can be used for selecting and starting a tool.

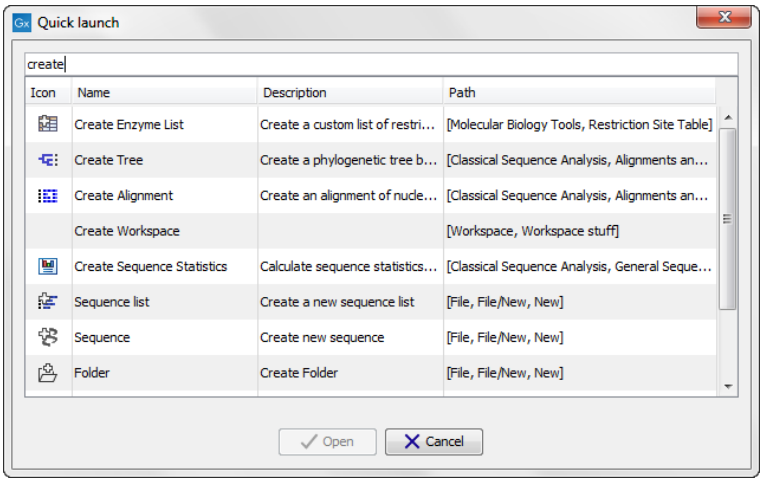



Figure 2.18: *Typing in the search field at the top will filter the list of tools to launch.*

### Favorites toolbox

Next to the **Toolbox** tab, you find the **Favorites** tab. This can be used for organizing and getting quick access to the tools you use the most. It consists of two parts as shown in figure 2.19.

**Favorites** You can manually add tools to the favorites menu simply by right-clicking the tool in the Toolbox. You can also right-click the Favorites folder itself and select **Add Tool**. To remove a tool, right-click and select **Remove from Favorites**. Note that you can also add complete folders to the favorites.

**Frequently used** The list of tools in this folder is automatically populated as you use the Workbench. The most frequently used tools are listed at the top.

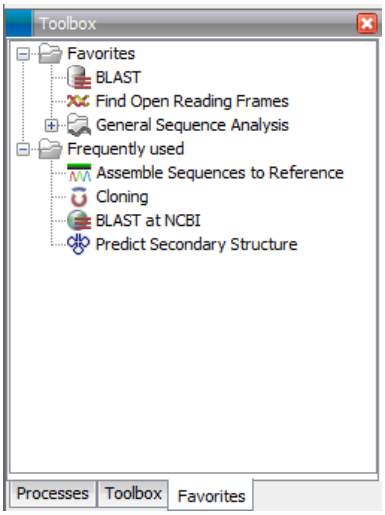



Figure 2.19: *Favorites toolbox.*

### 2.3.3 Status Bar

As can be seen from figure 2.1, the **Status Bar** is located at the bottom of the window. In the left side of the bar is an indication of whether the computer is making calculations or whether it is idle. The right side of the **Status Bar** indicates the range of the selection of a sequence. (See chapter 2.2.3 for more about the Selection mode button.)

## 2.4 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 **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.

### 2.4.1 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**

When the new **Workspace** is created, the heading of the program frame displays the name of the new **Workspace**. Initially, the selected elements in the **Navigation Area** is collapsed and the **View Area** is empty and ready to work with. (See figure 2.20).

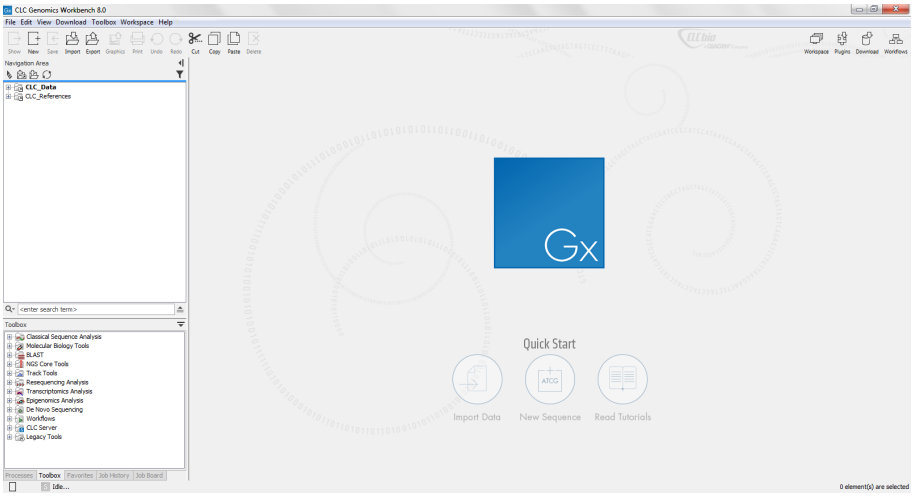



Figure 2.20: *An empty Workspace.*

### 2.4.2 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**

The name of the selected **Workspace** is shown after "*CLC Genomics Workbench*" at the top left corner of the main window, in figure 2.20 it says: (default).

### 2.4.3 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.

## 2.5 List of shortcuts

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

| Action | Windows/Linux | Mac OS X |
|---|---|---|
| Adjust selection | Shift + arrow keys | Shift + arrow keys |
| Adjust workflow layout | Shift + Alt + L | ⌘ + Shift + Alt + L |
| 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 |
| Maximize/restore size of View | 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 |
| Reverse zoom mode | press and hold Shift | press and hold Shift |
| 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 |
| Start Tool Quick Launch | Ctrl + Shift + T | ⌘ + Shift + T |
| Translate to Protein | Ctrl + Shift + P | ⌘ + Shift + P |
| Undo | Ctrl + Z | ⌘ + Z |
| Update folder | F5 | F5 |
| User Preferences | Ctrl + K | ⌘ + ; |

| Action | Windows/Linux | Mac OS X |
|---|---|---|
| Vertical scroll in read 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 | Alt + Scroll wheel | Alt + Scroll wheel |
| 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 |
| Zoom | Ctrl + Scroll wheel | Ctrl + 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 |

Combinations of keys and mouse movements are listed below.

| Action | Windows/Linux | Mac OS X | 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 that are not grouped together | Ctrl | ⌘ | Click elements |
| Select multiple elements that are grouped together | Shift | Shift | Click elements |

"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 . . . 62**
3.1.1 Data structure . . . 62
3.1.2 Create new folders . . . 64
3.1.3 Sorting folders . . . 65
3.1.4 Multiselecting elements . . . 65
3.1.5 Moving and copying elements . . . 65
3.1.6 Change element names . . . 67
3.1.7 Delete, restore and remove elements . . . 68
3.1.8 Show folder elements in a table . . . 68
**3.2 Customized attributes on data locations . . . 70**
3.2.1 Configuring which fields should be available . . . 70
3.2.2 Editing lists . . . 71
3.2.3 Removing attributes . . . 71
3.2.4 Changing the order of the attributes . . . 72
**3.3 Filling in values . . . 72**
3.3.1 What happens when a clc object is copied to another data location? . . . 74
3.3.2 Searching . . . 74
**3.4 Local search . . . 74**
3.4.1 What kind of information can be searched? . . . 75
3.4.2 Quick search . . . 75
3.4.3 Advanced search . . . 78
3.4.4 Search index . . . 80


This chapter explains the 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.

## 3.1 Navigation Area

The **Navigation Area** is located in the left side of the screen, under the **Toolbar** (see figure 3.1). It is used for organizing and navigating data. Its behavior is similar to the way files and folders are usually displayed on your computer.

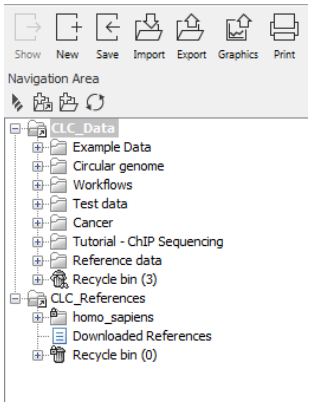



Figure 3.1: *The Navigation Area.*

To provide more space for viewing data, you can hide **Navigation Area** and the **Toolbox** by clicking the hide icon ( ) at the top.

### 3.1.1 Data structure

The data in the **Navigation Area** is organized into a number of **Locations**. When the *CLC Genomics Workbench* is started for the first time, there is one location called *CLC_Data* (unless your computer administrator has configured the installation otherwise).

A location represents a folder on the computer: The data shown under a location in the **Navigation Area** is stored on the computer in the folder which the location points to.

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

#### Adding locations

Per default, there is one location in the **Navigation Area** called CLC_Data. It points to the following folder:

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

You can easily add more locations to the **Navigation Area**:

**File** | **New** | **Location ( )**

This will bring up a dialog where you can navigate to the folder you wish to use as your new location (see figure 3.4).

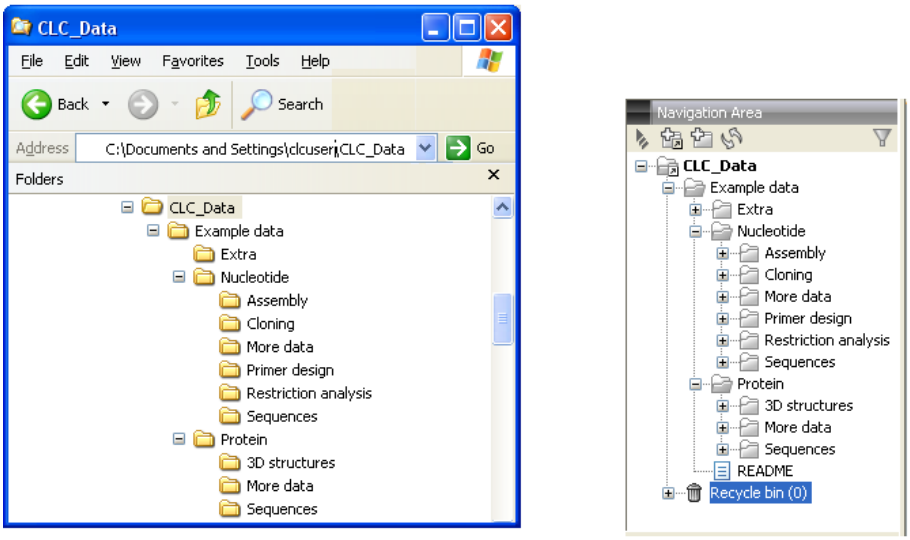



Figure 3.2: *In this example the location called 'CLC_Data' points to the folder at C:\Documents and settings\clcuser\CLC_Data.*

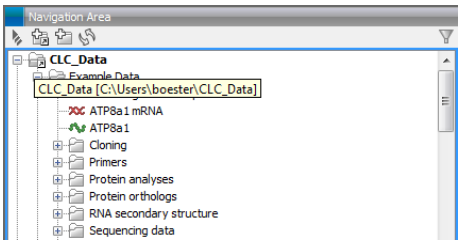



Figure 3.3: *Mousing over the location called 'CLC_Data' shows the full path to the system folder, which in this case is C:\Users\boester\CLC_Data.*

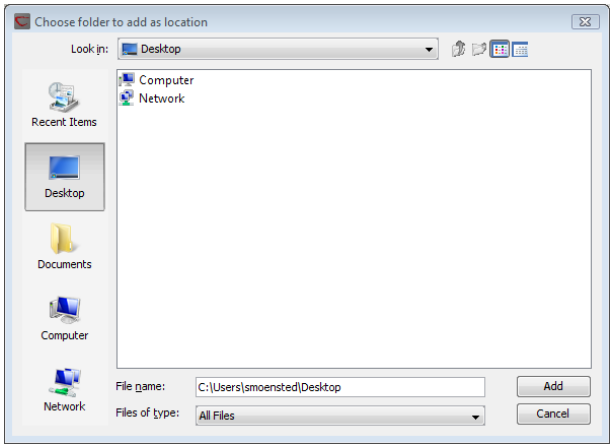



Figure 3.4: *Navigating to a folder to use as a new location.*

When you click **Open**, the new location is added to the **Navigation Area** as shown in figure 3.5.

The name of the new location will be the name of the folder selected for the location. To see where the folder is located on your computer, place your mouse cursor on the location icon ( ) for a second. This will show the path to the location.

**Sharing data** is possible if you add a location on a network drive. The procedure is similar to the one described above. When you add a location on a network drive or a removable drive, the location will appear *inactive* when you are not connected. Once you connect to the drive again,

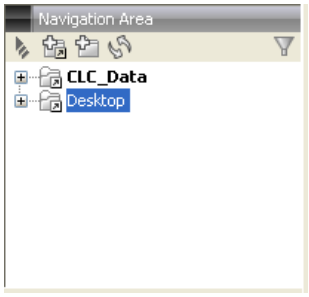



Figure 3.5: *The new location has been added.*

click **Update All** ( ) and it will become active (note that there will be a few seconds' delay from you connect).

#### 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 **Clicking once on the element** | **Right-click on the element** | **Show ( )**

or **Clicking once on the element** | **Right-click on the element** | **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** (), **Cut** () and **Paste** () from the **Edit** menu.
- Using Ctrl + C (⌘ + C on Mac), Ctrl + X (⌘ + X on Mac) and Ctrl + V (⌘ + V on Mac).
- Using **Copy** (), **Cut** () and **Paste** () in the **Toolbar**.
- Using drag and drop to move elements.
- Using drag and drop while pressing Ctrl / Command to copy elements.

In the following, all of these possibilities for moving and copying elements are described in further detail.

#### Copy, cut and paste functions

Copies of elements and folders can be made with the copy/paste function which can be applied in a number of ways:

**select the files to copy** | **right-click one of the selected files** | **Copy (  )** | **right-click the location to insert files into** | **Paste (  )**

or **select the files to copy** | **Ctrl + C (⌘ + C on Mac)** | **select where to insert files** | **Ctrl + P (⌘ + P on Mac)**

or **select the files to copy** | **Edit in the Menu Bar** | **Copy (  )** | **select where to insert files** | **Edit in the Menu Bar** | **Paste (  )**

If there is already an element of that name, the pasted element will be renamed by appending a number at the end of the name.

Elements can also be moved instead of copied. This is done with the cut/paste function:

**select the files to cut** | **right-click one of the selected files** | **Cut (  )** | **right-click the location to insert files into** | **Paste (  )**

or **select the files to cut** | **Ctrl + X (⌘ + X on Mac)** | **select where to insert files** | **Ctrl + V (⌘ + V on Mac)**

When you have cut the 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 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.

#### Move using drag and drop

Using drag and drop in the **Navigation Area**, as well as in general, is a four-step process:

**click the element** | **click on the element again, and hold left mouse button** | **drag the element to the desired location** | **let go of mouse button**

This allows you to:

- Move elements between different folders in the **Navigation Area**
- Drag from the **Navigation Area** to the **View Area**: A new view is opened in an existing **View Area** if the element is dragged from the **Navigation Area** and dropped next to the tab(s) in that **View Area**.
- Drag from the **View Area** to the **Navigation Area**: The element, e.g. a sequence, alignment, search report etc. is saved where it is dropped. If the element already exists, you are asked whether you want to save a copy. You drag from the **View Area** by dragging the tab of the desired element.

Use of drag and drop is supported throughout the program, also to open and re-arrange views (see section 2.1.6).

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.

#### Copy using drag and drop

To copy instead of move using drag and drop, hold the Ctrl (⌘ on Mac) key while dragging:

**click the element** | **click on the element again, and hold left mouse button** | **drag the element to the desired location** | **press Ctrl (⌘ on Mac) while you let go of mouse button release the Ctrl/⌘ button**

### 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 10.3.3.

### 3.1.7 Delete, restore and remove elements

When one deletes data from a data folder in the Workbench, it is moved to the recycle bin in 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:

**right-click the element** | **Delete (** **)**

or **select the element** | **press Delete key**

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 10.3.4.

**Items in a recycle bin can be restored** in two ways:

Drag the elements with the mouse into the folder where they used to be.

or **select the element** | **right click and choose 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:

**Edit in the Menu Bar** | **Empty 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.

### 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 3.6.

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.

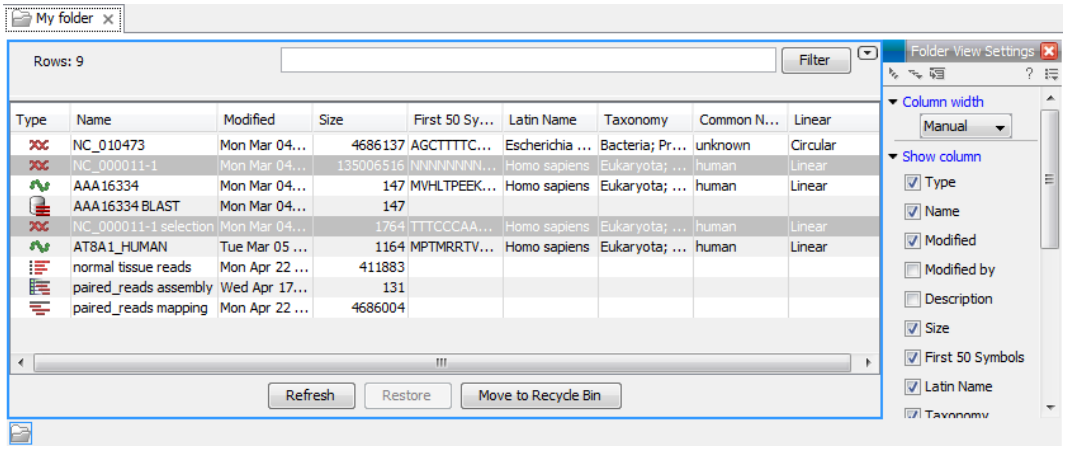



Figure 3.6: *Viewing the elements in a folder.*

### 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 3.7 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.

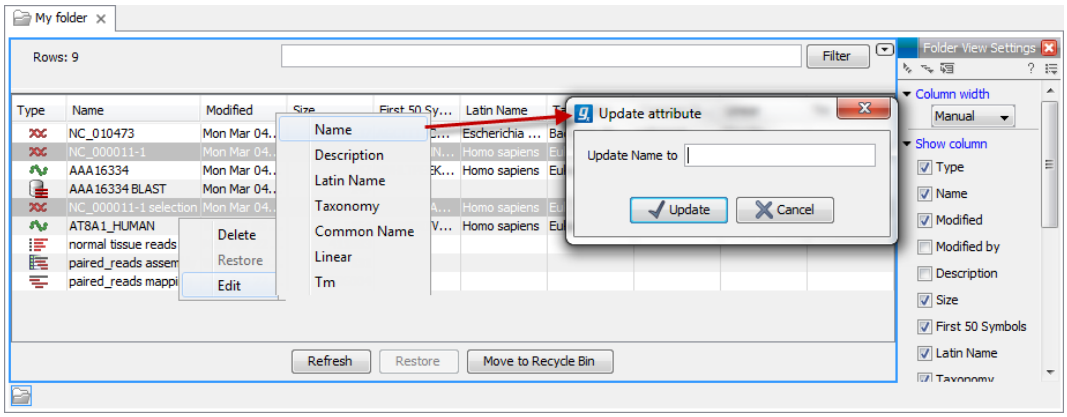



Figure 3.7: *Changing the common name of two sequences.*

**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 Customized attributes on data locations

The *CLC Genomics Workbench* makes it possible to define location-specific attributes on all elements stored in a data location. This could be company-specific information such as LIMS id, freezer position etc. Note that the attributes scheme belongs to a location, so if you have added multiple locations, they will have their own separate set of attributes.

**Note!** For *CLC Genomics Workbench*and *CLC Main Workbench*, a Metadata Import Plugin is available. The plugin consists of two tools: "Import Sequences in Table Format" and "Associate with metadata". These tools allow sequences to be imported from a tabular data source and make it possible to add metadata to existing objects.

### 3.2.1 Configuring which fields should be available

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 3.8.

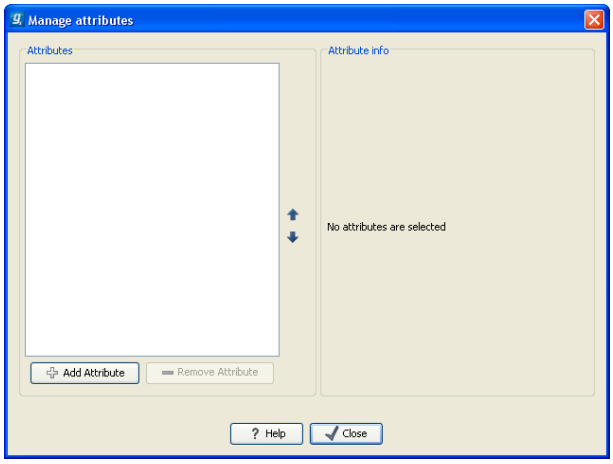



Figure 3.8: *Adding attributes.*

Click the **Add Attribute** ( ) button to create a new attribute. This will display the dialog shown in figure 3.9.

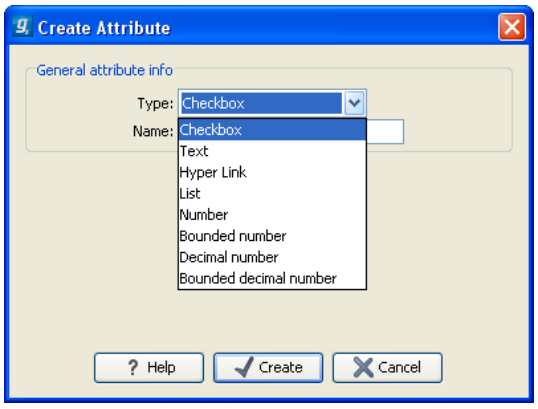



Figure 3.9: *The list of attribute types.*

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

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

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

### 3.2.2 Editing lists

Lists are a little special, since you have to define the items in the list. When you click a list 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 3.10).

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

### 3.2.3 Removing attributes

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

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

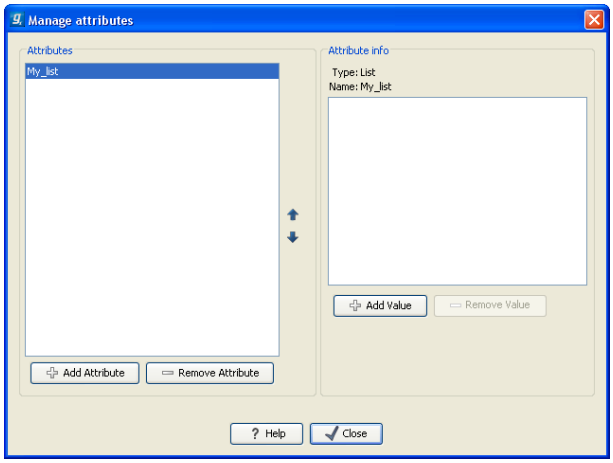



Figure 3.10: *Defining items in a list.*

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.

### 3.2.4 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 Filling in values

When a set of attributes has been created (as shown in figure 3.11), the end users can start filling in information.

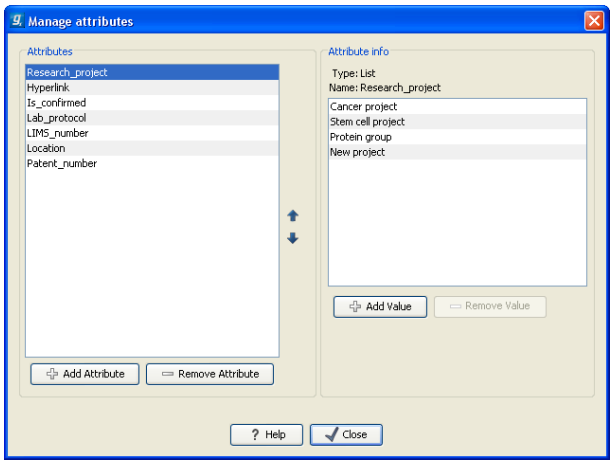



Figure 3.11: *A set of attributes defined in the attribute manager.*

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 3.12.




Figure 3.12: *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 3.13).

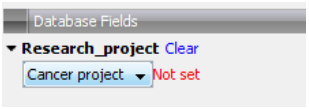



Figure 3.13: *An attribute which has not been set.*

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 3.13, 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).

### 3.3.1 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.

### 3.3.2 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 3.14).

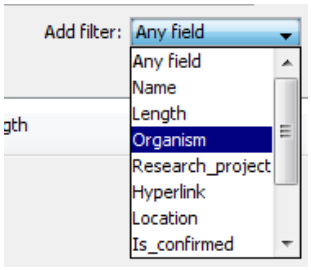



Figure 3.14: *The attributes from figure 3.11 are now listed in the search filter.*

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 3.15).

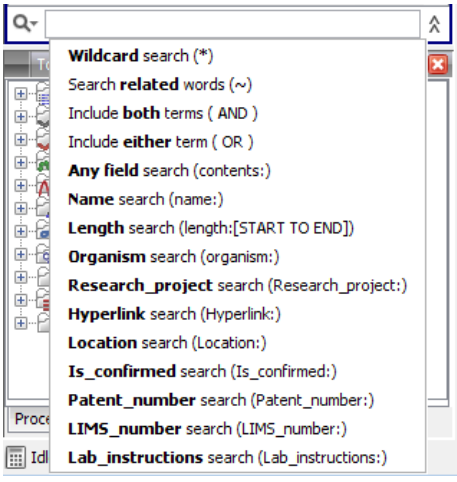



Figure 3.15: *The attributes from figure 3.11 are now available in the Quick Search as well.*

## 3.4 Local search

There are two ways of doing text-based searches of your data, as described in this section:

- **Quick-search** directly from the search field in the **Navigation Area**.
- **Advanced search** which makes it easy to make more specific searches.

In most cases, quick-search will find what you need, but if you need to be more specific in your search criteria, the advanced search is preferable.

### 3.4.1 What kind of information can be searched?

Below is a list of the different kinds of information that you can search for (applies to both quick-search and the advanced search).

- **Name.** The name of a sequence, an alignment or any other kind of element. The name is what is displayed in the **Navigation Area** per default.
- **Length.** The length of the sequence.
- **Organism.** Sequences which contain information about organism can be searched. In this way, you could search for e.g. *Homo sapiens* sequences.
- **Custom attributes.** Read more in section 3.2

Only the first item in the list, **Name**, is available for all kinds of data. The rest is only relevant for sequences.

If you wish to perform a search for sequence similarity, use Local BLAST (see section 12.1.3) instead.

### 3.4.2 Quick search

At the bottom of the **Navigation Area** there is a text field as shown in figure 3.16).

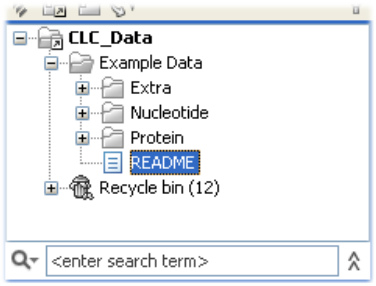



Figure 3.16: *Search simply by typing in the text field and press Enter.*

To search, simply enter a text to search for and press **Enter**.

#### Quick search results

To show the results, the search pane is expanded as shown in figure 3.17).

If there are many hits, only the 50 first hits are immediately shown. At the bottom of the pane you can click **Next** ( ) to see the next 50 hits (see figure 3.18).

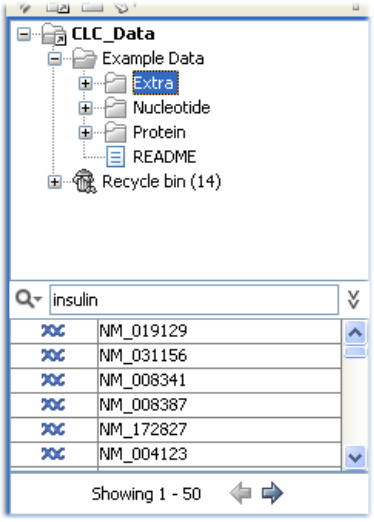



Figure 3.17: *Search results.*

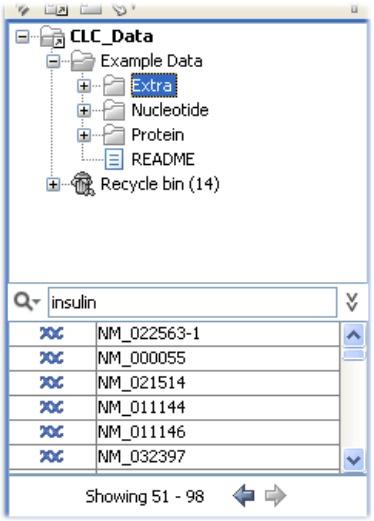



Figure 3.18: *Page two of the search results.*

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

Pressing the Alt key while you click a search result will high-light the search hit in its folder in the **Navigation Area**.

In the preferences (see Chapter 4), you can specify the number of hits to be shown.

### Special search expressions

When you write a search term in the search field, you can get help to write a more advanced search expression by pressing **Shift+F1**. This will reveal a list of guides as shown in figure 3.19.

You can select any of the guides (using mouse or keyboard arrows), and start typing. If you e.g. wish to search for sequences named BRCA1, select "Name search (name:)", and type "BRCA1". Your search expression will now look like this: "name:BRCA1".

The guides available are these:

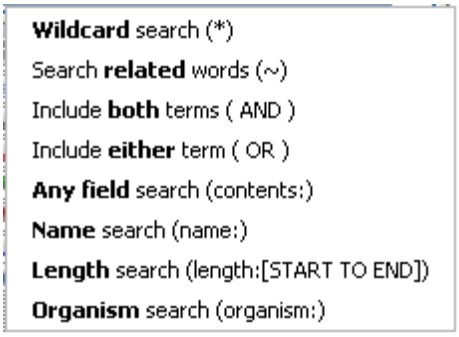



Figure 3.19: *Guides to help create advanced search expressions.*

- **Wildcard search (*)**. Appending an asterisk * to the search term will find matches starting with the term. E.g. searching for "brca*" will find both *brca1* and *brca2*.
- **Search related words ( )**. If you don't know the exact spelling of a word, you can append a question mark to the search term. E.g. "brac1*" will find sequences with a *brca1* gene.
- **Include both terms (AND)**. If you write two search terms, you can define if your results have to match both search terms by combining them with AND. E.g. search for "brca1 AND human" will find sequences where *both* terms are present.
- **Include either term (OR)**. If you write two search terms, you can define that your results have to match either of the search terms by combining them with OR. E.g. search for "brca1 OR brca2" will find sequences where *either* of the terms is present.
- **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 the name of element.
- **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 10.4).
- **Length search (length:[START TO END])**. Search for sequences of a specific length. E.g. search for sequences between 1000 and 2000 residues: "length:1000 TO 2000".

**Note!** If you have added attributes (see section 3.2), these will also appear on the list when pressing **Shift+F1**.

If you do not use this special syntax, you will automatically search for both name, description, organism, etc., and search terms will be combined as if you had put OR between them.

### Search for data locations

The search function can also be used to search for a specific URL. This can be useful if you work on a server and wish to share a data location with another user. A simple example is shown in figure 3.21. Right click on the object name in the **Navigation Area** (in this case ATP8a1 genomic sequence) and select "Copy". When you use the paste function in a destination outside the Workbench (e.g. in a text editor or in an email), the data location will become visible. The URL can now be used in the search field in the Workbench to locate the object.

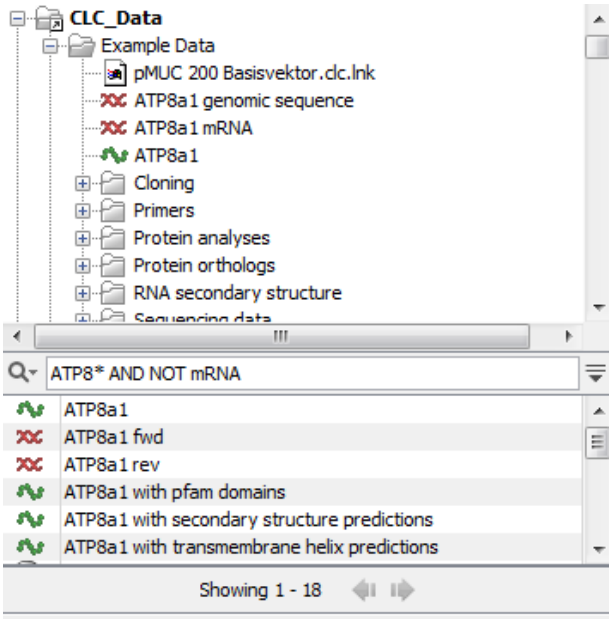



Figure 3.20: *An example of searching for elements with the name, description and organsim information that includes "ATP8" but do not include the term "mRNA".*

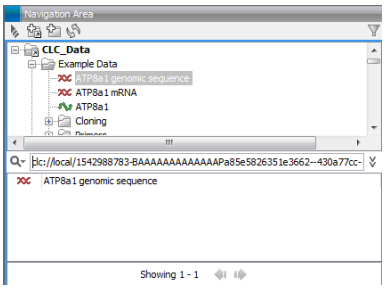



Figure 3.21: *The search field can also be used to search for data locations.*

### Quick search history

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

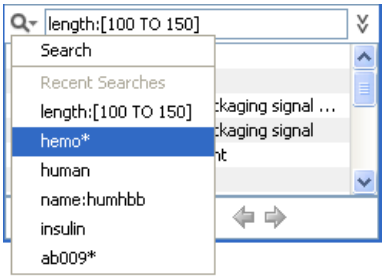



Figure 3.22: *Recent searches.*

Clicking one of the recent searches will conduct the search again.

## 3.4.3 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)**

This will open the search view as shown in figure 3.23

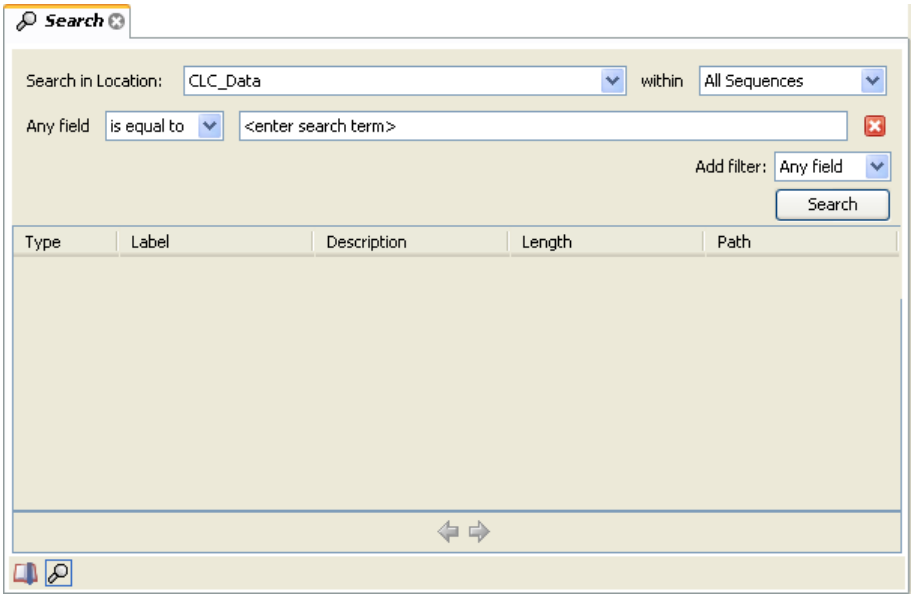



Figure 3.23: *Advanced search.*

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

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 3.4.2 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 10.4).

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 3.24.

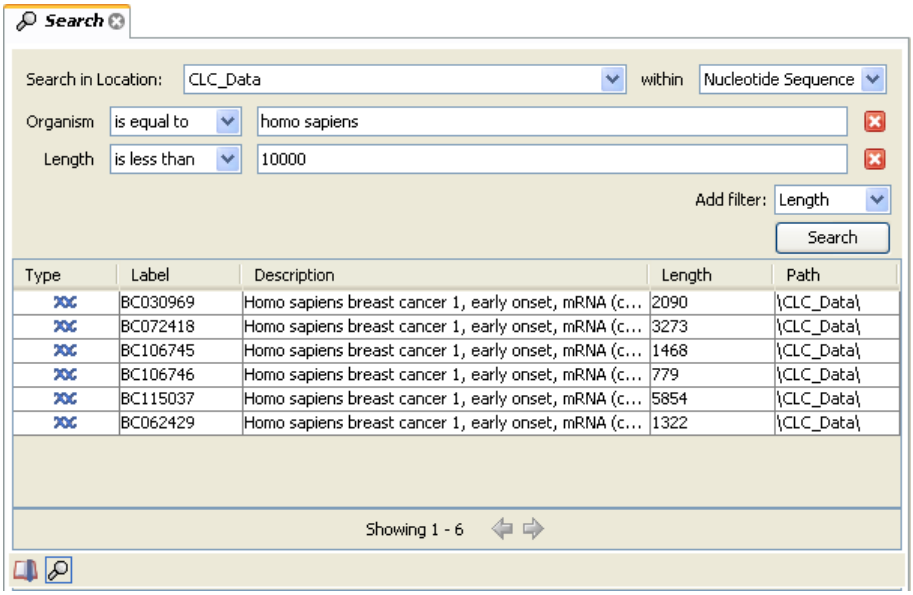



Figure 3.24: *Searching for human sequences shorter than 10,000 nucleotides.*

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.

### 3.4.4 Search index

This section has a technical focus and is not relevant if your search works fine.

However, if you experience problems with your search results: if you do not get the hits you expect, it might be because of an index error.

The *CLC Genomics Workbench* automatically maintains an index of all data in all locations in the **Navigation Area**. If this index becomes out of sync with the data, you will experience problems with strange results. In this case, you can rebuild the index:

**Right-click the relevant location** | **Location** | **Rebuild Index**

This will take a while depending on the size of your data. At any time, the process can be stopped in the process area, see section 2.3.1.

# Chapter 4

# User preferences and settings

**Contents**


**4.1 General preferences . . . . . 82**
**4.2 Default view preferences . . . . . 84**
4.2.1 Number formatting in tables . . . . . 85
4.2.2 Import and export Side Panel settings . . . . . 85
**4.3 Data preferences . . . . . 86**
**4.4 Advanced preferences . . . . . 87**
4.4.1 Default data location . . . . . 87
4.4.2 NCBI BLAST . . . . . 87
**4.5 Export/import of preferences . . . . . 87**
4.5.1 The different options for export and import . . . . . 88
**4.6 View settings for the Side Panel . . . . . 88**
4.6.1 Saving, removing and applying saved settings . . . . . 88


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)**

## 4.1 General preferences

The **General** preferences include:

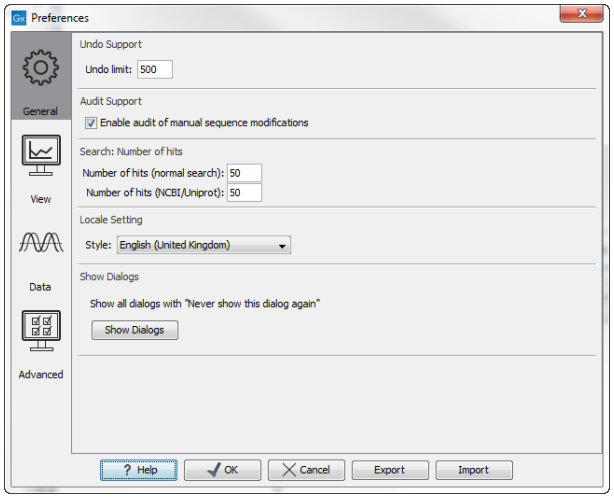



Figure 4.1: *Preferences include General preferences, View preferences, Data preferences, and Advanced settings.*

- **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 for more on this topic.
- **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** ( ) (see section 7).
- **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.




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

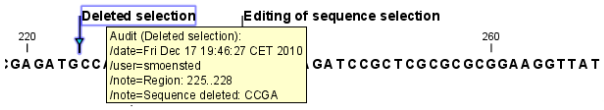



Figure 4.3: *Details of the editing.*

## 4.2 Default view preferences

There are six groups of default **View** settings:

1. **Toolbar**
2. **Show Side Panel**
3. **New View**
4. **Sequence Representation**
5. **User Defined View Settings**
6. **Molecule Project 3D Editor**

In general, these are default settings for the user interface.

The **Toolbar preferences** let you choose the size of the toolbar icons, and you can choose whether to display names below the icons.

The **Show Side Panel** setting allows you to choose whether to display the side panel.

The **New view** setting allows you to choose whether the **View preferences** are to be shown automatically when opening a new view. If this option is not chosen, you can press (Ctrl + U (⌘ + U on Mac)) to see the preferences panels of an open view.

The **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).

The **User Defined View Settings** gives you an overview of the different **Side Panel** settings that are saved for each view. See section 4.6 for 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 4.4).

In this example, the **CLC Standard Settings** is chosen as default.

The **Molecule Project 3D Editor** gives you the option to turn off the modern OpenGL rendering for **Molecule Projects** (see section 13.2.2).

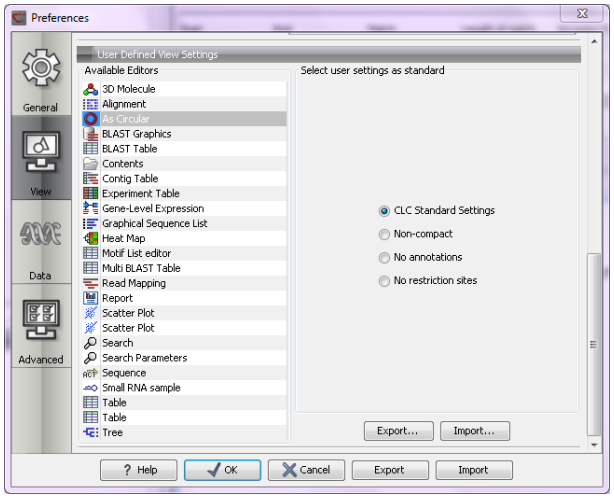



Figure 4.4: *Selecting the default view setting.*

### 4.2.1 Number formatting in tables

In the preferences, you can specify how the numbers should be formatted in tables (see figure 4.5).

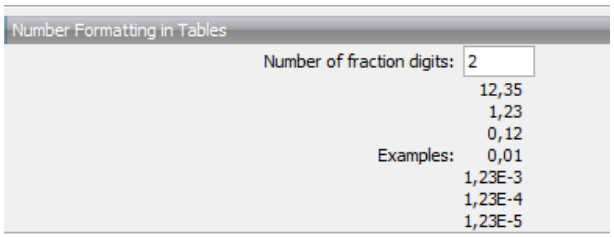



Figure 4.5: *Number formatting of tables.*

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.

### 4.2.2 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. Note that there is also another export button at the very bottom of the dialog, but this will export the other settings of the **Preferences** dialog (see section 4.5).

A dialog will be shown (see figure 4.6) that allows you to select which of the settings you wish to export.

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).

To import a **Side Panel** settings file, make sure you are at the bottom of the **View** panel of the

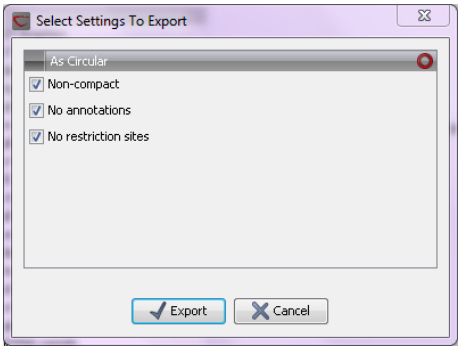



Figure 4.6: *Exporting all settings for circular views.*

**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).

The 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 4.7).

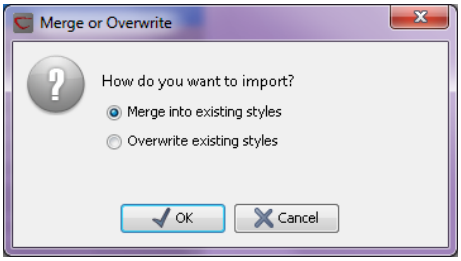



Figure 4.7: *When you import settings, you are asked if you wish to overwrite existing settings or if you wish to merge the new settings into the old ones.*

**Note!** If you choose to overwrite the existing settings, you will loose all the **Side Panel** settings that you have 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 6.5).
- 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, e.g. linker sequences:

- Linkers for importing 454 data (see section 6.3.1).

- Predefined primer additions for Gateway cloning (see section 19.2.1).
- Adapter sequences for trimming (see section 23.1.2).

## 4.4 Advanced preferences

The **Advanced** settings include the possibility to set up a proxy server. This is described in section 1.8.

### 4.4.1 Default data location

The default location is used when you e.g. import a file without selecting a folder or element in the **Navigation Area** first.

The default data location for CLC Workbenches is, by default, a folder called CLC_Data in a user's home area.

This can be changed to a different location for a particular user of the Workbench by going to

**Edit** | **Preferences**

and then choosing the **Advanced** tab. This holds a section called **Default Data Location** and here you can choose a default from a drop down list of data locations you have already added.

**Note!** The default location cannot be removed. You have to select another location as default first.

If the data area you want as your default is not already available in your Workbench, you need to first add it as a new data location (see section 3.1.1).

### 4.4.2 NCBI BLAST

#### URL to use for BLAST

It is possible to specify an alternate server URL to use for BLAST searches. The standard URL for the BLAST server at NCBI is: `http://blast.ncbi.nlm.nih.gov/Blast.cgi`.

**Note!** Be careful to specify a valid URL, otherwise BLAST will not work.

## 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 (Ctrl + K (⌘ + ; on Mac)) and do the following:

**Export** | **Select the relevant preferences** | **Export** | **Choose location for the exported file** | **Enter name of file** | **Save**

**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 4.2.2.

The process of importing preferences is similar to exporting:

> **Press Ctrl + K (⌘ + ; on Mac) to open Preferences | Import | Browse to and select the .cpf file | Import and apply preferences**

### 4.5.1 The different options for export and import

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 6.5).
- 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 *CLC Genomics Workbench* and is described in further detail in section 2.1.8.

When you have adjusted a view of e.g. a sequence, your settings in the **Side Panel** can be saved. When you open other sequences, which you want to display in a similar way, the saved settings can be applied. The options for saving and applying are available at the bottom of the **Side Panel** (see figure 4.8).

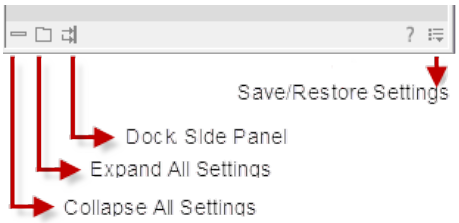



Figure 4.8: *At the bottom of the Side Panel you save the view settings*

### 4.6.1 Saving, removing and applying saved settings

To save and apply the saved settings, click ( ) seen in figure 4.8. This opens a menu where the following options are available (figure 4.9):

- **Save ... Settings.** ( ) The settings can be saved in two different ways. When you select either way of saving settings a dialog will open (see figure 4.10) where you can enter a name for your settings.

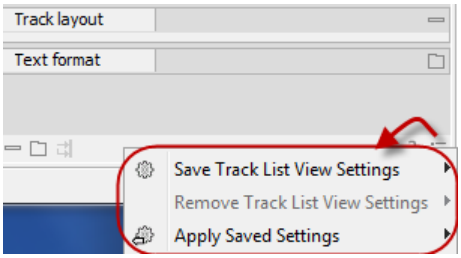



Figure 4.9: *When you have adjusted the side panel settings and would like to save these, this can be done with the "Save ... Settings" function, where "..." is the element you are working on - e.g. "Track List View", "Sequence View", "Table View", "Alignment View" etc. Saved settings can be deleted again with "Remove ... Settings" and can be applied to other elements with "Apply Saved Settings".*

- **For ... View in General** ( ) Will save the currently used settings with all elements of the same type as the one used for adjusting the settings. E.g. if you have selected to save settings "For Track View in General" the settings will be applied each time you open an element of the same type, which in this case means each time one of the saved tracks are opened from the **Navigation Area**. These "general" settings are user specific and will not be saved with or exported with the element.
- **On This Only** ( ) Settings can be saved with the specific element that you are working on in the View area and will not affect any other elements (neither in the View Area or in the **Navigation Area**). E.g. for a track you would get the option to save settings "On This Track Only". 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).

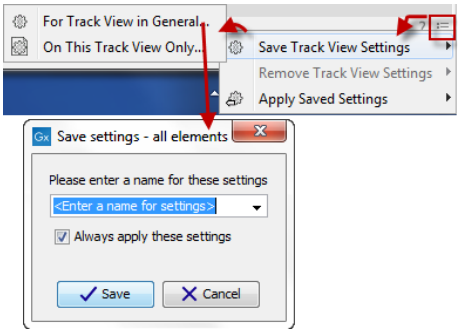



Figure 4.10: *The save settings dialog.Two options exist for saving settings. Click on the relevant option to open the dialog shown at the bottom of the figure.*

- **Remove ... Settings.** ( ) Gives you the option to remove settings specifically for the element that you are working on in the View Area, or on all elements of the same type. When you have selected the relevant option, the dialog shown in figure 4.11 opens and allows you to select which of the saved settings to remove.
  - **From ... View in General** ( ) Will remove the currently used settings on all elements of the same type as the one used for adjusting the settings. E.g. if you have selected to remove settings from all alignments using "From Alignment View in General", all alignments in your **Navigation Area** will be opened with the standard settings in stead.
  - **From This ... Only** ( ) When you select this option, the selected settings will only be removed from the particular element that you are working on in the View area and will not affect any other elements (neither in the View Area or in the **Navigation Area**).

The settings for this particular element will be replaced with the CLC standard settings ( ).

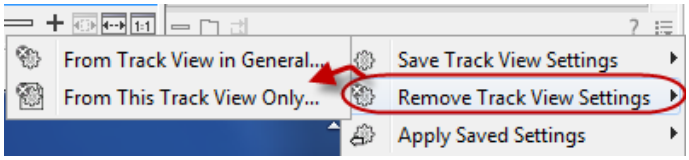



Figure 4.11: *The remove settings dialog for a track.*

- **Apply Saved Settings.** ( ) This is a submenu containing the settings that you have previously saved (figure 4.12). By clicking one of the settings, they will be applied to the current view. You will also see a number of pre-defined view settings in this submenu. They are meant to be examples of how to use the **Side Panel** and provide quick ways of adjusting the view to common usages. At the bottom of the list of settings you will see **CLC Standard Settings** which represent the way the program was set up, when you first launched it. ( )

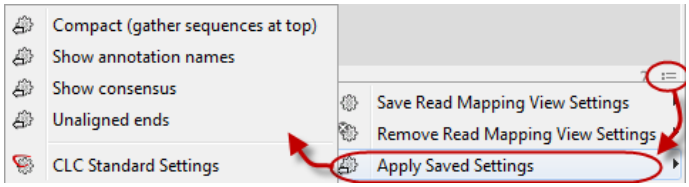



Figure 4.12: *Applying saved settings.*

The 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.

If you wish to export the settings that you have saved, this can be done in the **Preferences** dialog under the **View** tab (see section 4.2.2).

# Chapter 5

# Printing

## Contents


**5.1 Selecting which part of the view to print . . . . . 92**
**5.2 Page setup . . . . . 93**
5.2.1 Header and footer . . . . . 94
**5.3 Print preview . . . . . 94**


*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 6.5) 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.

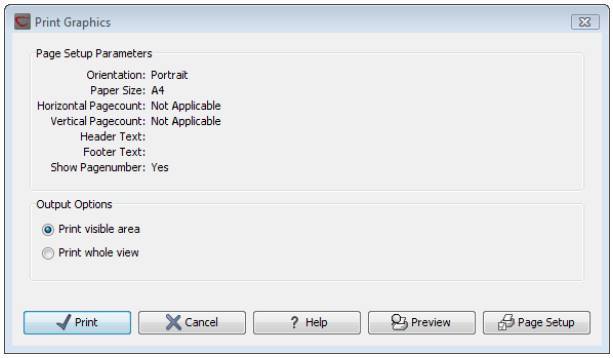



Figure 5.1: *The Print dialog.*

## 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.

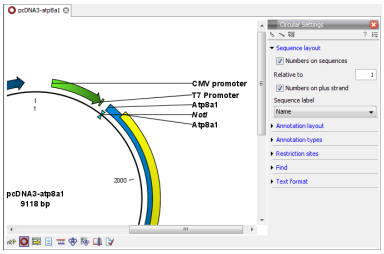



Figure 5.2: *A circular sequence as it looks on the screen.*

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.

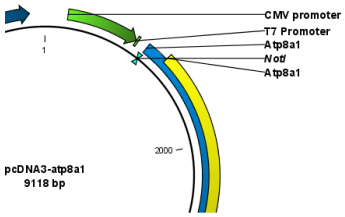



Figure 5.3: *A print of the sequence selecting Print visible area.*

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.

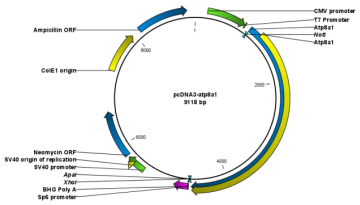



Figure 5.4: *A print of the sequence selecting Print whole view. The whole sequence is shown, even though the view is zoomed in on a part of the sequence.*

## 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

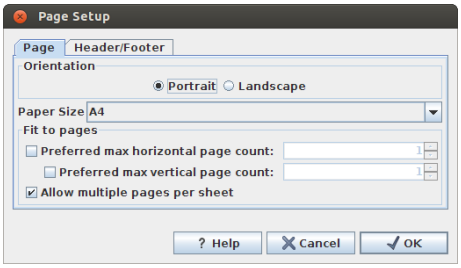



Figure 5.5: *Page Setup.*

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.
- **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.

**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**.

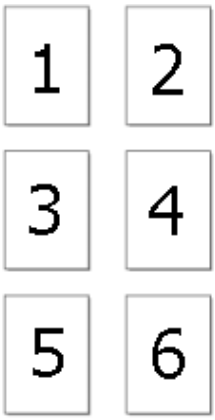



Figure 5.6: *An example where Fit to pages horizontally is set to 2, and Fit to pages vertically is set to 3.*

### 5.2.1 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.

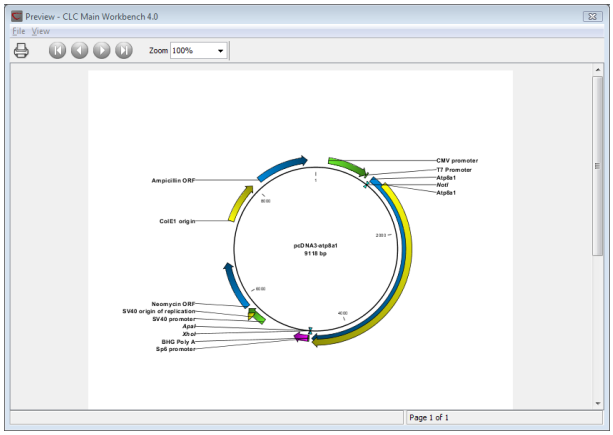



Figure 5.7: *Print preview.*

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 . . . 96**
6.1.1 Import using the import dialog . . . 96
6.1.2 Import using drag and drop . . . 97
6.1.3 Import using copy/paste of text . . . 97
6.1.4 External files . . . 97
**6.2 Import tracks . . . 98**
**6.3 Import high-throughput sequencing data . . . 100**
6.3.1 454 from Roche Applied Science . . . 101
6.3.2 Illumina . . . 103
6.3.3 SOLiD from Life Technologies . . . 108
6.3.4 Fasta read files . . . 111
6.3.5 Sanger sequencing data . . . 112
6.3.6 Ion Torrent PGM from Life Technologies . . . 114
6.3.7 Complete Genomics . . . 115
6.3.8 General notes on handling paired data . . . 116
6.3.9 SAM and BAM mapping files . . . 116
**6.4 Data export . . . 120**
6.4.1 Export of folders and multiple elements in CLC format . . . 124
6.4.2 Export of dependent elements . . . 124
6.4.3 Export history . . . 125
6.4.4 The CLC format . . . 126
6.4.5 Backing up data from the CLC Workbench . . . 126
6.4.6 Export of workflow output . . . 127
6.4.7 Export of tables . . . 127
**6.5 Export graphics to files . . . 128**
6.5.1 Which part of the view to export . . . 129
6.5.2 Save location and file formats . . . 130
6.5.3 Graphics export parameters . . . 131
6.5.4 Exporting protein reports . . . 133
**6.6 Export graph data points to a file . . . 133**

6.7 Copy/paste view output . . . . . 134

*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 J.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.

### 6.1.1 Import using the import dialog

To start the import using the import dialog:

**click Import () in the Toolbar | 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.

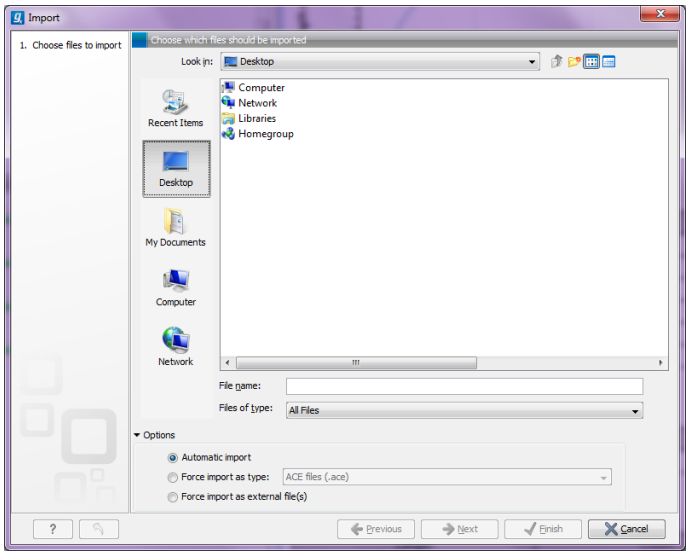



Figure 6.1: *The import dialog.*

Next, select one or more files or folders to import and click **Next**.

This allows you 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.

### 6.1.2 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.

### 6.1.3 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.

### 6.1.4 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 1.7).

## 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.

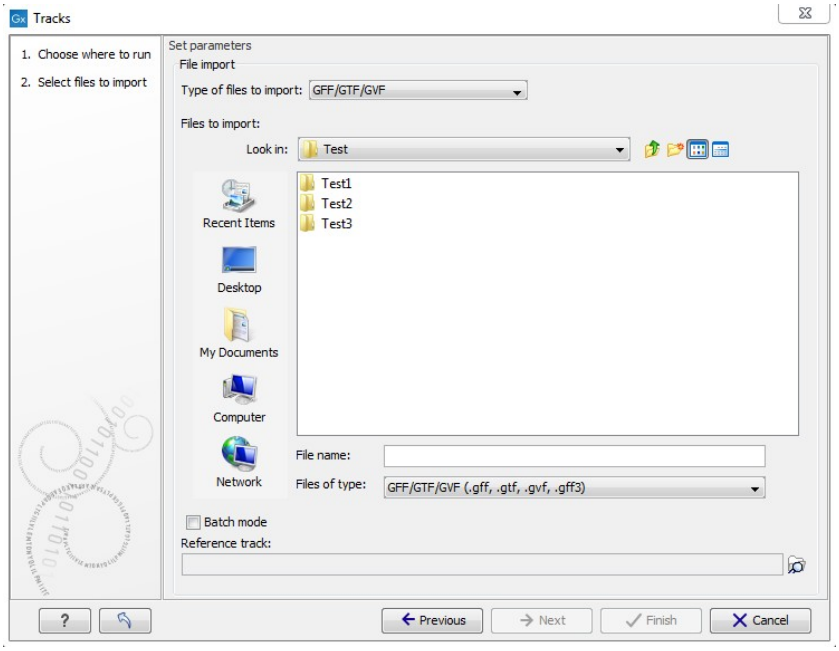



Figure 6.2: *Select files to import.*

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 24.4).

**GFF/GTF/GVF** Annotations in gff/gtf/gvf formats. This is explained in detail in the user manual for the GFF annotation plugin:

http://www.clcbio.com/clc-plugin/annotate-sequence-with-gff-file/. This can be particularly useful when working with transcript annotations downloaded from from Ensembl available in gvf format: http://www.ensembl.org/info/data/ftp/index.html.

**VCF** This is the file format used for variants by the 1000 Genomes Project and it has become a standard format. Read how to access data at http://www.1000genomes.org/data#DataAccess. When importing a single VCF file, you will get a track for each sample contained in the VCF file. In cases where more than one sample is contained in a VCF file, you can choose to import the files together or individually by using the batch mode found in the lower left side of the wizard 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 import more than one VCF file that each contain more than one sample, 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. E.g. 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.

**Complete Genomics master var file** This is the file format used by Complete Genomics for all kinds of variant data and can be used to analyze and visualize the variant calls made by Complete Genomics. Please note that you can import evidence files with the read alignments into the *CLC Genomics Workbench* as well (refer to the Complete Genomics import section of the Workbench user manual).

**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.

**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, and then go to 'Complete COSMIC data' and click on the file link. Users must first register to download the database. Import the file as a track, and then go to Data Management, scroll down to COSMIC and click on **Workflow Configuration**. Click on 'Select own' and select the track that has just been imported. Through Import->Tracks we support the following COSMIC databases in tsv format that can be manually downloaded from the COSMIC ftp site:

- Complete COSMIC data (file: CosmicCompleteExport.tsv)
- Complete mutation data (file: CosmicMutantExport.tsv)
- All Mutation in census genes (file: CosmicMutantExportCensus.tsv)

Please see chapter J.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 in the http://www.clcsupport.com/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.

## 6.3 Import high-throughput sequencing data

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

- The 454 FLX System from Roche
- Illumina's Genome Analyzer, HiSeq and MiSeq

- SOLiD system from Applied Biosystems (read mapping is performed in color space, see section 25.4)
- Ion Torrent from Life Technologies
- Complete Genomics (only processed data - master var and evidence files)

The reason for having dedicated tools for this is to standardize the data so that most downstream analyses and visualization of the data works seamlessly with all sequencing platforms. In addition to these formats, mapped data in SAM/BAM format can also be imported.

This section will describe the various importers in detail.

Clicking on the **Import** ( ) button in the top toolbar will bring up a list of the supported data types as shown in figure 6.3.

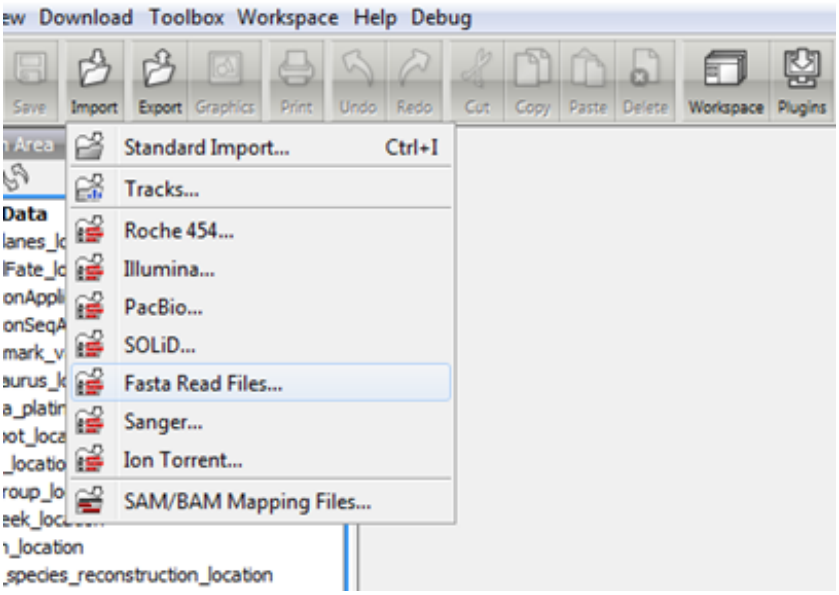



Figure 6.3: *Choosing what kind of data you wish to import.*

Select the appropriate format and then fill in the information as explained in the following sections.

Please note that alignments of *Complete Genomics* data can be imported using the SAM/BAM importer, see section 6.3.7 below.

### 6.3.1 454 from Roche Applied Science

Choosing the Roche 454 import will open the dialog shown in figure 6.4.

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:

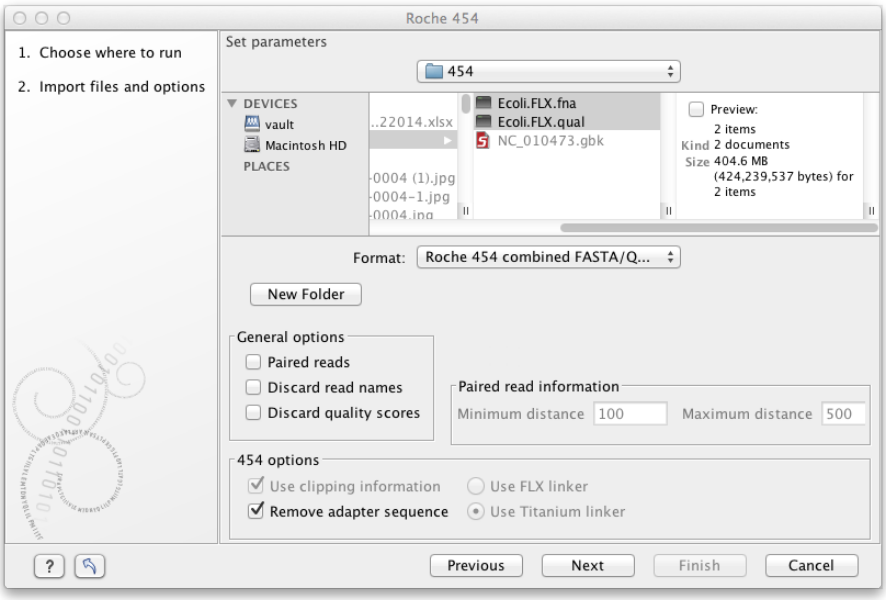



Figure 6.4: *Importing data from Roche 454.*

- 454 FASTA files (`.fna`) which contain the sequence data.
- Quality files (`.qual`) which contain the quality scores.

For all formats, 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 6.5. 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 6.3.8.

- **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.

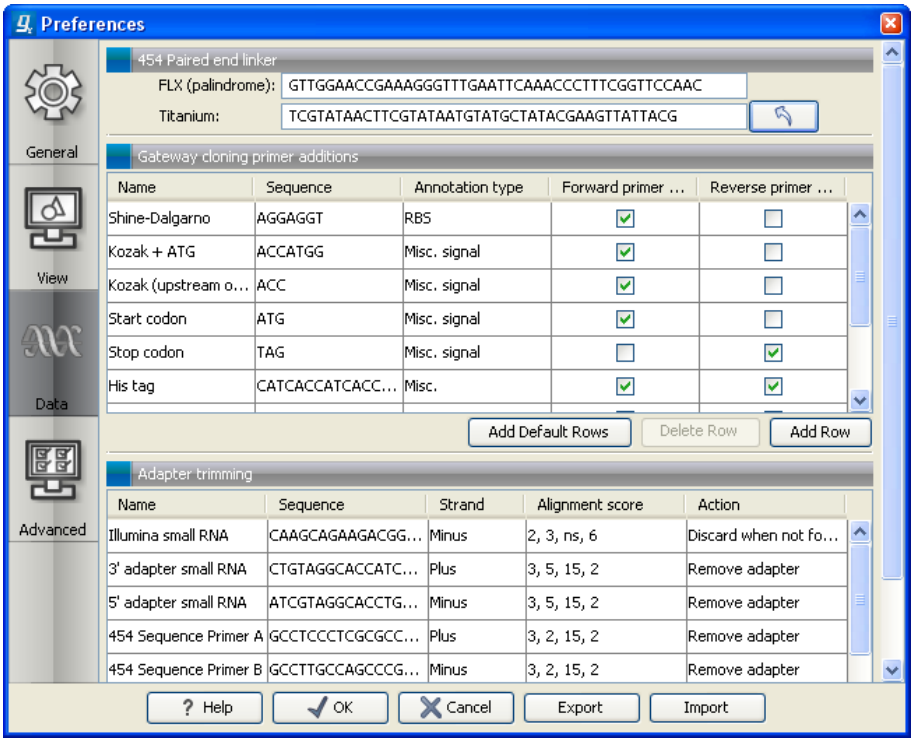



Figure 6.5: *Specifying linkers for 454 import.*

**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 8.2). 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 8.1).

### 6.3.2 Illumina

The *CLC Genomics Workbench* supports data from Illumina's Genome Analyzer, HiSeq 2000 and the MiSeq systems. Choosing the Illumina import will open the dialog shown in figure 6.6.

The file formats accepted are:

- Fastq
- Scarf
- Qseq

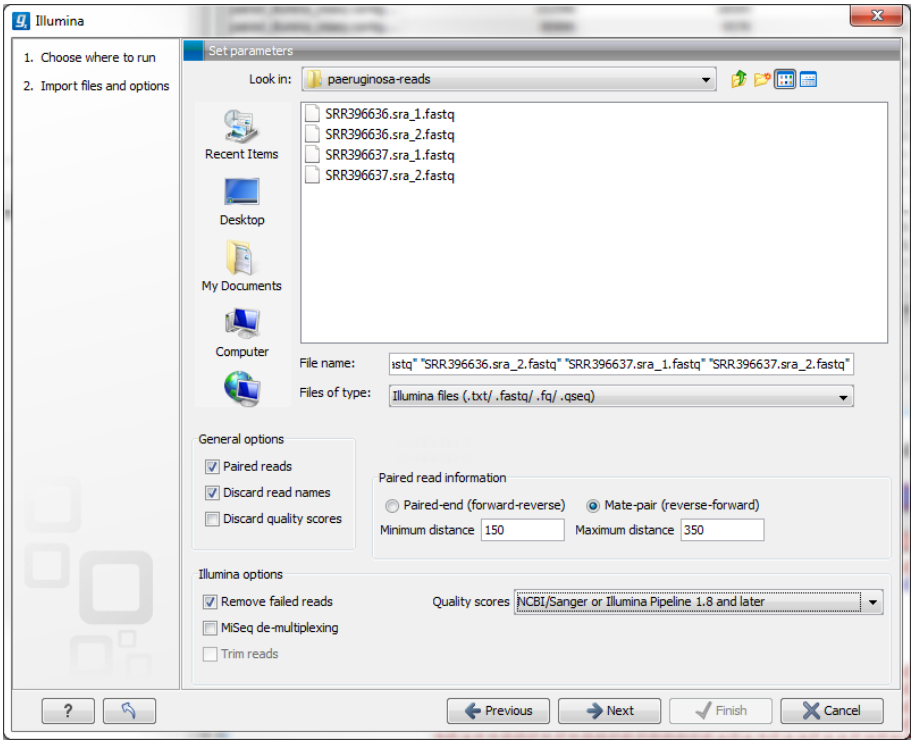



Figure 6.6: *Importing data from Illumina systems.*

Paired data in any of these formats can be imported.

Note that there is information inside qseq and fastq files specifying whether a read has passed a quality filter or not. If you check **Remove failed reads** these reads will be ignored during import. For qseq files there is a flag at the end of each read with values 0 (failed) or 1 (passed). In this example, the read is marked as failed and if Remove failed reads is checked, the read is removed.

```
M10  68  1  1  28680  29475  0  1  CATGGCCGTACAGGAAACACACATCATAGCATCACACGA  BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB  0
```

For fastq files, part of the header information for the quality score has a flag where Y means failed and N means passed. In this example, the read has not passed the quality filter:

```
@EAS139:136:FC706VJ:2:2104:15343:197393 1:Y:18:ATCACG
```

**Note!** In the **Illumina pipeline 1.5-1.7**, the letter B in the quality score has a special meaning. 'B' is used as a trim clipping. This means that when selecting Illumina pipeline 1.5-1.7, the *reads are automatically trimmed* when a B is encountered in the input file. This will happen also if you choose to discard quality scores during import.

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.

**For all formats, compressed data in gzip format is also supported (.gz).**

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

- **Paired reads**. For paired import, you can select whether the data is **Paired-end** or **Mate-pair**. For paired data, the Workbench expects the first reads of the pairs to be in one file and

the second reads of the pairs to be in another. When importing one pair of files, the first file in a pair will is assumed to contain the first reads of the pair, and the second file is assumed to contain the second read in a pair. So, for example, if you had specified that the pairs were in forward-reverse orientation, then the first file would be assumed to contain the forward reads. The second file would be assumed to contain the reverse reads.

When loading files containing paired data, the *CLC Genomics Workbench* sorts the files selected according to rules based on the file naming scheme:

- For files coming off the CASAVA1.8 pipeline, we organize pairs according to their identifier and chunk number. Files named with `_R1_` are assumed to contain the first sequences of the pairs, and those with `_R2_` in the name are assumed to contain the second sequence of the pairs.
- For other files, we sort them all alphanumerically, and then group them two by two. This means that files 1 and 2 in the list are loaded as pairs, files 3 and 4 in the list are seen as pairs, and so on.

In the simplest case, the files are typically named as shown in figure 6.6. In this case, the data is paired end, and the file containing the forward reads is called `s_1_1_sequence.txt` and the file containing reverse reads is called `s_1_2_sequence.txt`. Other common filenames for paired data, like `_1_sequence.txt`, `_1_qseq.txt`, `_2_sequence.txt` or `_2_qseq.txt` will be sorted alphanumerically. In such cases, files containing the final `_1` should contain the first reads of a pair, and those containing the final `_2` should contain the second reads of a pair.

For files from CASAVA1.8, files with base names like these: ID_R1_001, ID_R1_002, ID_R2_001, ID_R2_002 would be sorted in this order:

1. ID_R1_001
2. ID_R2_001
3. ID_R1_002
4. ID_R2_002

The data in files ID_R1_001 and ID_R2_001 would be loaded as a pair, and ID_R1_002, ID_R2_002 would be loaded as a pair.

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.

If you do not choose to discard your read names on import (see next parameter setting), you can quickly check that your paired data has imported in the pairs you expect by looking at the first few sequence names in your imported paired data object. The first two sequences should have the same name, except for a `1` or a `2` somewhere in the read name line.

Paired-end and mate-pair data are handled the same way with regards to sorting on filenames. Their data structure is the same the same once imported into the Workbench. The only difference is that the expected orientation of the reads: reverse-forward in the case of mate pairs, and forward-reverse in the case of paired end data. Read more about handling paired data in section 6.3.8.

- **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 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. Read more about the quality scores of Illumina below.
- **MiSeq de-multiplexing**. For MiSeq multiplexed data, one file includes all the reads containing barcodes/indices from the different samples (in case of paired data it will be two files). Using this option, the data can be divided into groups based on the barcode/index. This is typically the desired behavior, because subsequent analysis can then be executed in batch on all the samples and results can be compared at the end. This is not possible if all samples are in the same file after import. The reads are connected to a group using the last number in the read identifier.
- **Trim reads**. This option applies to Illumina Pipeline 1.5 to 1.7. In this pipeline, the value 2 (B) has special meaning and is used as a trim clipping. This means that when selecting Illumina Pipeline 1.5 and later, the reads are trimmed when a B is encountered in the input file if the **Trim reads** option is checked.

Click **Next** to adjust how to handle the results (see section 8.2). 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 8.1).

### Quality scores in the Illumina platform

The quality scores in the FASTQ format come in different versions. You can read more about the FASTQ format at `http://en.wikipedia.org/wiki/FASTQ_format`. When you select to import Illumina data and click **Next** there is an option to use different quality score schemes at the bottom of the dialog (see figure 6.7).

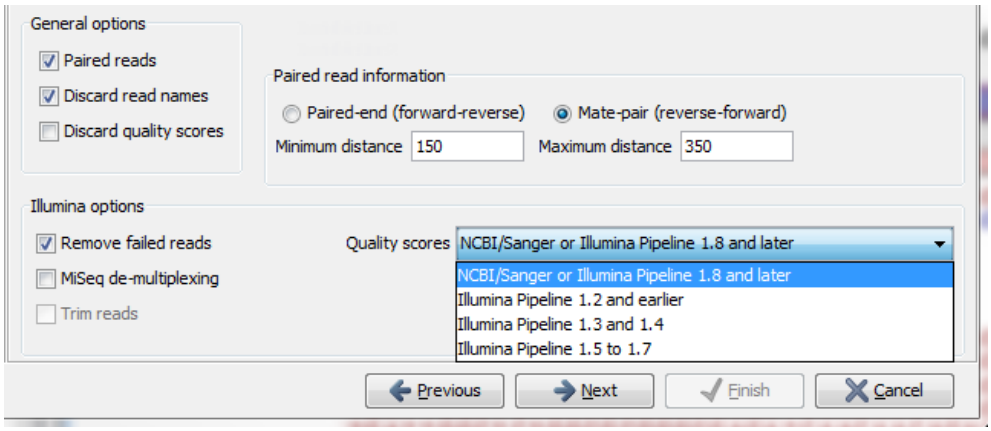



Figure 6.7: *Selecting the quality score scheme.*

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. This means that when selecting Illumina Pipeline 1.5 and later, the reads are trimmed when a B is encountered in the input file if the **Trim reads** option is checked.

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
IIIIIIIIIIIIIIIIIIIIIIIIFIIII'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 SOLiD from Life Technologies

Choosing the SOLiD import will open the dialog shown in figure 6.8.

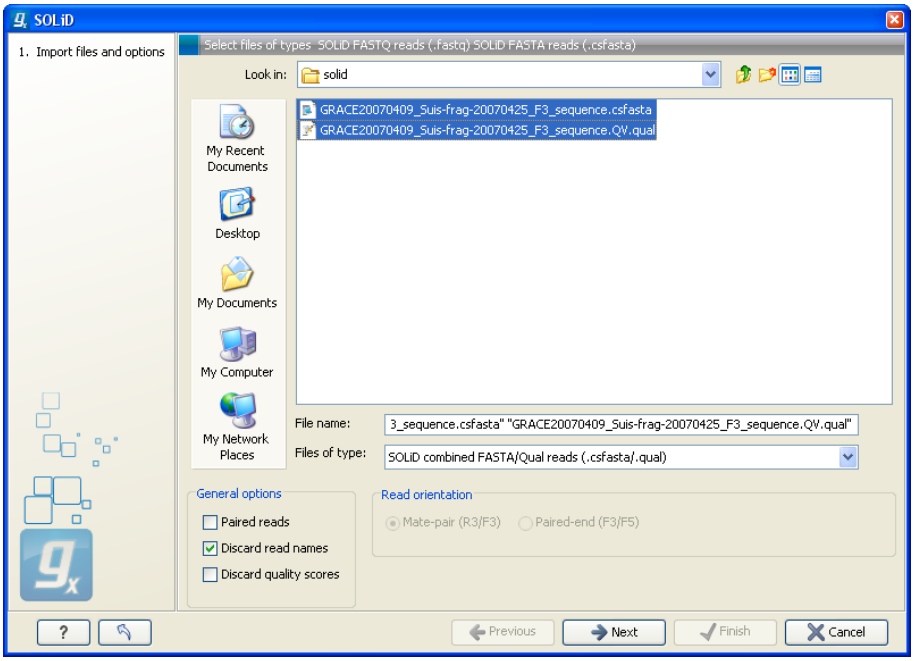



Figure 6.8: *Importing data from SOLiD from Applied Biosystems.*

There are two formats accepted: the XSQ format which is the native format of newer SOLiD systems, and the csfasta format which is the color space version of fasta format.

#### The XSQ format

An XSQ file can contain results from multiple libraries produced from the same sequencing run. These are identified by a barcode on each read, and when the XSQ file is produced, each read is placed into its appropriate library based on its barcode. The XSQ importer creates separate sequence lists for each library.

Sometimes when an XSQ file is produced, a barcode cannot be identified accurately enough to place the read into a specific library, or the read is for some other reason not assigned to a library. In this case, the read is placed into an "Unclassified" or "Unassigned" library.

In the case of paired reads, it sometimes happens that one read of a pair could not be read. When the XSQ file is imported in the *CLC Genomics Workbench*, the other read of such a pair is placed into a sequence list with " (single)" appended to the name, whereas all intact pairs are placed (alternating) into a sequence list with " (paired)" appended to the name. Thus, two sequence lists are produced for the library.

Hence, when importing data in XSQ format the number of imported files can vary. In the example shown here, where the XSQ file contain a library with the name "Main" (containing paired reads) and an "Unclassified" library (containing reads where e.g. the barcode could not be read), the imported data are segregated into the following sequence lists:

1. Main (single)
2. Main (paired)
3. Unclassified

An XSQ file sometimes contains reads in both base space and color space, and when that is the case, each read library in the XSQ file that contains reads in both formats will result in two files, with " (base space)" and " (color space)" appended to their names, respectively.

**The csfasta format**

If you want to import quality scores with csfasta files, qual files should also be provided. The reads in a csfasta file look like this:

```
>2_14_26_F3
T0112131222002211230321112210212101313322221O1
>2_14_192_F3
T1100212211003100301200220322211132102211222 3
>2_14_233_F3
T0110013323111212123120223102033122011321112 3
>2_14_294_F3
T2130121323000000213232 12232.03300033102330332
```

All reads start with a T which specifies the right phasing of the color sequence.

If a reads has a `.` as you can see in the last read in the example above, it means that the color calling was ambiguous (this would have been an `N` if we were in base space). In this case, the Workbench simply cuts off the rest of the read, since there is no way to know the right phase of the rest of the colors in the read. If the read starts with a dot, it is not imported. If all reads start with a dot, a warning dialog will be displayed. The handling of dots is identical for XSQ and csfasta files.

In the quality file, the equivalent value is `-1`, and this will also cause the read to be clipped.

When the example above is imported into the Workbench, it looks as shown in figure 6.9.

For more information about color space, please see section 25.4.

In addition to the csfasta and XSQ formats used by SOLiD, you can also input data in fastq format. This is particularly useful for data downloaded from the Sequence Read Archive at

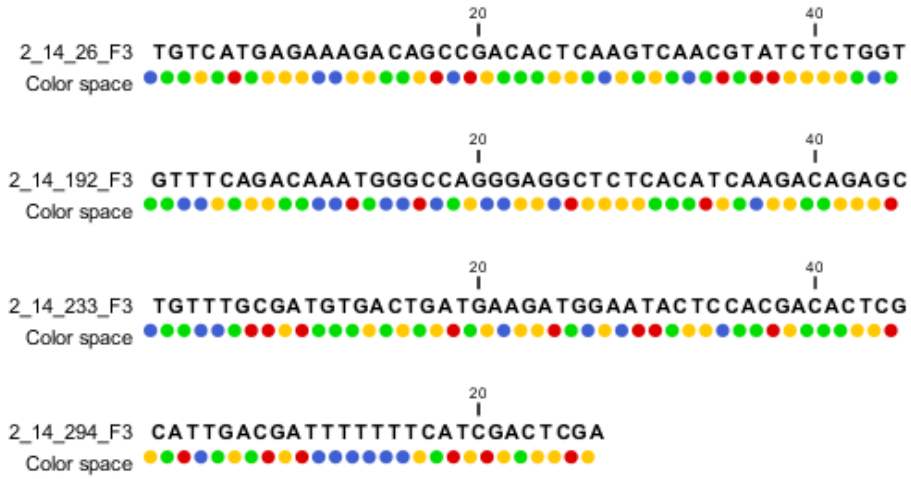



Figure 6.9: *Importing data from SOLiD from Applied Biosystems. Note that the fourth read is cut off so that the color following the dot are not included*

NCBI (http://www.ncbi.nlm.nih.gov/Traces/sra/). An example of a SOLiD fastq file is shown here with both quality scores and the color space encoding:

```
@SRR016056.1.1 AMELIA_20071210_2_YorubanCGB_Frag_16bit_2_51_130.1 length=50
T3100031312131021102231222331121211302212120133 2213
+SRR016056.1.1 AMELIA_20071210_2_YorubanCGB_Frag_16bit_2_51_130.1 length=50
!*%;2'%%050%'0'3%%5*.%%%),%%%%&%%%%%%'%%%%%'%%3+%%%
@SRR016056.2.1 AMELIA_20071210_2_YorubanCGB_Frag_16bit_2_51_223.1 length=50
T2000220112002121101201033221112213321233122130 2222
+SRR016056.2.1 AMELIA_20071210_2_YorubanCGB_Frag_16bit_2_51_223.1 length=50
!%%)%'))'&'%(((&%/&)%+(%%%&%%%%%%%%%%%%%%%%+%%%%%%+'
```

For all formats, compressed data in gzip format is also supported (.gz).

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

- **Paired reads**. When you import paired data, two different protocols are supported:
  - **Mate-pair**. For mate-pair data, the reads should be in two files with `_F3` and `_R3` in front of the file extension. The orientation of the reads is expected to be forward-forward.
  - **Paired-end**. For paired-end data, the reads should be in two files with `_F3` and `_F5-P2` or `_F5-BC`. The orientation is expected to be forward-reverse.

  Read more about handling paired data in section 6.3.8. Please note that for XSQ files, the pairing protocol is defined in the file itself, which means that the choices of protocol will be ignored.

  An example of a complete list of the four files needed for a SOLiD mate-paired data set including quality scores:

  ```
  dataset_F3.csfasta   dataset_F3.qual
  dataset_R3.csfasta   dataset_R3.qual
  ```

or

```
dataset_F3.csfasta   dataset_F3_.QV.qual
dataset_R3.csfasta   dataset_R3_.QV.qual
```

- **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 choose to discard quality scores, you do not need to select a .qual file when importing csfasta files.

Click **Next** to adjust how to handle the results (see section 8.2). 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 8.1).

### 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. For import of other fasta format data, such as reference sequences, please use the (  )**Standard Import**, described in section 6 as this import format also includes the descriptions.

The dialog for importing data in fasta format is shown in figure 6.10.

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 6.3.8.
- **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.

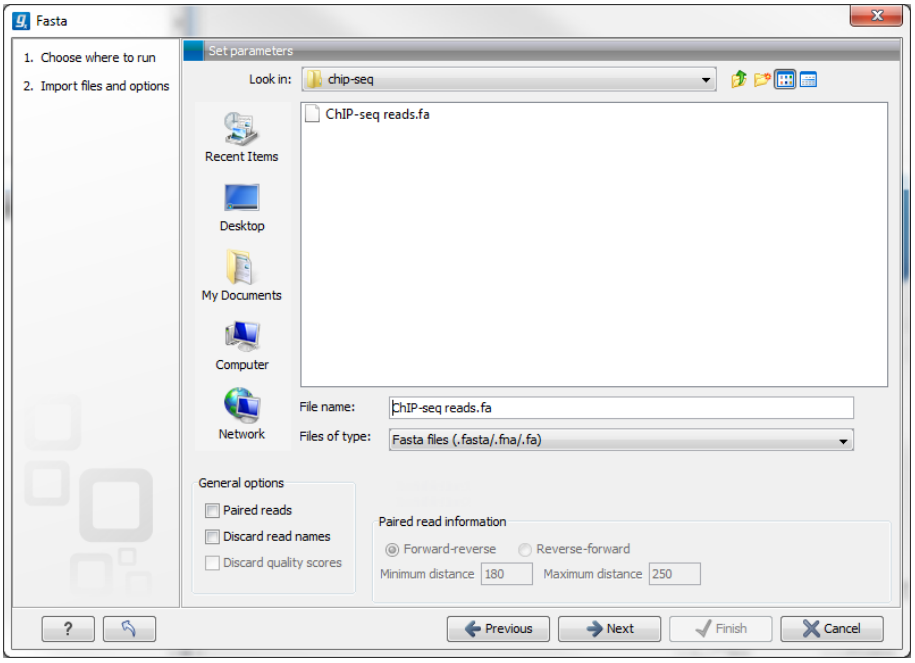



Figure 6.10: *Importing data in fasta format.*

- **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 8.2). 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 8.1).

### 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).

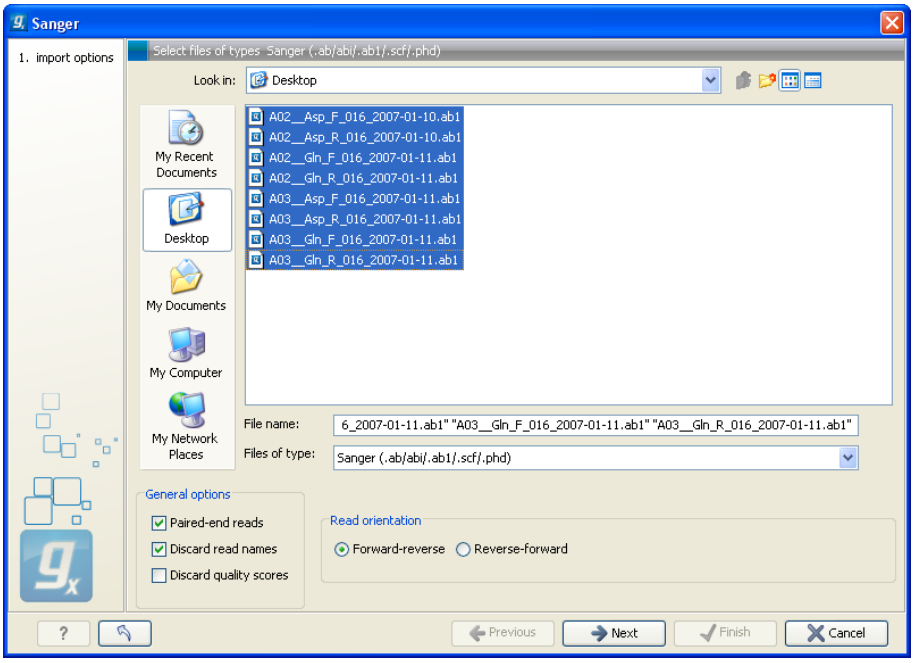



Figure 6.11: *Importing data from Sanger sequencing.*

The dialog for importing data Sanger sequencing data is shown in figure 6.11.

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 6.3.8.
- **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 8.2). 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 8.1).

### 6.3.6 Ion Torrent PGM from Life Technologies

Choosing the Ion Torrent import will open the dialog shown in figure 6.12.

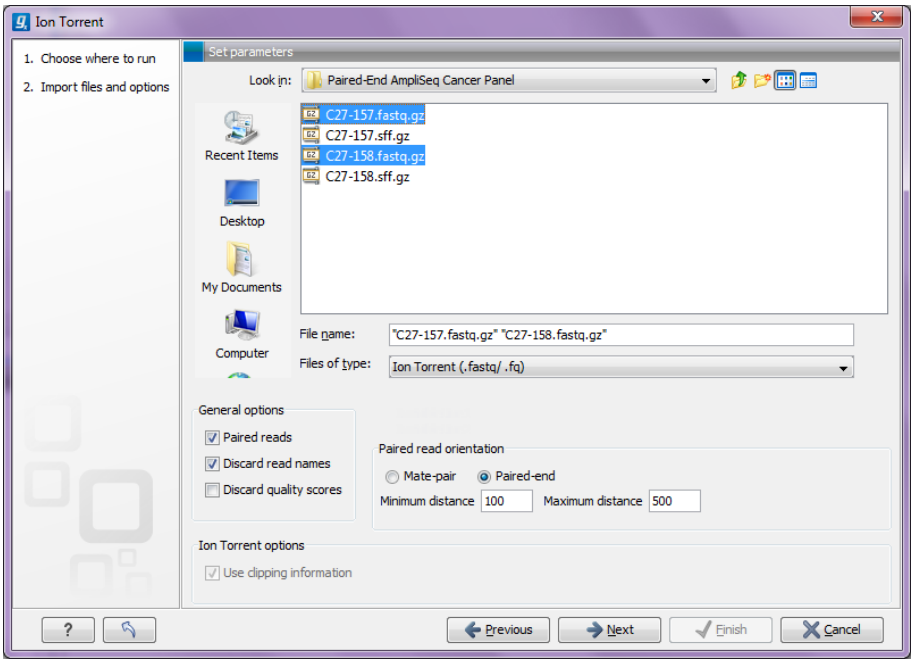



Figure 6.12: *Importing data from Ion Torrent.*

We support import of two kinds of data from the Ion Torrent system:

- SFF files (`.sff`)
- Fastq files (`.fastq`). Quality scores are expected to be in the NCBI/Sanger format (see section 6.3.2)

For all formats, compressed data in gzip format is also supported (.gz).

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

- **Paired reads**. The *CLC Genomics Workbench* supports both paired end and mate pair protocols.

  **Paired end** Paired end data from Ion Torrent comes in two files per data set. The first file in is assumed to contain the first reads of the pair, and the second file is assumed to contain the second read in a pair. On import, the orientation of the reads is set to forward - reverse. When the reads have been imported, there will be one file with intact pairs, and one file where one part of the pair is missing (in this case, "single" is appended to the file name). The Workbench connects the right sequences together in the pair based on the read name. Read more about handling paired data in section 6.3.8.

  **Mate pair** The mate pair protocol for Ion Torrent entails that the two reads are separated by a linker sequence. During import of paired data, the linker sequence is removed and the two reads are separated and put into the same sequence list. You can change the linker sequence in the **Preferences** (in the **Edit** menu) under **Data**. When looking for

the linker sequence, the Workbench requires 80 % of the maximum alignment score, using the following scoring scheme: matches = 1, mismatches = -2 and indels = -3. 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 Ion Torrent mate pair 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 6.3.8.

- **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.

For sff files, you can also decide whether to use the clipping information in the file or not.

### 6.3.7 Complete Genomics

With *CLC Genomics Workbench 8.0* you can import *evidence* and *variation* files from Complete Genomics.

The variation files can be imported as tracks (see section 6.2).

The evidence files can be imported using the SAM/BAM importer, see section 6.3.9.

In order to import the evidence data file it need to be converted first. This is achieved using the CGA tools that can be downloaded from `http://www.completegenomics.com/sequence-data/cgatools/`.

The procedure for converting the data is the following.

1. Download the human genome in fasta format and make sure the chromosomes are named `chr<number>.fa`, e.g. `chr9.fa`.
2. Run the **fasta2crr** tool with a command like this:
   `cgatools fasta2crr --input chr9.fa --output chr9.crr`
3. Run the **evidence2sam** tool with a command like this:
   `cgatools evidence2sam --beta -e evidenceDnbs-chr9-.tsv -o chr9.sam -s chr9.crr`
   where the .tsv file is the evidence file provided by Complete Genomics (you can find sample data sets on their ftp server: `ftp://ftp2.completegenomics.com/`).
4. **Import** (  ) the fasta file from 1. into the Workbench.
5. Use the SAM/BAM importer (section 6.3.9) to import the file created by the evidence2sam tool.

Please refer to the CGA documentation for a description about these tools. Note that this is not software supported by CLC bio.

### 6.3.8 General notes on handling paired data

During import, information about paired data (distances and orientation) can be specified (see figure 6.3 and figure 6.6 for data import examples of Roche 454 and Illumina reads, respectively) and is 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 same 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 reverse 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 10.4 as shown in figure 6.13.

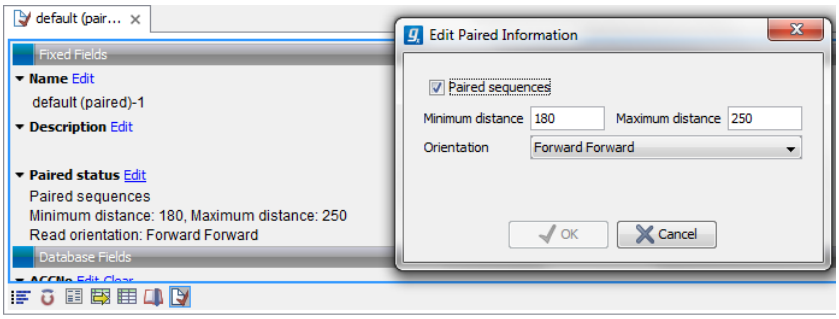



Figure 6.13: *The paired orientation and distance.*

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.

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.

### 6.3.9 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 *CLC Genomics 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 K.

**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. The sequence lists are given names of this form for single reads "<read group id> [read group sample] (single) un-mapped reads" and this form for paired reads "<read group id> [read group sample] (paired) un-mapped reads".

  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 6.14).

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, an entry will appear in the "Status" column indicating this. E.g "Length differs" or "Input missing"[1].

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.

**Some 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 6.15). 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).

[1] If you are using a CLC Genomics Server to import files located on the Server (rather than locally), then checks for corresponding reference names and lengths cannot be carried out, so nothing will be reported in this section of the Wizard. This means you will be able to continue to launch the import with correct or incorrect reference sets specified. However, any inconsistencies in these will lead to the import task failing with an error related to this.

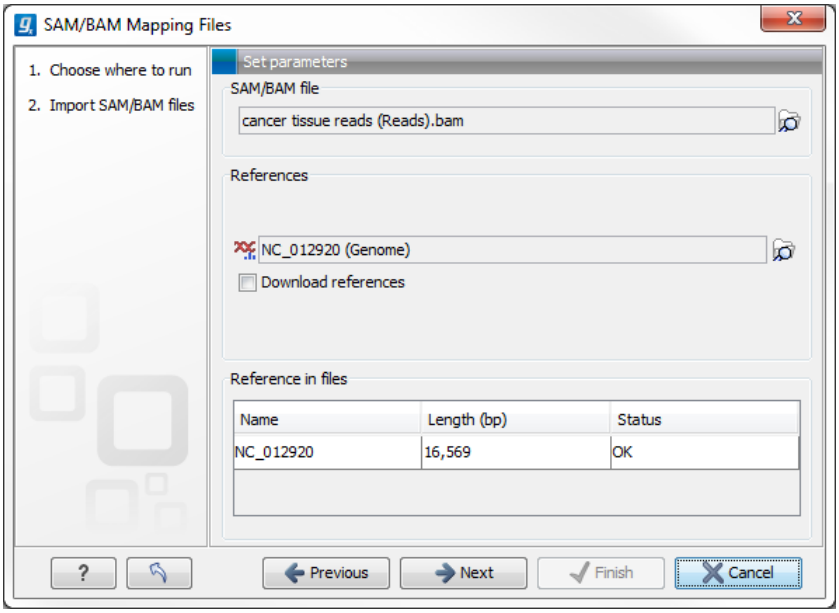



Figure 6.14: *Defining SAM/BAM file and reference sequence(s).*

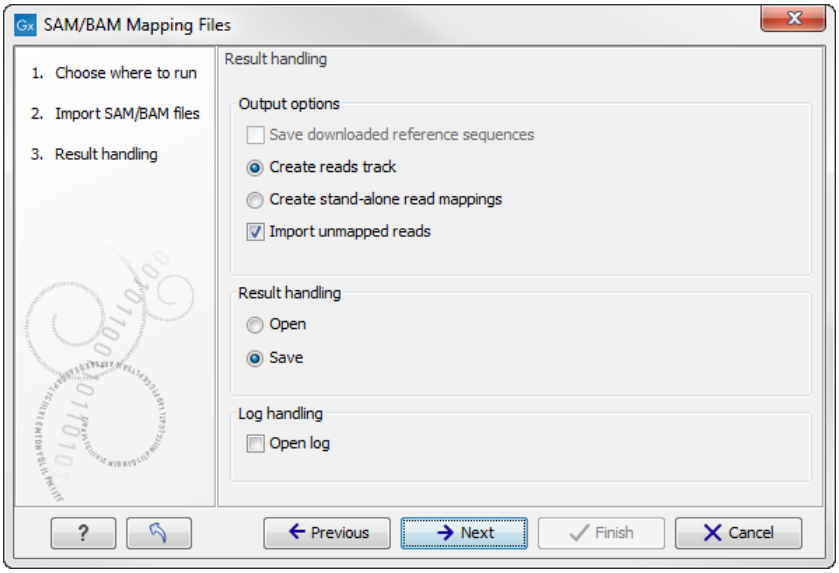



Figure 6.15: *Specify the result handling.*

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 8.2).

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 Data export

The exporter can be used to:

- Export bioinformatic data in most of the formats that can be imported. There are a few exceptions (see section J.1).
- Export one or more data elements at a time to a given format. When multiple data elements are selected, each is written out to an individual file, unless compression is turned on, or "Output as single file" is selected.

The standard export functionality can be launched using the Export button on the toolbar, or by going to the menu:

**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 6.4.2.

The general steps when configuring a standard export job are:

- (Optional) Select the data to export 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 format the data should be exported to.
- Select the data to export, or confirm the data to export if it was already selected via the **Navigation Area**.
- Configure the parameters. This includes compression, multiple or single outputs, and naming of the output files, along with other format-specific settings where relevant.
- Select where the data should be exported to.
- Click on the button labeled **Finish**.

**Selecting data for export - part I.** You can select the data elements to export **before** you run the export tool **or after** the format to export to has been selected. If you are not certain which formats are supported for the data being exported, we recommend selecting the data in the **Navigation Area** before launching the export tool.

**Selecting a format to export to.** When data is pre-selected in the **Navigation Area** before launching the export tool you will see a column in the export interface called **Supported formats**. Formats that the selected data elements can be exported to are indicated by a "Yes" in this column. Supported formats will appear at the top of the list of formats. See figure 6.16.

Formats that cannot be used for export of the selected data have a "No" listed in the **Supported formats** column. If you have selected multiple data elements of different types, then formats

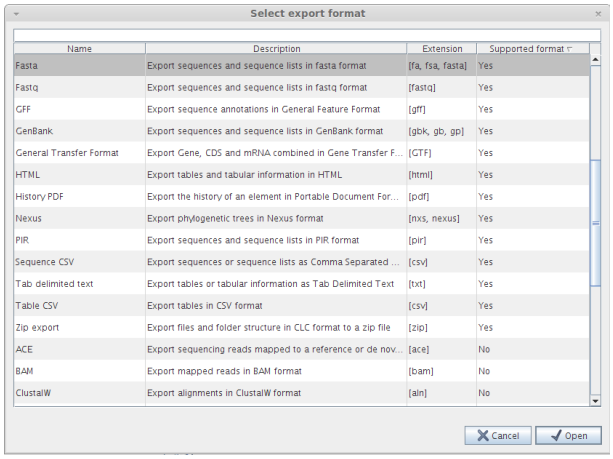



Figure 6.16: *The Select exporter dialog where sequence lists were pre-selected in the Navigation Area before launching the export tool. Here, the formats sequence lists can be exported to are listed at the top, with a Yes in the Selected formats column. Other formats are found below, with No in this column.*

which can be used for some of the selected data elements but not all of them are indicated by the text "For some elements" in this column.

Please note that the information in the **Supported formats** column only refers to the data already selected in the **Navigation Area**. If you are going to choose your data later in the export process, then the information in this column will not be pertinent.

Only one export format is available if you select a folder to be exported. This is described in more detail in section 6.4.1.

**Finding a particular format in the list.** You can quickly find a particular format by using the text box at the top of the exporter window as shown in figure 6.17, where formats that include the term VCF are searched for. This search term will remain in place the next time the Export tool is launched. Just delete the text from the search box if you no longer wish only the formats with that term to be listed.

When the desired export format has been identified, click on the button labeled **Open**.

**Selecting data for export - part II.** A dialog appears, with a name reflecting the format you have chosen. For example if the "Variant Call Format" (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.

In figure 6.18 we show the selection of a variant track for export to VCF format.

The parameters under **Basic export parameters** and **File name** are offered when exporting to any format. There may be additional parameters for particular export formats. This is illustrated here with the VCF exporter, where a reference sequence track must be selected. See figure 6.19.

**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

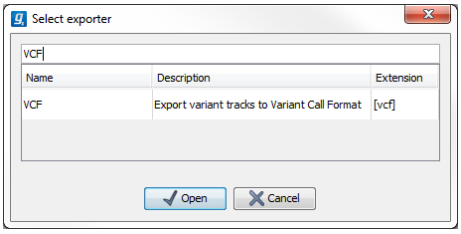



Figure 6.17: *The text field has been used to search for VCF format in the Select exporter dialog.*

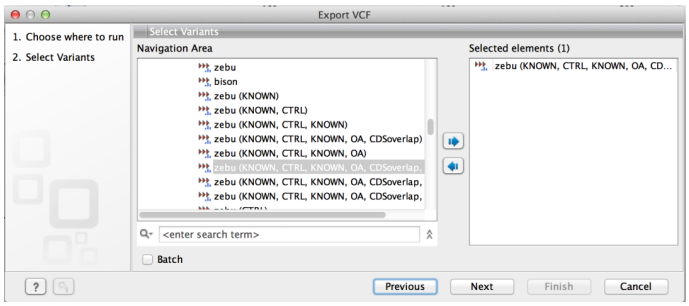



Figure 6.18: *The Select exporter dialog. Select the data element(s) to export.*

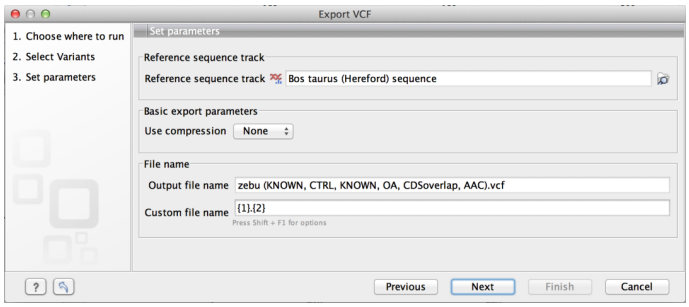



Figure 6.19: *Set the export parameters. When exporting in VCF format, a reference sequence track must be selected.*

exported file for each data element individually.

**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.

**Choosing the exported file name(s)** The default setting for the **File name** is to use the original data element name as the basename and the export format as the suffix.

When exporting just one data element, or exporting to a zip file, the desired filename could just be typed in the Custom file name box.

When working with the export of multiple files, using some combination of the terms shown by default in this field and in figure 6.21 are recommended. Clicking in the **Custome file name** field with the mouse and then simultaneously pressing the Shift + F1 keys bring up a list of the available terms that can be included in this field. You can see that "{1}" is the name of the input element and "{2}" is the file name extension. It is possible to change the input file name and the file extension name. We will look at an example to illustrate this:

In this example we would like to change the export file format to .fasta in a situation where .fa was the default format that would be used if you kept the default file extension suggestion ("{2}"). To do this replace "{2}" with ".fasta" in the "Custom file name field". You can see that when changing "{2}" to ".fasta" , the file name extension in the "Output file name" field automatically changes to the new format (see figure 6.20).

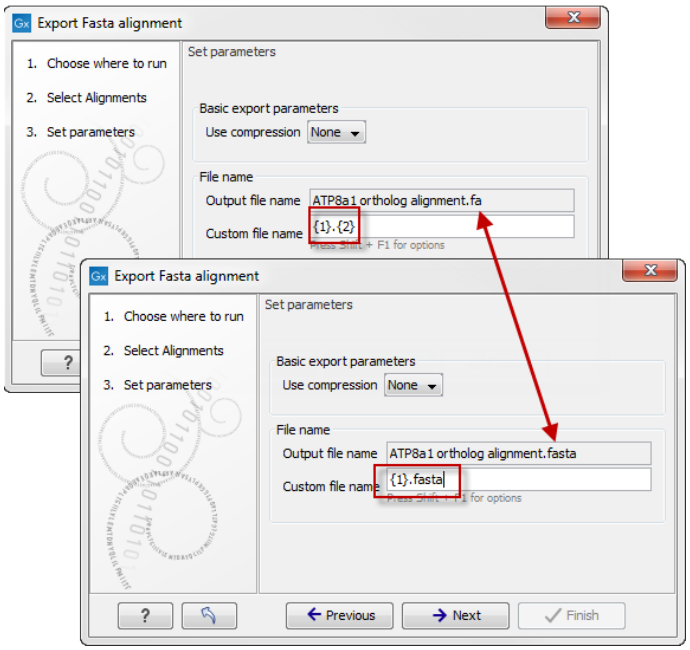



Figure 6.20: *The file name extension can be changed by typing in the preferred file name format.*

As you add or remove text and terms in the **Custome 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. When working with multiple files, only the name of the first one is shown. Just move the mouse cursor over the name shown in the **Output file name** field to show a listing of the all the filenames.

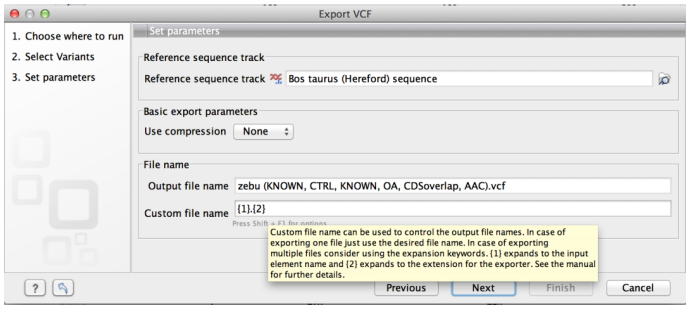



Figure 6.21: *Use the custom file name pattern text field to make custom names.*

The last step is to specify the exported data should be saved (figure 6.22).

**A 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.

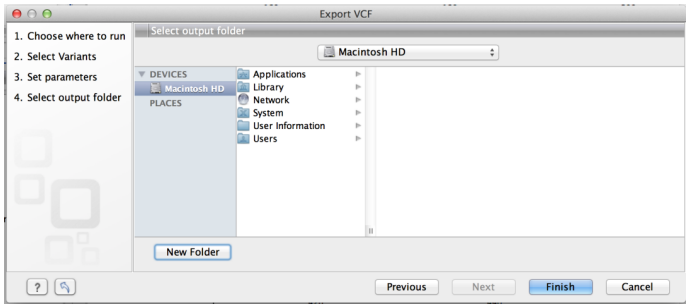



Figure 6.22: *Select where to save the exported data.*

### 6.4.1 Export of folders and multiple elements in CLC format

In the list of export formats presented is one called zip format. Choosing this format means that you wish to export the selected data element(s) or folders to a single, compressed CLC format file. This is useful in cases where you wish to exchange data between workbenches or as part of a simple backup procedure.

A zip file generated this way can be imported directly into a CLC Workbench using the Standard Import tool and leaving the import type as Automatic.

**Note!** When exporting multiple files, the names will be listed in the "Output file name" text field with only the first file name being visible and the rest being substituted by "...", but will appear in a tool tip if you hover the mouse over that field (figure 6.23).

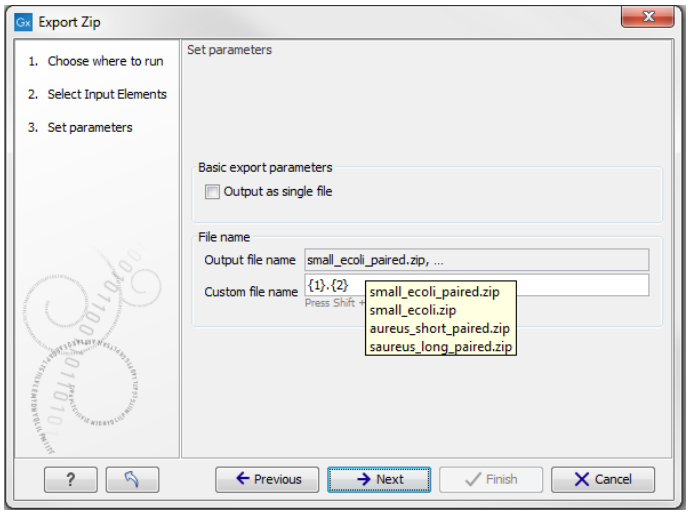



Figure 6.23: *The output file names are listed in the "Output file name" text field.*

### 6.4.2 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**

In this case, the import type can be left as Automatic.

### 6.4.3 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. 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** as the format to export to. See figure 6.24.
- 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. See figure 6.25.
- Select where the data should be exported to.
- Click on the button labeled **Finish**.

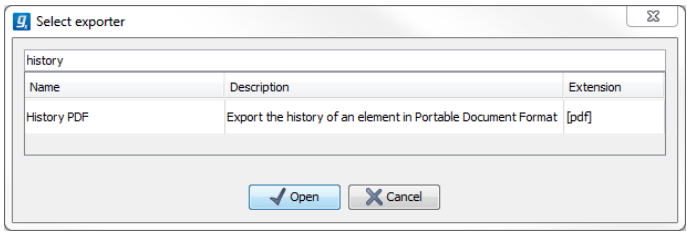



Figure 6.24: *Select "History PDF" for exporting the history of an element.*

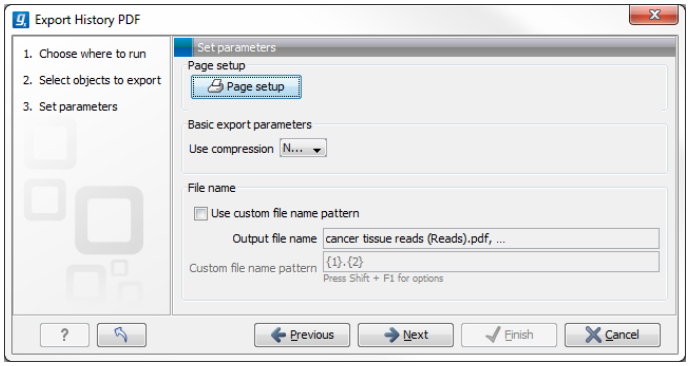



Figure 6.25: *When exporting the history in PDF, it is possible to adjust the page setup.*

### 6.4.4 The CLC format

The *CLC Genomics Workbench* stores bioinformatic data in CLC format. The CLC format contains data, as well as information about that data like history information and comments you may have added.

A given data element in the Workbench can contain different types of data. This is reflected when exporting data, as the choice of different export formats can lead to the extraction of some parts of that data object rather than others. The part of the data exported reflects the type of data a given format can support. As a simple example, if you export the results of an alignment to Annotation CSV format, you will get just the annotation information. If you exported to Fasta alignment format, you would get the aligned sequences in fasta format, but no annotations.

The CLC format holds all the information for a given data object. Thus if you plan to share the data with colleagues who also have a CLC Workbench or you are communicating with the CLC Support team and you wish to share the data from within the Workbench, exporting to CLC format is usually the best choice as all information associated with that data object in your Workbench will then be available to the other person who imports that data.

If you are planning to share your data with someone who does not have access to a CLC Workbench, then you will wish to export to another data format. Specifically, one they can use with the software they are working with.

### 6.4.5 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).

#### 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://www.clcsupport.com/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. You can later re-import the zip file into the Workbench by going to:

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

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** .

### 6.4.6 Export of workflow output

The output from a workflow can be exported by adding one or more workflow export elements (figure 6.26). Multiple elements can be selected by holding down the Ctrl key while clicking on the desired elements.

When the workflow has been created, you can set the export parameters and the location to export data to by double clicking on each export element or leave fields empty and unlocked if you wish users of the Workflow to enter this information when the Workflow is launched.

### 6.4.7 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.

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

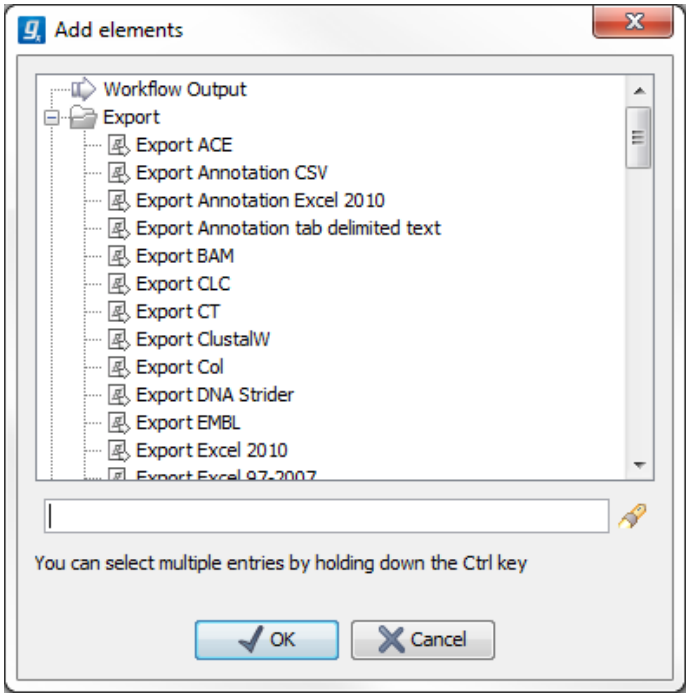



Figure 6.26: *Pressing "Add element" enables addition of workflow export elements.*

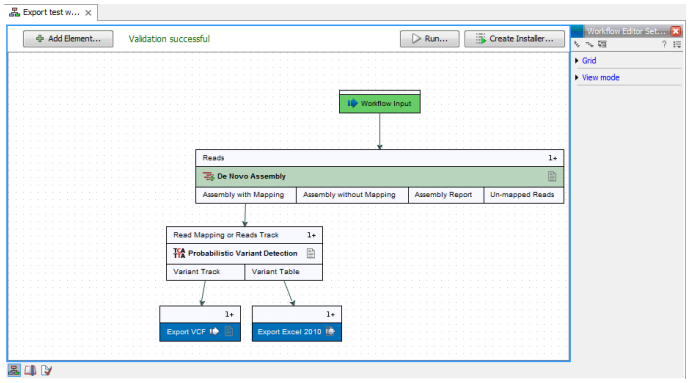



Figure 6.27: *A simple workflow with two export elements. The variant track will be exported in VCF format and the variant table in Excel format.*

from the server or using command line tools, the default number of exported decimals is 3.

## 6.5 Export graphics to files

*CLC Genomics Workbench* supports export of graphics into a number of formats. This way, the visible output of your work can easily be saved and used in presentations, reports etc. The **Export Graphics** function ( ) is found in the **Toolbar**.

*CLC Genomics Workbench* uses a WYSIWYG principle for graphics export: 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 in the program. When you export it, the graphics file will look exactly the same way.

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

**select tab of View | Graphics ( ) on Toolbar**

This will display the dialog shown in figure 6.28.

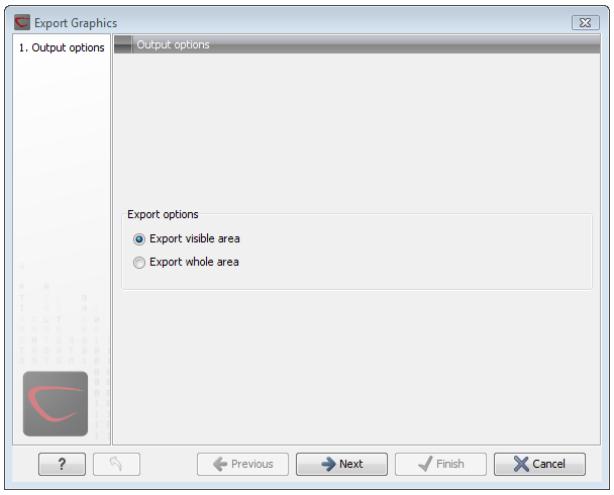



Figure 6.28: *Selecting to export whole view or to export only the visible area.*

### 6.5.1 Which part of the view to export

In this dialog you can choose to:

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

These options are available for all views that can be zoomed in and out. In figure 6.29 is a view of a circular sequence which is zoomed in so that you can only see a part of it.

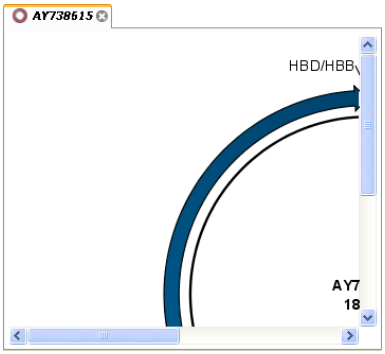



Figure 6.29: *A circular sequence as it looks on the screen.*

When selecting **Export visible area**, the exported file will only contain the part of the sequence that is *visible* in the view. The result from exporting the view from figure 6.29 and choosing **Export visible area** can be seen in figure 6.30.

On the other hand, if you select **Export whole view**, you will get a result that looks like figure 6.31. This means that the graphics file will also include the part of the sequence which is not visible when you have zoomed in.

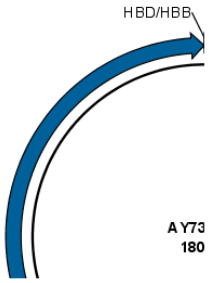



Figure 6.30: *The exported graphics file when selecting Export visible area.*

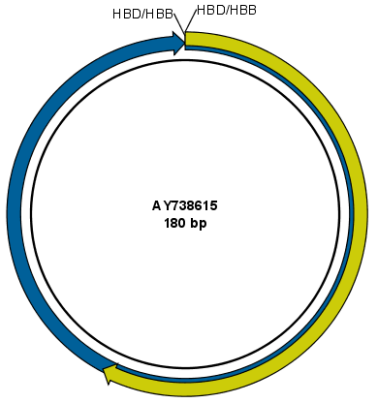



Figure 6.31: *The exported graphics file when selecting Export whole view. The whole sequence is shown, even though the view is zoomed in on a part of the sequence.*

For 3D structures, this first step is omitted and you will always export what is shown in the view (equivalent to selecting **Export visible area**).

Click **Next** when you have chosen which part of the view to export.

### 6.5.2 Save location and file formats

In this step, you can choose name and save location for the graphics file (see figure 6.32).

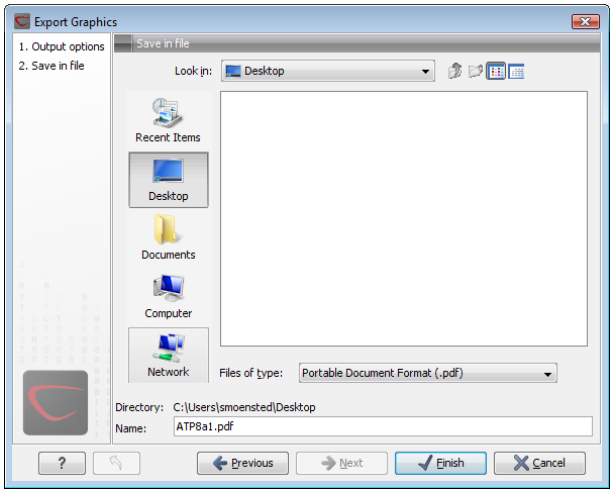



Figure 6.32: *Location and name for the graphics file.*

*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.

#### Vector graphics

Vector graphic is a collection of shapes. Thus what is stored is e.g. 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 e.g. graphs and reports, but less usable for e.g. 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 like e.g. 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 6.1.4 for more about importing external files into *CLC Genomics Workbench*.

### 6.5.3 Graphics export parameters

When you have specified the name and location to save the graphics file, you can either click **Next** or **Finish**. Clicking **Next** allows you to set further parameters for the graphics export, whereas clicking **Finish** will export using the parameters that you have set last time you made a graphics export in that file format (if it is the first time, it will use default parameters).

#### Parameters for bitmap formats

For bitmap files, clicking **Next** will display the dialog shown in figure 6.33.

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

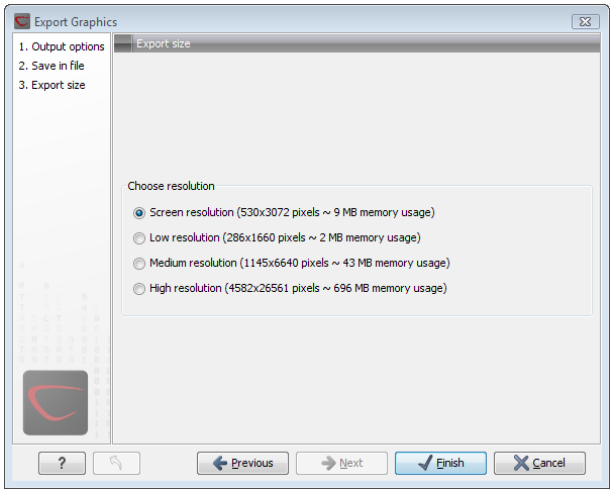



Figure 6.33: *Parameters for bitmap formats: size of the graphics file.*

- Screen resolution
- Low resolution
- Medium resolution
- High resolution

The actual size in pixels is displayed in parentheses. An estimate of the memory usage for exporting the file is also shown. If the image is to be used on computer screens only, a low resolution is sufficient. If the image is going to be used on printed material, a higher resolution is necessary to produce a good result.

### Parameters for vector formats

For pdf format, clicking **Next** will display the dialog shown in figure 6.34 (this is only the case if the graphics is using more than one page).

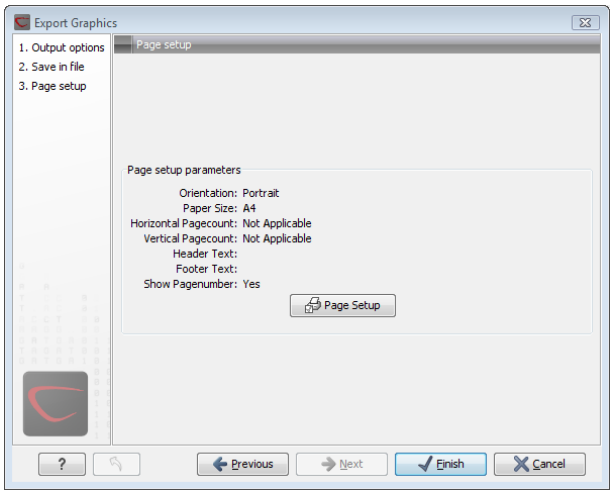



Figure 6.34: *Page setup parameters for vector formats.*

The settings for the page setup are shown, and clicking the **Page Setup** button will display a dialog where these settings can ba adjusted. This dialog is described in section 5.2.

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.

### 6.5.4 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.

## 6.6 Export graph data points to a file

Data points for graphs displayed along the sequence or along an alignment, mapping or BLAST result, can be exported to a semicolon-separated text file (csv format). An example of such a graph is shown in figure 6.35. This graph shows the coverage of reads of a read mapping (produced with *CLC Genomics Workbench*).

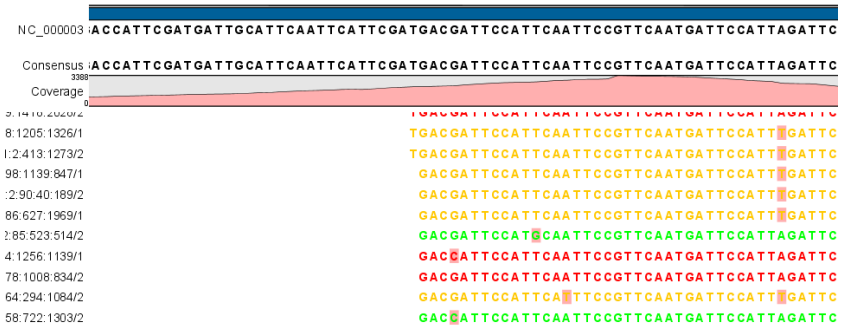



Figure 6.35: *A graph displayed along the mapped reads. Right-click the graph to export the data points to a file.*

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 6.36 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.

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.

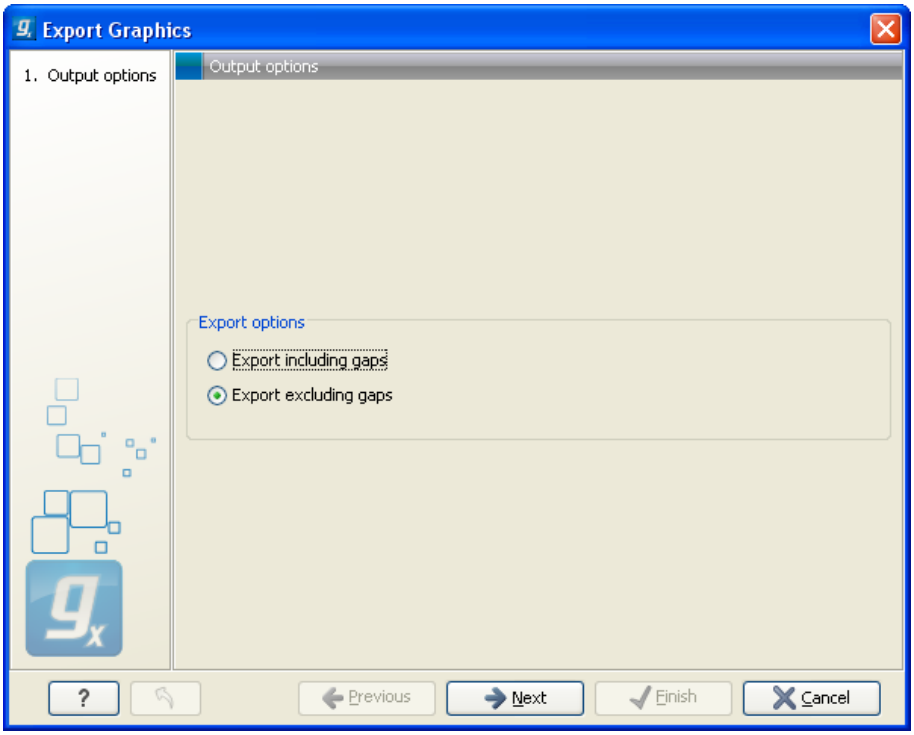



Figure 6.36: *Choosing to include data points with gaps*

The output format of the file is like this:

```
"Position";"Value";
"1";"13";
"2";"16";
"3";"23";
"4";"17";
...
```

## 6.7 Copy/paste view output

The content of tables, e.g. in 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 which is useful if you use programs like Microsoft Word and Excel. There is a huge number of 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.

First step is to select the desired elements in the view:

> **click a line in the Folder Content view** | **hold Shift-button** | **press arrow down/up key**

See figure 6.37.

When the elements are selected, do the following to copy the selected elements:

> **right-click one of the selected elements** | **Edit** | **Copy (  )**

Then:

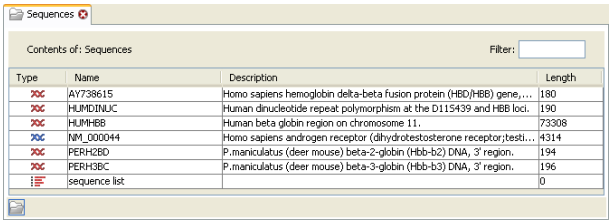



Figure 6.37: *Selected elements in a Folder Content view.*

**right-click in the cell A1 | Paste ( )**

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

# History log

**Contents**


7.1 Element history . . . . . 136
7.1.1 Sharing data with history . . . . . 137


*CLC Genomics Workbench* keeps a log of all operations you make in the program. If e.g. you rename a sequence, align sequences, create a phylogenetic tree or translate a sequence, you can always go back and check what you have done. In this way, you are able to document and reproduce previous operations.

This can be useful in several situations: It can be used for documentation purposes, where you can specify exactly how your data has been created and modified. It can also be useful if you return to a project after some time and want to refresh your memory on how the data was created. Also, if you have performed an analysis and you want to reproduce the analysis on another element, you can check the history of the analysis which will give you all parameters you set.

This chapter will describe how to use the **History** functionality of *CLC Genomics Workbench*.

## 7.1 Element history

You can view the history of all elements in the **Navigation Area** except files that are opened in other programs (e.g. Word and pdf-files). The history starts when the element appears for the first time in *CLC Genomics Workbench*. To view the history of an element:

**Select the element in the Navigation Area** | **Show (** **) in the Toolbar** | **History (** **)**

or **If the element is already open** | **History (** **) at the bottom left part of the view**

This opens a view that looks like the one in figure 7.1.

When an element's history is opened, the newest change is submitted in the top of the view. The following information is available:

- **Title**. The action that the user performed.
- **Date and time**. Date and time for the operation. The date and time are displayed according

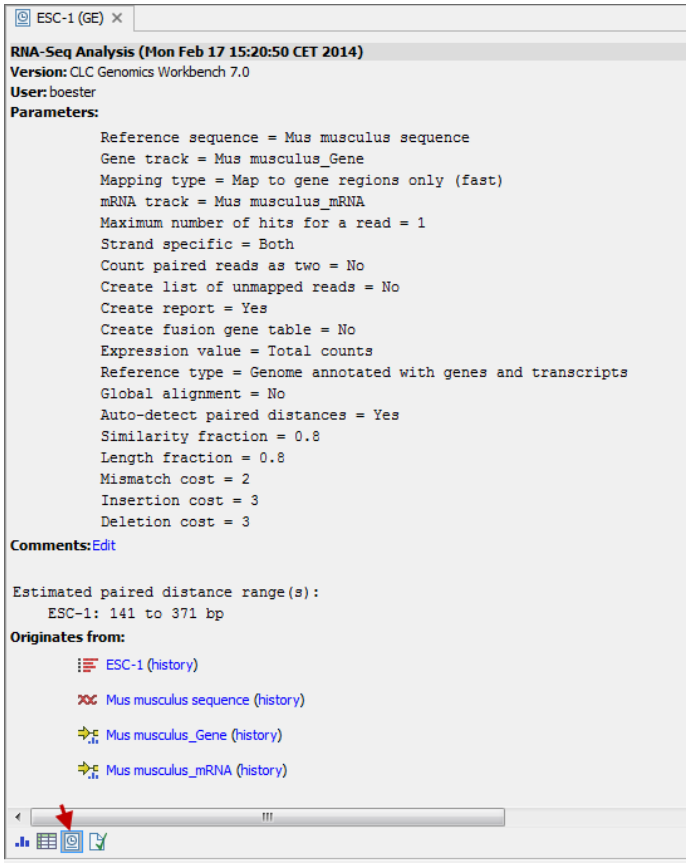



Figure 7.1: *An element's history.*

to your locale settings (see section 4.1).

- **Version**. The Workbench type and version that has been used.
- **User**. The user who performed the operation. If you import some data created by another person in a CLC Workbench, that person's name will be shown.
- **Parameters**. Details about the action performed. This could be the parameters that were chosen for an analysis.
- **Comments**. By clicking **Edit** you can enter your own comments regarding this entry in the history. These comments are saved.
- **Originates from**. This information is usually shown at the bottom of an element's history. Here, you can see which elements the current element originates from. If you have e.g. created an alignment of three sequences, the three sequences are shown here. Clicking the element selects it in the **Navigation Area**, and clicking the 'history' link opens the element's own history.

### 7.1.1 Sharing data with history

The history of an element is attached to that element, which means that exporting an element in CLC format (*.clc) will export the history too. In this way, you can share folders and files with others while preserving the history. If an element's history includes source elements (i.e. if

there are elements listed in 'Originates from'), they must also be exported in order to see the full history. Otherwise, the history will have entries named "Element deleted". An easy way to export an element with all its source elements is to use the **Export Dependent Elements** function described in section 6.4.2.

The history view can be printed. To do so, click the **Print** icon (  ). The history can also be exported as a pdf file:

> **Select the element in the Navigation Area | Export (  ) | in "File of type" choose History PDF | Save**

# Chapter 8

# Batching and result handling

**Contents**


**8.1 Batch processing . . . . . 139**
8.1.1 Batch overview . . . . . 140
8.1.2 Batch filtering and counting . . . . . 141
8.1.3 Setting parameters for batch runs . . . . . 141
8.1.4 Running the analysis and organizing the results . . . . . 142
**8.2 How to handle results of analyses . . . . . 142**
8.2.1 Table outputs . . . . . 142
8.2.2 Batch log . . . . . 143
**8.3 Working with tables . . . . . 144**
8.3.1 Filtering tables . . . . . 145


## 8.1 Batch processing

Most of the analyses in the **Toolbox** are able to perform the same analysis on several elements in one batch. This means that analyzing large amounts of data is very easily accomplished. As an example, if you use the **Find Binding Sites and Create Fragments ( )** tool available in *CLC Genomics Workbench* and supply five sequences as shown in figure 8.1, the result table will present an overview of the results for all five sequences.

This is because the input sequences are pooled before running the analysis. If you want individual outputs for each sequence, you would need to run the tool five times, or alternatively use the **Batching mode**.

Batching mode is activated by clicking the **Batch** checkbox in the dialog where the input data is selected. Batching simply means that each data set is run separately, just as if the tool has been run manually for each one. For some analyses, this simply means that each input sequence should be run separately, but in other cases it is desirable to pool sets of files together in one run. This selection of data for a batch run is defined as a **batch unit**.

When batching is selected, the data to be added is the folder containing the data you want to batch. The content of the folder is assigned into batch units based on this concept:

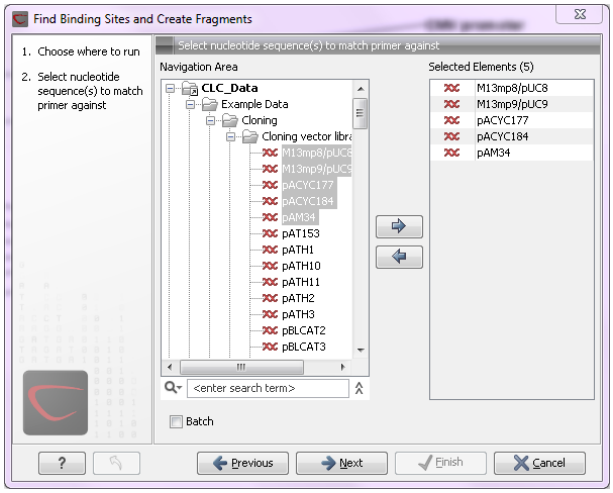



Figure 8.1: *Inputting five sequences to Find Binding Sites and Create Fragments.*

- All subfolders are treated as individual batch units. This means that if the subfolder contains several input files, they will be pooled as one batch unit. Nested subfolders (i.e. subfolders within the subfolder) are ignored.
- All files that are not in subfolders are treated as individual batch units.

An example of a batch run is shown in figure 8.2.

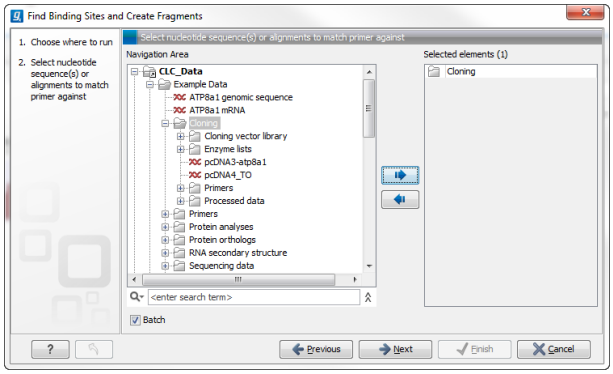



Figure 8.2: *The Cloning folder includes both folders and sequences.*

The `Cloning` folder that is found in the example data (see section 1.6.2) contains two sequences ( ) and four folders ( ). If you click **Batch**, only folders can be added to the list of selected elements in the right-hand side of the dialog. To run the contents of the Cloning folder in batch, double-click to select it.

When the Cloning folder is selected and you click **Next**, a batch overview is shown.

### 8.1.1 Batch overview

The batch overview lists the batch units to the left and the contents of the selected unit to the right (see figure 8.3).

In this example, the two sequences are defined as separate batch units because they are located at the top level of the Cloning folder. There were also four folders in the Cloning folder (see

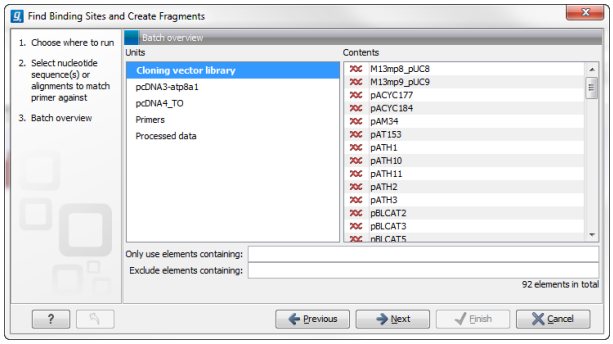



Figure 8.3: *Overview of the batch run.*

figure 8.2), and three of them are listed as well. This means that the contents of these folders are pooled in one batch run (you can see the contents of the `Cloning vector library` batch run in the panel at the right-hand side of the dialog). The reason why the `Enzyme lists` folder is not listed as a batch unit is that it does not contain any sequences.

In this overview dialog, the Workbench has filtered the data so that only the types of data accepted by the tool is shown (DNA sequences in the example above).

### 8.1.2 Batch filtering and counting

At the bottom of the dialog shown in figure 8.3, the Workbench counts the number of files that will be run in total (92 in this case). This is counted across all the batch units.

In some situations it is useful to filter the input for the batching based on names. As an example, this could be to include only paired reads for a mapping, by only allowing names where "paired" is part of the name.

This is achieved using the **Only use elements containing** and **Exclude elements containing** text fields. Note that the count is dynamically updated to reflect the number of input files based on the filtering.

If a complete batch unit should be removed, you can select it, right-click and choose **Remove Batch Unit**. You can also remove items from the contents of each batch unit using right-click and **Remove Element**.

### 8.1.3 Setting parameters for batch runs

For some tools, the subsequent dialogs depend on the input data. In this case, one of the units is specified as parameter prototype and will be used to guide the choices in the dialogs. Per default, this will be the first batch unit (marked in bold), but this can be changed by right-clicking another batch unit and click **Set as Parameter Prototype**.

Note that the Workbench is validating a lot of the input and parameters when running in normal "non-batch" mode. When running in batch, this validation is not performed, and this means that some analyses will fail if combinations of input data and parameters are not right. Therefore batching should only be used when the batch units are very homogenous in terms of the type and size of data.

### 8.1.4 Running the analysis and organizing the results

At the last dialog before clicking **Finish**, it is only possible to use the **Save** option. When a tool is run in batch mode, the default behavior is to place the result files in the same folder as the input files. In the example shown in figure 8.3, the result of the two single sequences will be placed in the Cloning folder, whereas the results for the `Cloning vector library` and `Processed data` runs will be placed inside these folders.

However, there is an option to save the results in a separate folder structure by checking **Into separate folders**. This will allow you to specify a new save destination, and the *CLC Genomics Workbench* will create a subfolder for each batch unit where the results are saved..

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 of this is different between Workbench and Server:

- When running the batch job in the 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 is done in order to avoid many parallel analyses that would draw on the same compute resources and slow down the computer.
- When this is run on a CLC Server (see `http://clcbio.com/server`), all the processes are placed in the queue, and the queue is then taking care of distributing the jobs. This means that if the server set-up includes multiple nodes, the jobs can be run in parallel.

If you need to stop the whole batch run, you need to stop the "master" process.

## 8.2 How to handle results of analyses

This section will explain how results generated from tools in the Toolbox are handled by *CLC Genomics Workbench*. Note that this also applies to tools not running in batch mode (see above). All the analyses in the **Toolbox** are performed in a step-by-step procedure. First, you select elements for analyses, and then there are a number of steps where you can specify parameters (some of the analyses have no parameters, e.g. when translating DNA to RNA). The final step concerns the handling of the results of the analysis, and it is almost identical for all the analyses so we explain it in this section in general.

In this step, shown in figure 8.4, you have two options:

- **Open.** This will open the result of the analysis in a view. This is the default setting.
- **Save.** This means that the result will not be opened but saved to a folder in the **Navigation Area**. If you select this option, click **Next** and you will see one more step where you can specify where to save the results (see figure 8.5). In this step, you also have the option of creating a new folder or adding a location by clicking the buttons (  )/ (  ) at the top of the dialog.

### 8.2.1 Table outputs

Some analyses also generate a table with results, and for these analyses the last step looks like figure 8.6.

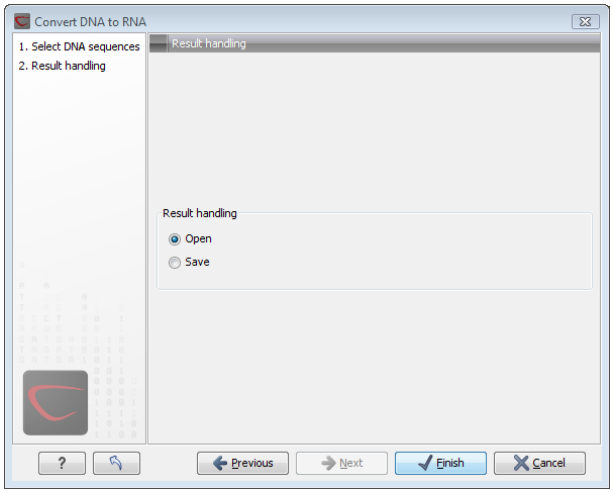



Figure 8.4: *The last step of the analyses exemplified by Translate DNA to RNA.*

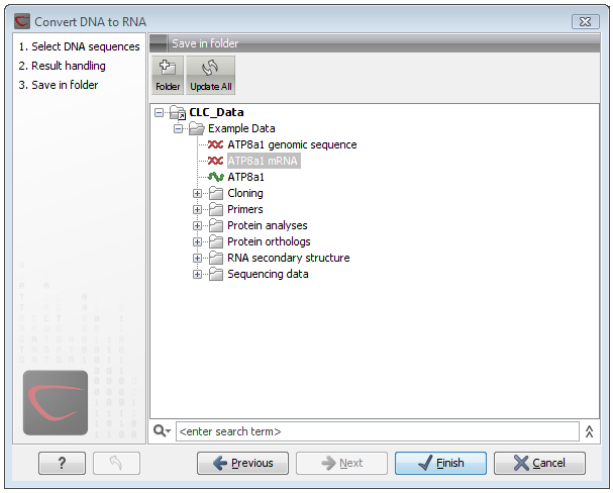



Figure 8.5: *Specify a folder for the results of the analysis.*

In addition to the **Open** and **Save** options you can also choose whether the result of the analysis should be added as annotations on the sequence or shown on a table. If both options are selected, you will be able to click the results in the table and the corresponding region on the sequence will be selected.

If you choose to add annotations to the sequence, they can be removed afterwards by clicking **Undo** ( ) in the **Toolbar**.

### 8.2.2 Batch log

For some analyses, there is an extra option in the final step to create a log of the batch process (see e.g. figure 8.6). This log will be created in the beginning of the process and continually updated with information about the results. See an example of a log in figure 8.7. In this example, the log displays information about how many open reading frames were found.

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.

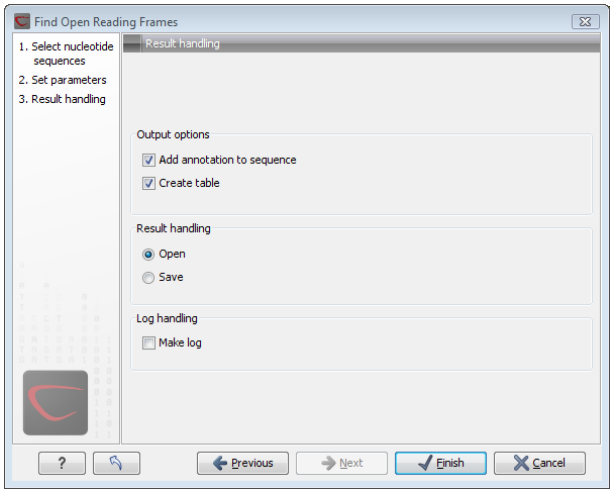



Figure 8.6: *Analyses which also generate tables.*

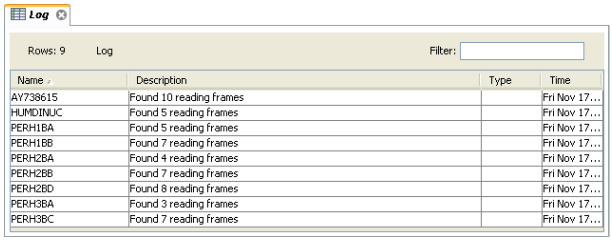



Figure 8.7: *An example of a batch log when finding open reading frames.*

## 8.3 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 8.8 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.

### Table viewing options

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

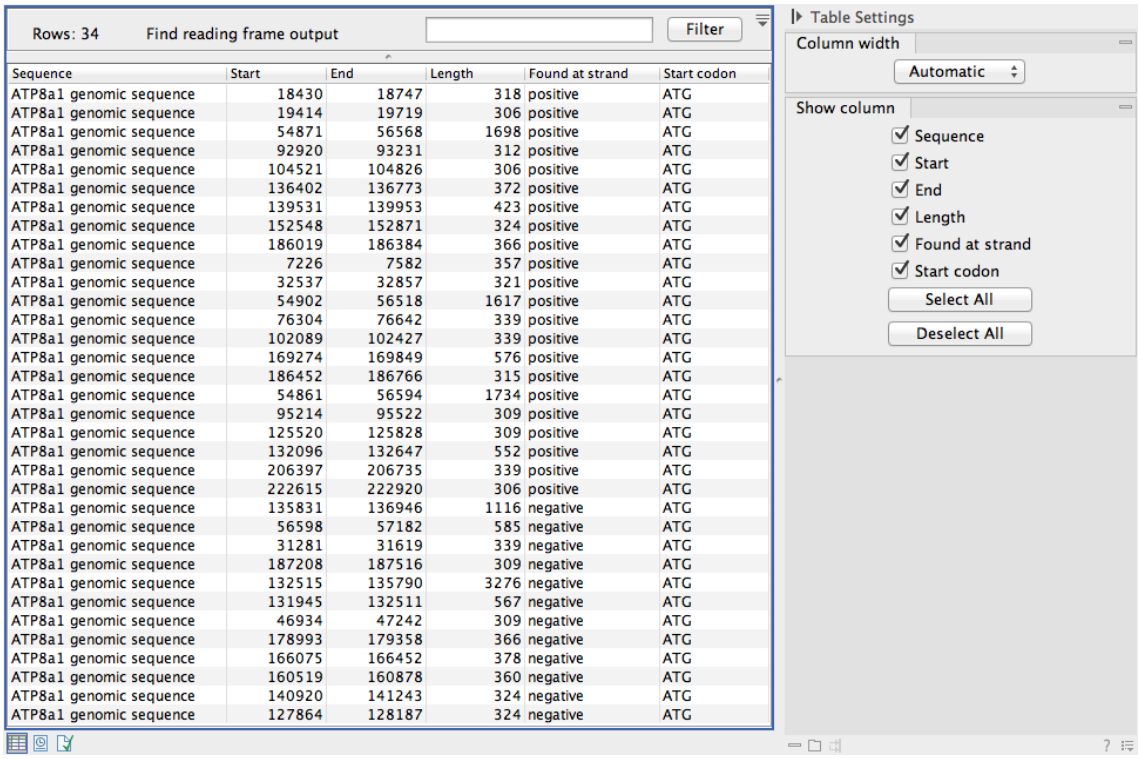
For example, the columns that can be dispalyed in the table are listed in the section called **Show column**. The checkboxes allow you to see or hide any of the available columns for that table.

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 by adjusted manually. When the table content exceeds the size of the viewing area, a horizontal scroll becomes available for navigation across the columns.

### Sorting tables

You can **sort** table according to the values of a particular column by clicking a column header. (Pressing Ctrl - ⌘ on Mac - while you click will refine the existing sorting).

Clicking once will sort in ascending order. A second click will change the order to descending. A



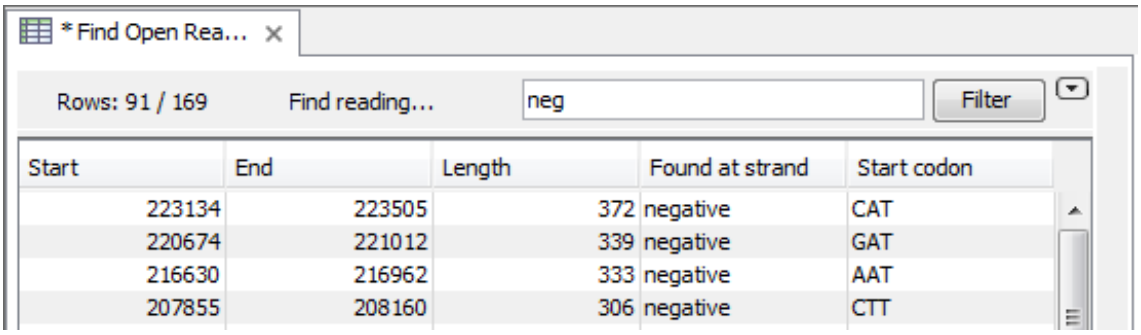

| Sequence | Start | End | Length | Found at strand | Start codon |
|---|---|---|---|---|---|
| ATP8a1 genomic sequence | 18430 | 18747 | 318 | positive | ATG |
| ATP8a1 genomic sequence | 19414 | 19719 | 306 | positive | ATG |
| ATP8a1 genomic sequence | 54871 | 56568 | 1698 | positive | ATG |
| ATP8a1 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 | 186452 | 186766 | 315 | positive | ATG |
| ATP8a1 genomic sequence | 54861 | 56594 | 1734 | positive | ATG |
| ATP8a1 genomic sequence | 95214 | 95522 | 309 | positive | ATG |
| ATP8a1 genomic sequence | 125520 | 125828 | 309 | positive | ATG |
| ATP8a1 genomic sequence | 132096 | 132647 | 552 | positive | ATG |
| ATP8a1 genomic sequence | 206397 | 206735 | 339 | positive | ATG |
| ATP8a1 genomic sequence | 222615 | 222920 | 306 | positive | ATG |
| ATP8a1 genomic sequence | 135831 | 136946 | 1116 | negative | ATG |
| ATP8a1 genomic sequence | 56598 | 57182 | 585 | negative | ATG |
| ATP8a1 genomic sequence | 31281 | 31619 | 339 | negative | ATG |
| ATP8a1 genomic sequence | 187208 | 187516 | 309 | negative | ATG |
| ATP8a1 genomic sequence | 132515 | 135790 | 3276 | negative | ATG |
| ATP8a1 genomic sequence | 131945 | 132511 | 567 | negative | ATG |
| ATP8a1 genomic sequence | 46934 | 47242 | 309 | negative | ATG |
| ATP8a1 genomic sequence | 178993 | 179358 | 366 | negative | ATG |
| ATP8a1 genomic sequence | 166075 | 166452 | 378 | negative | ATG |
| ATP8a1 genomic sequence | 160519 | 160878 | 360 | negative | ATG |
| ATP8a1 genomic sequence | 140920 | 141243 | 324 | negative | ATG |
| ATP8a1 genomic sequence | 127864 | 128187 | 324 | negative | ATG |

Figure 8.8: *A table showing the results of an open reading frames analysis.*

third click will set the order back its original order.

### 8.3.1 Filtering tables

The final concept to introduce is **Filtering**. The table filter as an advanced and a simple mode. The simple mode is the default and is applied simply by typing text or numbers (see an example in figure 8.9).[1]




| Start | End | Length | Found at strand | Start codon |
|---|---|---|---|---|
| 223134 | 223505 | 372 | negative | CAT |
| 220674 | 221012 | 339 | negative | GAT |
| 216630 | 216962 | 333 | negative | AAT |
| 207855 | 208160 | 306 | negative | CTT |

Figure 8.9: *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 (also the ones that are not shown). 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.

[1]Note that for tables with more than 10000 rows, you have to actually click the **Filter** button for the table to take effect.

However, if you wish to make use of numerical information or make more complex filters, you can switch to the advanced mode by clicking the **Advanced filter** ( ) button. The advanced filter is structure in a different way: First of all, you can have more than one criterion in the 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 case-insensitive)

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

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

Both for the simple and the advanced filter, there is a counter at the upper left corner which tells you the number of rows that pass the filter (91 in figure 8.9 and 15 in figure 8.10).

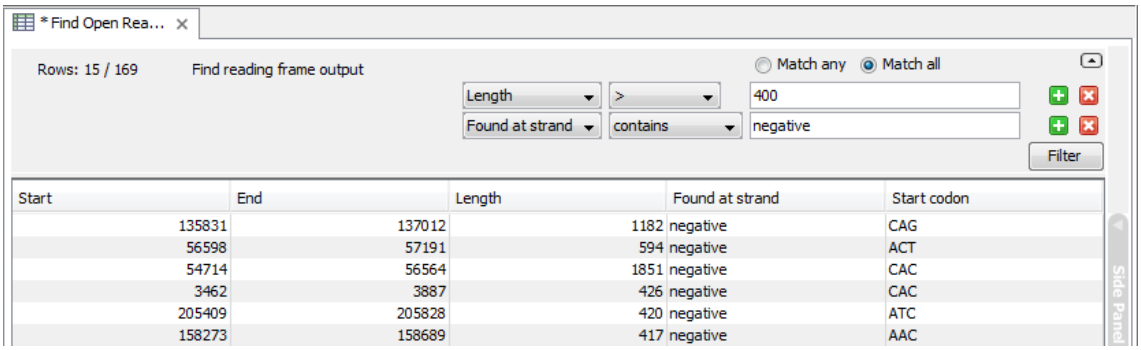



Figure 8.10: *The advanced filter showing open reading frames larger than 400 that are placed on the negative strand.*

# Chapter 9

# Workflows

## Contents


**9.1 Creating a workflow . . . . . 149**
9.1.1 Adding workflow elements . . . . . 149
9.1.2 Configuring workflow elements . . . . . 150
9.1.3 Locking and unlocking parameters . . . . . 151
9.1.4 Connecting workflow elements . . . . . 152
9.1.5 Input and output . . . . . 155
9.1.6 Layout . . . . . 158
9.1.7 Input modifying tools . . . . . 158
9.1.8 Workflow validation . . . . . 160
9.1.9 Workflow creation helper tools . . . . . 161
9.1.10 Adding to workflows . . . . . 163
9.1.11 Snippets in workflows . . . . . 163
9.1.12 Supported data flows . . . . . 164
**9.2 Distributing and installing workflows . . . . . 165**
9.2.1 Creating a workflow installation file . . . . . 166
9.2.2 Installing a workflow . . . . . 170
9.2.3 Workflow identification and versioning . . . . . 172
9.2.4 Automatic update of workflow elements . . . . . 172
**9.3 Executing a workflow . . . . . 173**
**9.4 Open copy of installed workflow . . . . . 174**


The *CLC Genomics Workbench* provides a framework for creating, distributing, installing and running workflows. Workflows created in the Workbench can also be installed on a *CLC Genomics Server*.

A workflow consists of a series of connected tools where the output of one tool is used as input for another tool. In this way you create a workflow that for example makes a read mapping, uses the mapped reads as input for variant detection, and performs filtering of the variant track. Once the workflow is set up, it can be installed (either in your own Workbench or on a Server or it can be sent to a colleague). In that way it becomes possible to analyze lots of samples using the same standard pipeline, the same reference data and the same parameters.

This chapter will first explain how to create a new workflow, and next go into details about the installation and execution of a workflow. For information about installing a workflow on the *CLC Genomics Server*, please see the user manual at http://www.clcbio.com/usermanuals.

## 9.1 Creating a workflow

A workflow can be created by pressing the "Workflows" button ( ) in the toolbar and then selecting "New Workflow..." ( ).

Alternatively, a workflow can be created via the menu bar:

**File** | **New** | **Workflow ( )**

This will open a new view with a blank screen where a new workflow can be created.

### 9.1.1 Adding workflow elements

First, click the **Add Element** ( ) button at the bottom (or use the shortcut Shift + Alt + E). This will bring up a dialog that lists the elements and tools, which can be added to a workflow (see figure 9.1).

Alternatively elements can be dragged directly from the **Toolbox** into the workflow. Not all elements are workflow enabled. This means that only workflow enabled elements can be dropped in the workflow.

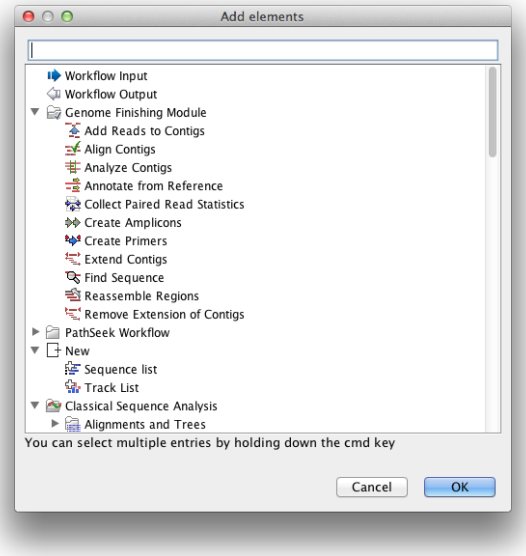



Figure 9.1: *Adding elements in the workflow.*

Elements that can be selected in the dialog are mostly tools from the Toolbox. However, there are two special elements on the list; the elements that are used for input and output. These two elements are explained in section 9.1.5.

You can select more than one element in the dialog by pressing Ctrl (⌘ on Mac) while selecting. Click OK when you have selected the relevant tools (you can always add more later on).

You will now see the selected elements in the editor (see figure 9.2).

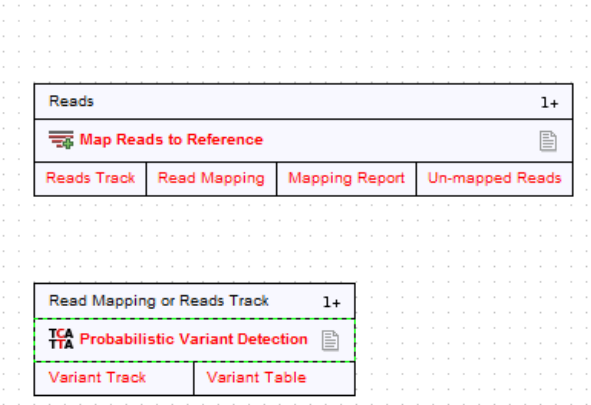



Figure 9.2: *Read mapping and variant calling added to the workflow.*

Once added, you can move and re-arrange the elements by dragging with the mouse (grab the box with the name of the element).

### 9.1.2 Configuring workflow elements

Each of the tools can be configured by right-clicking the name of the tool as shown in figure 9.3.

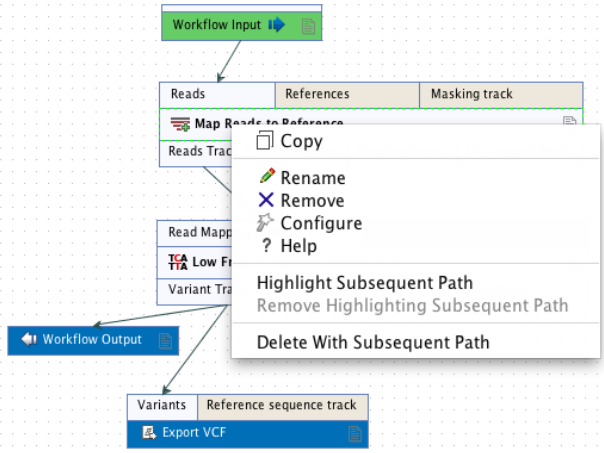



Figure 9.3: *Configuring a tool.*

The first option you are presented with is the option to **Rename** the element. This is for example useful when you wish to discriminate several copies of the same tool in a workflow. The name of the element is also visible as part of the process description when the workflow is executed. Right click on the tool in the workflow and select "Rename" or click on the tool in the workflow and use the F2 key as a shortcut.

With the **Remove** option, elements can be removed from the workflow. The shortcut Alt + Shift + R removes all elements from the workflow.

You can also **Configure** the tool from the right click menu or alternatively it can be done by double-clicking the element. This will open a dialog with options for setting parameters, selecting reference data etc. An example is shown in figure 9.4.

Click through the dialogs using **Next** and press **Finish** when you are done. This will save the

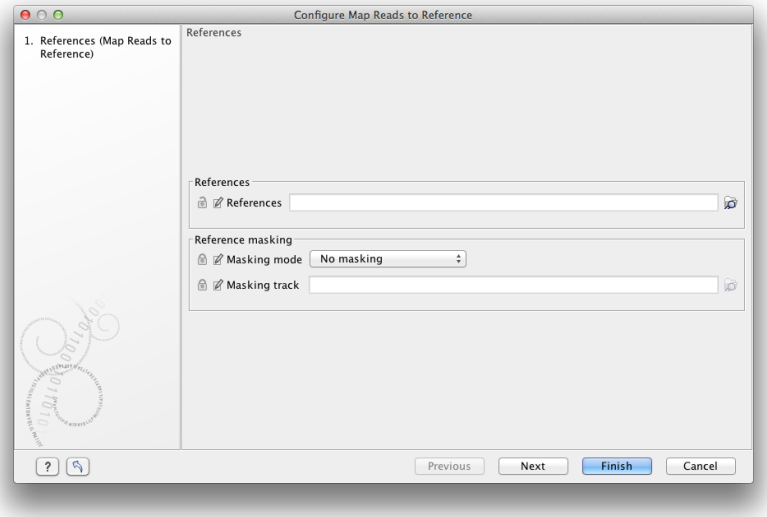



Figure 9.4: *Configuring read mapper parameters.*

parameter settings that will then be applied when the workflow is executed.

You can also change the name of the parameter into something that fits the vocabulary of the users that are intended to execute the workflow. This is done by clicking the edit icon ( ) and entering a new name.

Note that reference data are a bit special. In the example with the read mapper in figure 9.3, you have to define a reference genome. This is done by pointing to data in the **Navigation Area**. If you distribute the workflow and install it in a different setting where this data is not accessible, the installation procedure will involve defining the new reference data to use (e.g. the reference genome sequence for read mapping). This is explained in more detail in section 9.2.

In some workflows, many elements use the same reference data, and there is a quick way of configuring all these: right-click the empty space and choose **Configure All References**. This will show a dialog listing all reference data needed by the workflow.

The lock icons in the dialog are used for specifying whether the parameter should be locked and unlocked as described in the next section. Hereby it is possible to lock so the workflow runs with the same parameters like references(s) every time.

Once an element has been configured, the workflow element gets a darker color to make it easy to see which elements have been configured.

With **Highlight Subsequent Path** the path from Name of the tool that was clicked on and further downstream will be highlighted whereas all other elements will be grayed out (figure 9.5). The **Remove Highlighting Subsequent Path** reverts the highlighting to the normal workflow layout.

Instead of configuring the various tools individually, the **Configuration Editor** enable specification of all settings, references, masking parameters etc. through a single wizard window (figure 9.6). The Editor is accessed through the ( ) icon located lower left corner.

### 9.1.3 Locking and unlocking parameters

Figure 9.7 shows the different stages in a workflow.

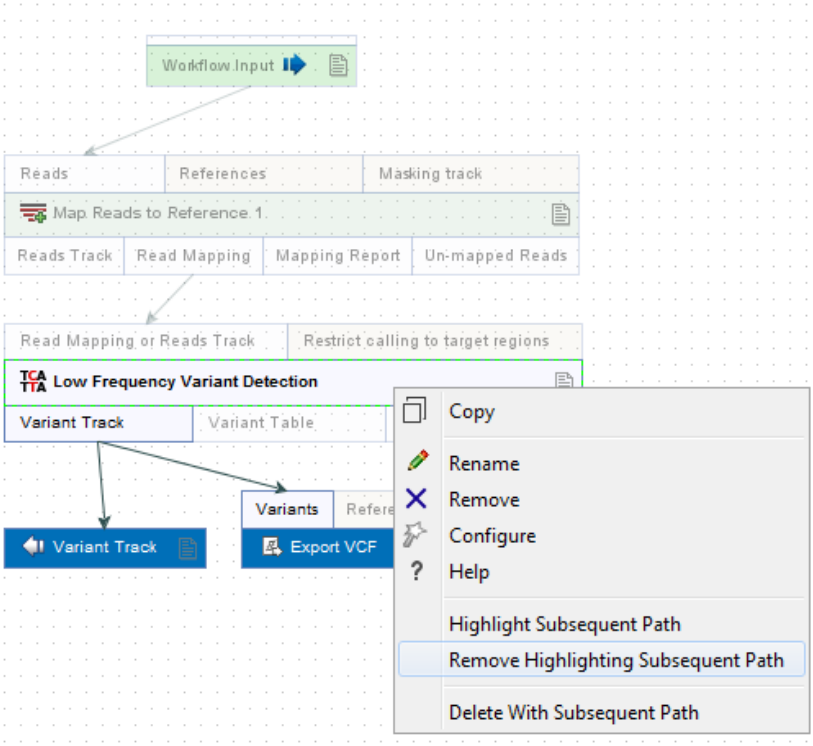



Figure 9.5: *Highlight path from the selected tool and downstream.*

At the top, the workflow creation is illustrated. Workflow creation is explained above. Next, the workflow can be installed in a Workbench or Server (explained in section 9.2). Subsequently, the workflow can be executed as any other tool in the **Toolbox**.

At the creation step, the workflow creator can specify which parameters should be locked or unlocked. If a parameter is locked, it means that it cannot be changed neither in the installation nor the execution step. The lock icons shown in figure 9.4 specifies whether the parameter should be open or locked.

If the parameter is left open, it is possible to adjust it as part of the installation (see section 9.2). Furthermore, it can also be locked at this stage.

Parameters that are left open both from the workflow creation and installation, will be available for adjustment when the workflow is executed.

Please note that data parameters per default are marked as unlocked. When installing the workflow somewhere else, the connection to the data needs to be re-established, and this is only possible when the parameter is unlocked. 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.

### 9.1.4 Connecting workflow elements

Figure 9.8 explains the different parts of a workflow element.

At the top of each element a description of the required type of input is found. In the right-hand side, a symbol specifies whether the element accepts multiple incoming connections, e.g. `+1`

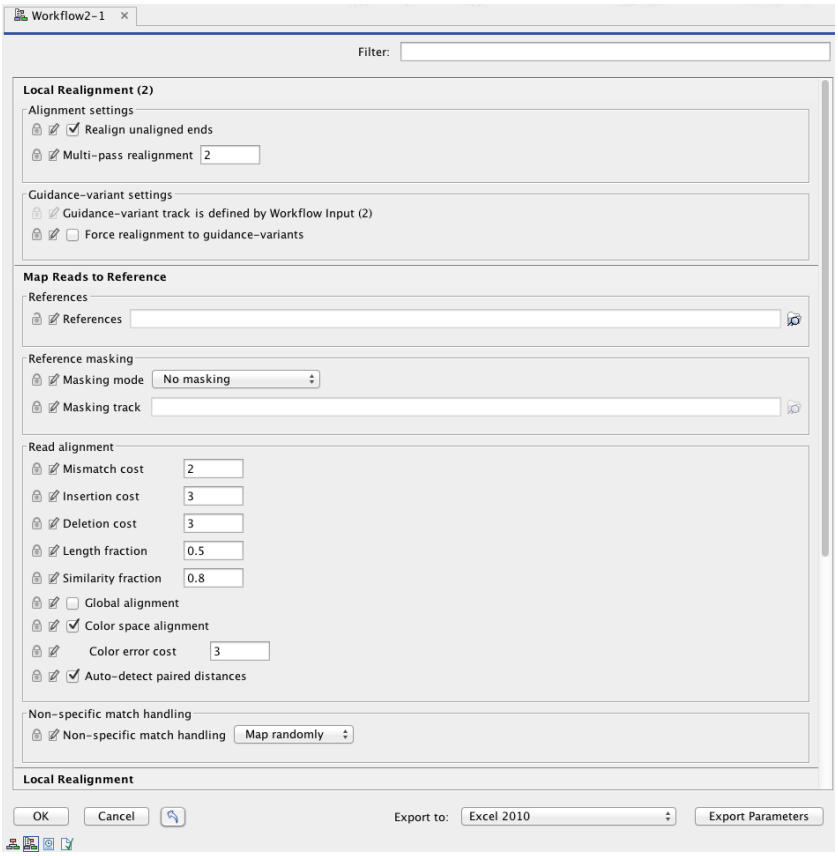



Figure 9.6: *The Configuration Editor enable setting of all configurable tools through a single window.*

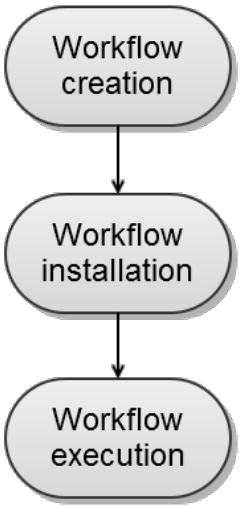



Figure 9.7: *The life cycle of a workflow.*

means that more than one output can be connected, and no symbol means that only one can be connected. At the bottom of each element there are a number of small boxes that represent the different kinds of output that is produced. In the example with the read mapper shown in figure 9.2, the read mapper is able to produce a reads track, a report etc.

Each of the output boxes can be connected to further analysis in three ways:

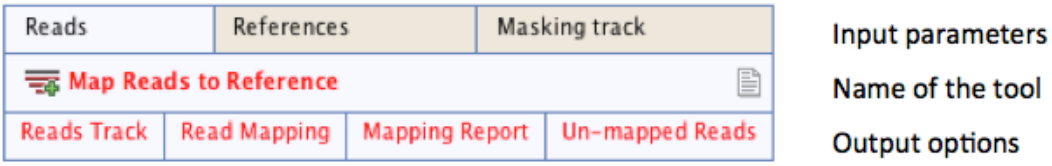



Figure 9.8: *A workflow element consists of three parts: input, name of the tool, and output.*

- By dragging with the mouse from the output into the input box of the next element. This is shown in figure 9.9. A green border around the box will tell you when the mouse button can be released, and an arrow will connect the two elements (see figure 9.10).
- Right-clicking the output box will display a list of the possible elements that this output could be connected to. You can also right-click the input box of an element and connect this to a matching output of another element.
- Alternatively, if the element to connect to is not already added, you can right-click the output and choose **Add Element to be Connected**. This will bring up the dialog from figure 9.1, but only showing the tools that accepts this particular output. Selecting a tool will both add it to the workflow and connect with the output you selected. You can also add an upstream element of workflow in the same way by right-clicking the input box.

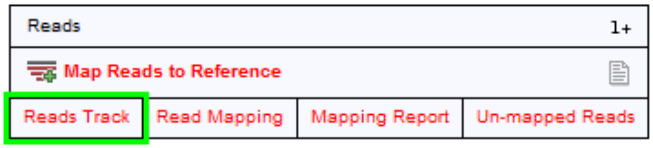



Figure 9.9: *Dragging the reads track output with the mouse.*

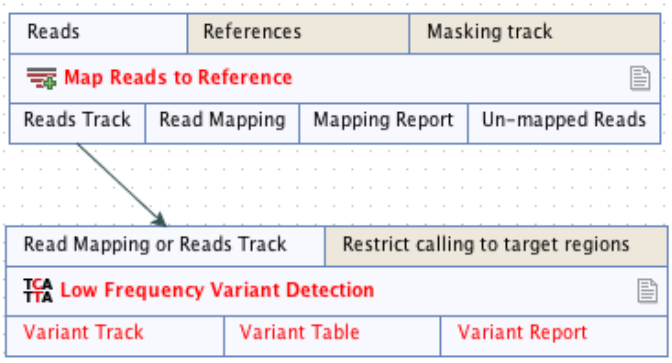



Figure 9.10: *The reads track is now used for variant calling.*

All the logic of combining output and input is based on matching the type of input. So the read mapper creates a reads track and a report as output. The variant caller accepts reads tracks as input but not mapping reports. This means that you will not be able to connect the mapping report to the variant caller.

Figure 9.11 demonstrates how one tool can receive input from two different sources; 1) a reads track that is the input that hold the data that is to be analyzed (in this case reads that is to be locally realigned), and 2) a parameter that can have different functions depending on the tool

that it is connected to (in this case the InDel track is used as a guidance track for the local realignment. In other situations the parameter track could be used for e.g. annotation or could provide a reference sequence).

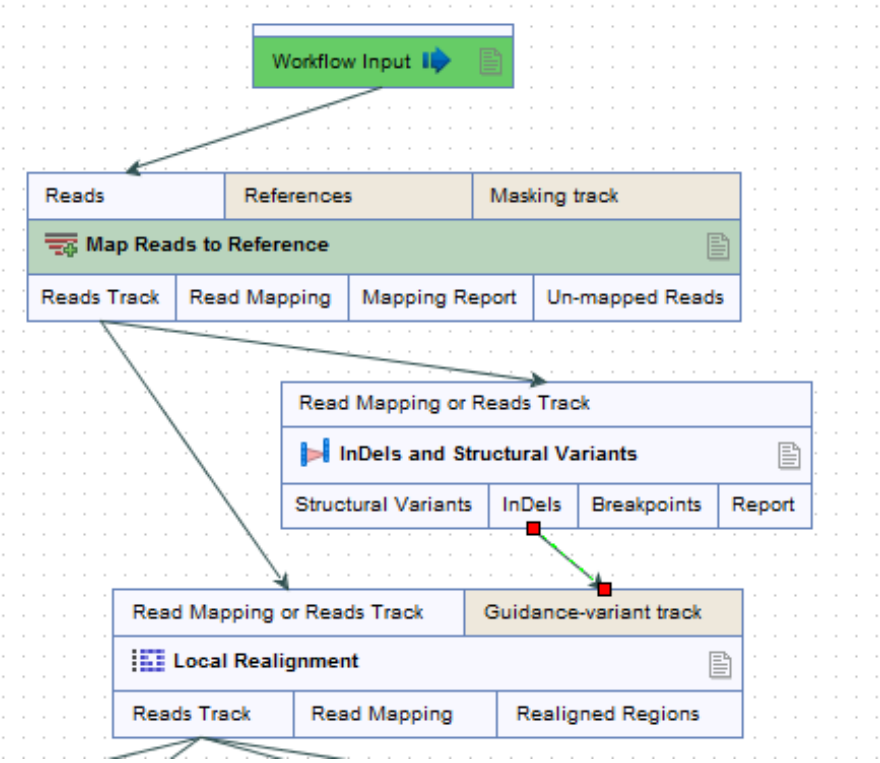



Figure 9.11: *A tool can receive input from both the generated output from another tool (in this example a reads track) and from a parameter (in this case InDels detected with the InDels and Structural Variants tool).*

### 9.1.5 Input and output

Besides connecting the elements together, you have to decide what the input and the output of the workflow should be. We will first look at specification of the output, which is done by right-clicking the output box of any tool and selecting **Use as Workflow Output** as shown in figure 9.12.

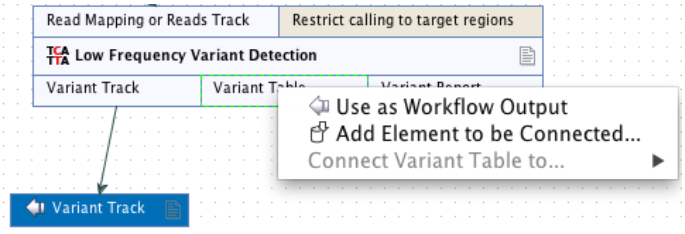



Figure 9.12: *Selecting a workflow output.*

You can mark several outputs this way throughout the workflow. Note that no intermediate results

are saved unless they are marked as workflow output[1].

By double-clicking the output box, you can specify how the result should be named as shown in figure 9.13.

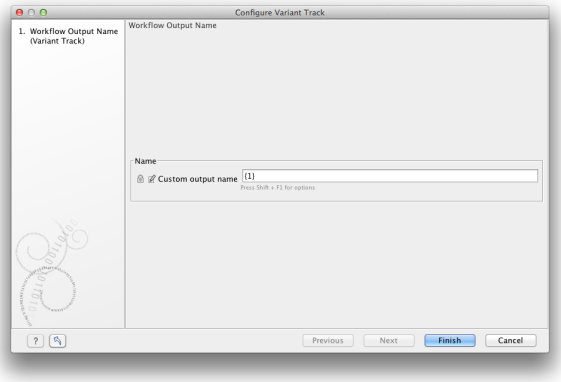



Figure 9.13: *Specifying naming of a workflow output.*

In this dialog you can enter a name for the output result, and you can make use of two dynamic placeholders for creating this name (press Shift + F1 to get assistance):

- {1} Represents the default name of the result. When running the tool outside of a workflow, this is the name given to the result.
- {2} Represents the name of the workflow input (not the input to this particular tool but the input to the entire workflow).

An example of a meaningful name to a variant track could be `{2} variant track` as shown in figure 9.14. If your workflow input is named `Sample 1`, the result would be `Sample 1 variant track`.

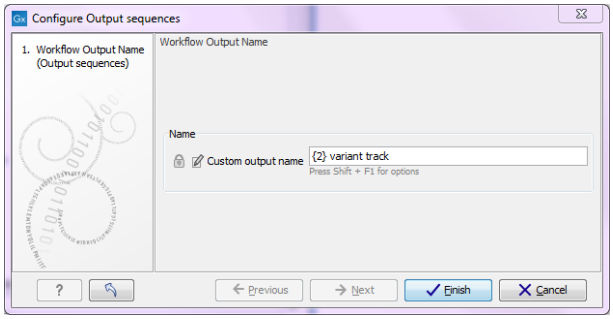



Figure 9.14: *Providing a custom name for the result.*

In addition to output, you also have to specify where the data should go into the workflow by adding an element called **Workflow Input**. This can be done by:

[1]When the workflow is executed, all the intermediate results are indeed saved temporarily but they are automatically deleted when the workflow is completed. If a part of the workflow fails, the intermediate results are not deleted.

- Right-clicking the input box of the first tool and choosing **Connect to Workflow Input**. By dragging from the workflow input box to other input boxes several tools can use the input data directly.
- Pressing the button labeled **Add Element** (or right-click somewhere in the workflow background area and select **Add Element** from the menu that appears). The input box must then be connected to the relevant tool(s) in the workflow by dragging from the Workflow Input box to the "input description" part of the relevant tool(s) in the workflow.

At this point you have only prepared the workflow for receiving input data, but not specified which data to use as input. To be able to do this you must first save the workflow. When this has been done, the button labeled **Run** is enabled which allows you to start executing the workflow. When you click on the button labeled **Run** you will be asked to provide the input data.

Multiple input files can be used when:

- Data is generated within the Workflow
- Data is held within the Workbench
- Data is a combination of the two situations above

Please note that once the multiple input feature is used in a workflow, it is not possible to run the workflow in batch mode. Please also note that the input elements can be renamed, which is useful if you want to be able to discriminate the elements in the process description shown during workflow execution.

The example in figure 9.15 shows how to generate a track list in a workflow. It is possible to include reference tracks and also tracks from any step, or multiple steps within the workflow as long as output Read Tracks are defined and linked to.

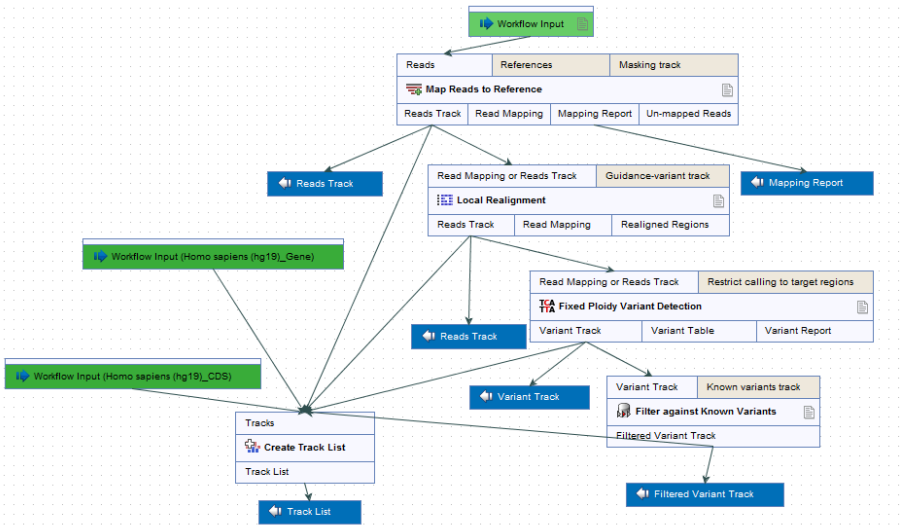



Figure 9.15: *Generation of a track list including data generated within the Workflow, as well as data held in the Workbench.*

### 9.1.6 Layout

The workflow layout can be adjusted automatically. Right clicking in the workflow editor will bring up a pop-up menu with the option "Layout". Click on "Layout" to adjust the layout of the selected elements (Figure 9.16). Only elements that have been connected will be adjusted.

**Note!** The layout can also be adjusted with the quick command Shift + Alt + L.

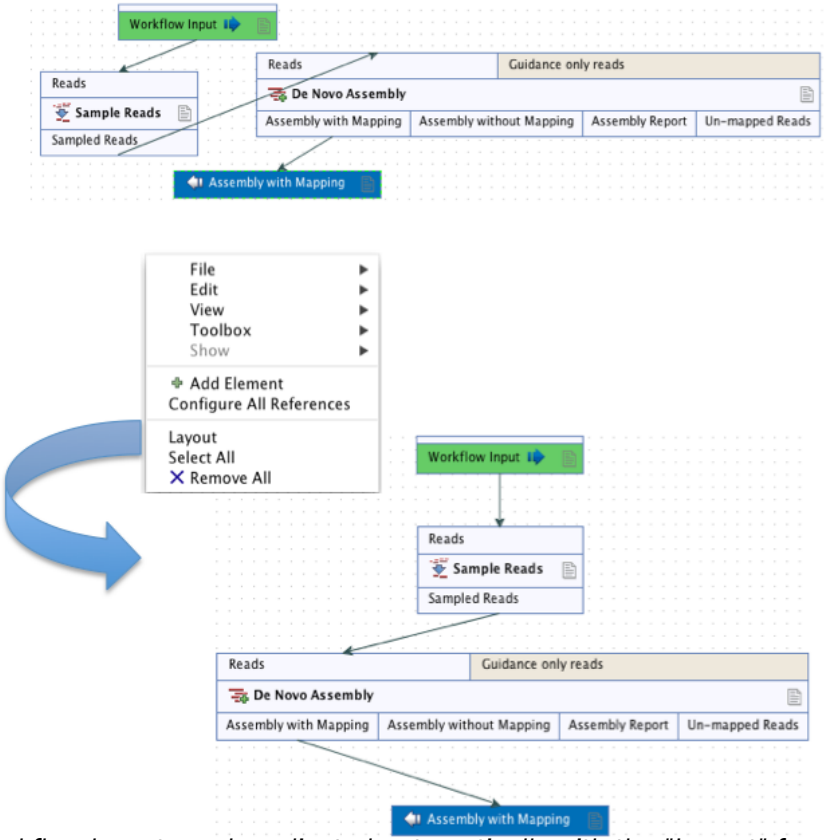



Figure 9.16: *A workflow layout can be adjusted automatically with the "Layout" function.*

**Note!** 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 e.g. an email or a text or presentation program.

### 9.1.7 Input modifying tools

An input modifying tool is a tool that manipulates its input objects (e.g. adds annotations) without producing a new object. This behavior differs from the rest of the tools and requires special handling in the workflow.

In the workflow an input modifying tool is marked with the symbol ( ) (figure 9.17).

Restrictions apply to workflows that contain input modifying tools. For example, branches are not allowed where one of the elements is a modifying tool (see figure 9.18), as it cannot be guaranteed which workflow branch will be executed first, which in turn means that different runs




Figure 9.17: *Input modifying tools are marked with the letter M.*

can result in production of different objects. Hence, if a workflow is constructed with a branch where one of the succeeding elements is a modifying tool, a message in red letters will appear saying "Branching before a modifying tool can lead to non-deterministic behavior". In such a situation the "Run" and "Create Installer" buttons will be disabled (figure 9.18).

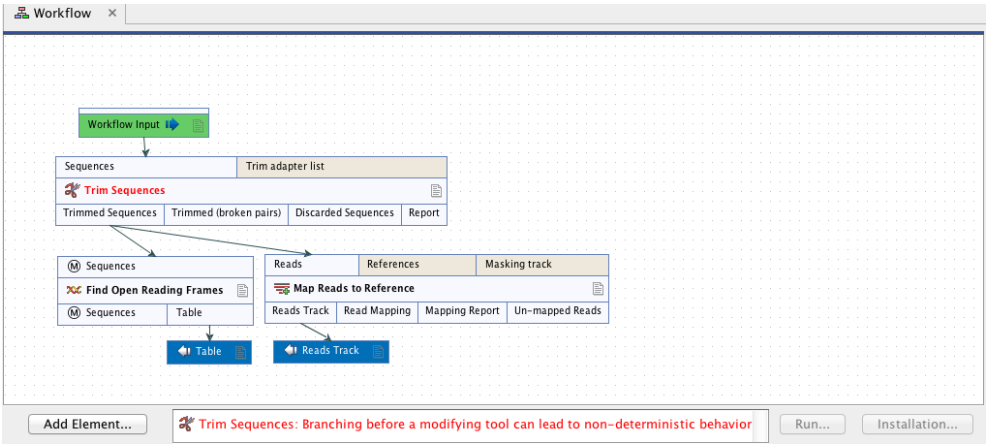



Figure 9.18: *A branch containing an input modifying tool is not allowed in a workflow.*

The problem can be solved by resolving the branch by putting the elements in the right order (with respect to order of execution). This is shown in figure 9.19 that also shows that the "Run" and "Create Installer" buttons are now enabled. In addition, a message in green letters has appeared saying "Validation successful".

As input modifying tools only modify existing objects without producing a new object, it is not possible to add a workflow output element directly after an input modifying tool (figure 9.20). A workflow output element can only be added when other tools than input modifying tools are included in the workflow.

If the situation occur where more input modifying tools are used succeedingly, a copy of the object will be created in addition to using the modified object as input at the next step of the chain (see figure 9.21). In order to see this output you must right click on the output option (marked with a red arrow in figure 9.21) and select "Use as Workflow Output".

When running a workflow where a workflow output has been added after the first input modifying tool in the chain (see figure 9.22) 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. When running this workflow you will be able to see the copy of the output from the first input modifying tool in the

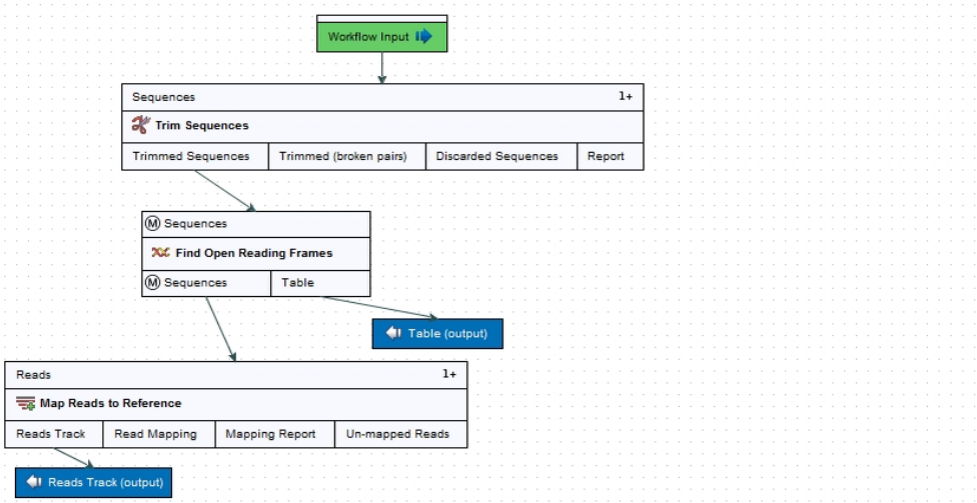



Figure 9.19: *A branch containing an input modifying tool has been resolved and the workflow can now be run or installed.*

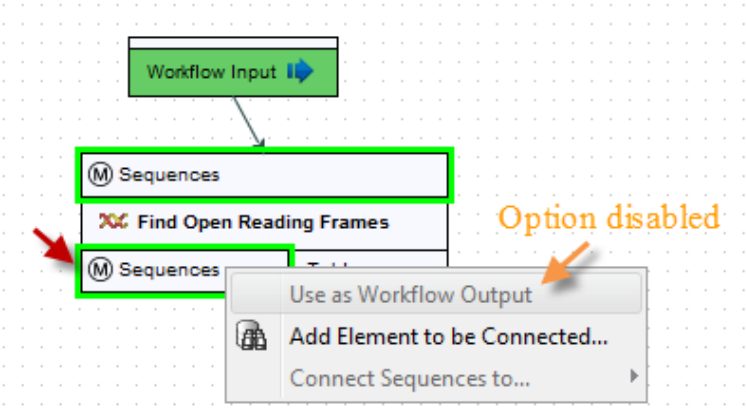



Figure 9.20: *A workflow output element cannot be added if the workflow only contains an input modifying tool.*

**Navigation Area** (at the destination that you selected when running the workflow).

### 9.1.8 Workflow validation

At the bottom of the view, there is a text with a status of the workflow (see figure 9.23). It will inform about the actions you need to take to finalize the workflow.

The validation may contain several lines of text. Scroll the list to see more lines. If one of the errors pertain to a specific element in the workflow, clicking the error will highlight this element.

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.

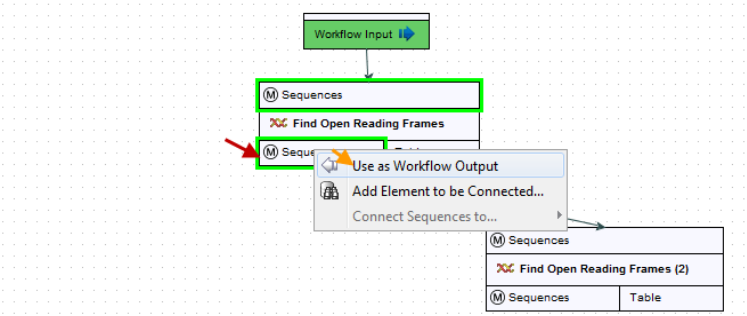



Figure 9.21: *A workflow output element can be added when more than one input modifying tool is used succeedingly (despite that the workflow only contains input modifying tools). Select "Use as Workflow Output" to make a copy of the output.*

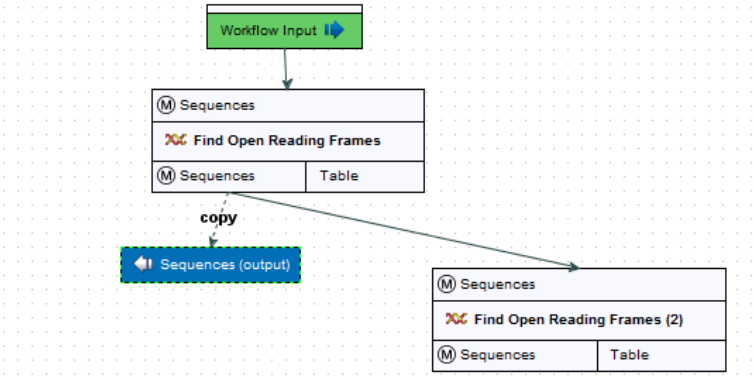



Figure 9.22: *A workflow output element can be added when more than one input modifying tool is used succeedingly (despite that the workflow only contains input modifying tools). Note that this output is marked with "copy" to indicate that this is a copy of the result that is used as input at the next level in the chain.*

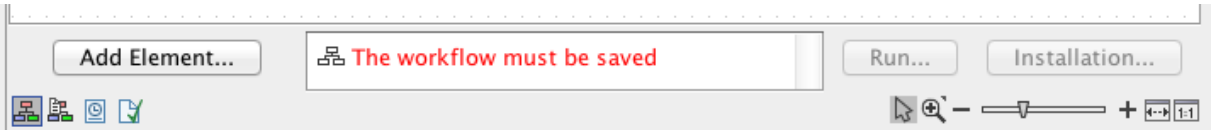



Figure 9.23: *A workflow is constantly validated at the bottom of the view.*

- Additional checks that the workflow is consistent.

Once these conditions are fulfilled, the status will be "Validation successful", the **Run** button is enabled. Clicking this 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 9.1.2), there will be a dialog asking for this as part of the test run.

### 9.1.9 Workflow creation helper tools

In the workflow editor **Side Panel**, you will find the following workflow display settings that can be useful to know (figure 9.24):

#### Grid

- Enable grid You can display a grid and control the spacing and color of the grid. Per

default, the grid is shown, and the workflow elements snap to the grid when they are moved around.

**View mode**

- Collapsed The elements of the workflow can be collapsed to allow a cleaner view and especially for large workflows this can be useful.
- Highlight used elements Ticking **Highlight used elements** (or using the shortcut Alt + Shift + U) will show all elements that are used in the workflow whereas unused elements are grayed out.
- Rulers Vertical and horizontal rules can be visualised
- Auto Layout Ticking **Auto Layout** will ensure rearrangement of elements once new elements are added.
- Connections to background Connecting arrows are shown behind elements. This may easy reading of element names and accessible parameters.

**Design**

- Round elements Enable rounding of the element boxes.
- Show shadow Shadows of element boxes can be added.
- Configured elements Background color can be customized.
- Inpupt elements Background color can be customized.
- Edges Color of connecting arrows can be customized.

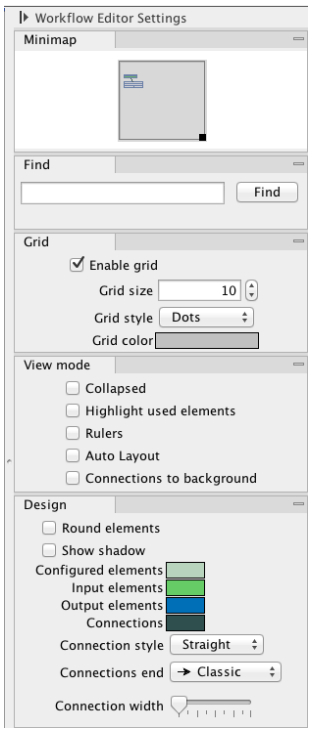



Figure 9.24: *The Side Panel of the workflow editor.*

### 9.1.10 Adding to workflows

Additional elements can be added to an already existing workflow by dragging it from the navigation area into the workflow editor and joining more elements as necessary. The new workflow must be saved and validated before it can be executed. Two or more workflows can be joined by dragging and dropping one from the Navigation Area, into another that is already open in the main viewing area. The output of one must be connected to the input of the next to allow the whole workflow to run in one go.

Workflows do not need to be valid to be dragged in to the workflow editor, but they must have been migrated to the current version of the workbench.

### 9.1.11 Snippets in workflows

When creating a new workflow, you will often have a number of connected elements that are shared between workflows. Instead of building workflows from scratch it is possible to reuse components of an existing workflow. These components are called snippets and can exist of e.g. a read mapper and a variant caller.

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 9.25.

When you have clicked on "Install as snippet" the dialog shown in figure 9.26 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 on the button labeled **OK**. This will install your snippet and the installed snippet will now appear in the **Side Panel** under the "Snippets" tab (see figure 9.27)

Right-clicking on the installed snippet in the **Side Panel** will bring up the following options (figure 9.28):

- **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.
- **Migrate** Migrates the snippet (if migration is required).

If you right-click on the top-level folder you get the options shown in figure 9.29:

- **Create new group** Creates a new folder under the selected folder.

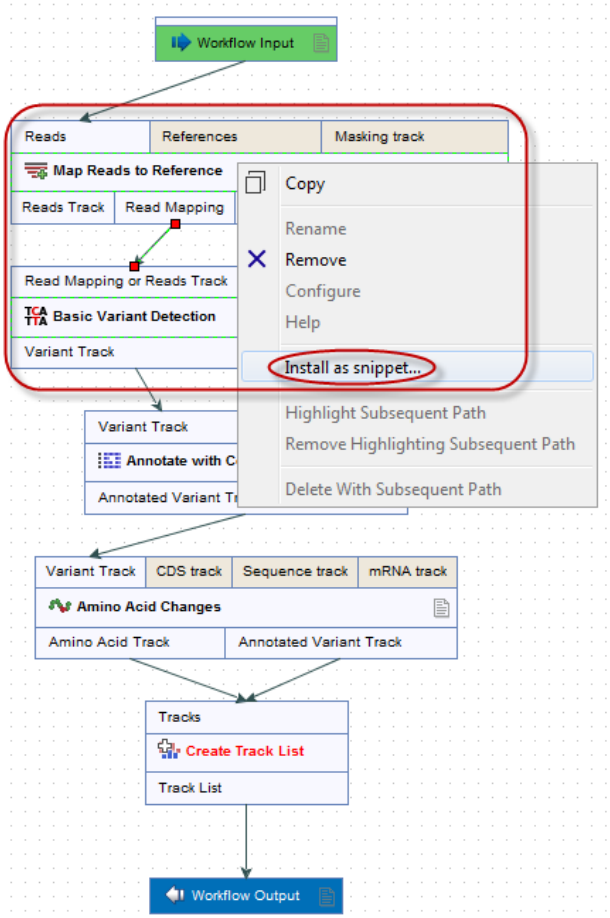



Figure 9.25: *The selected elements are highlighted with a red box in this figure. Select "Install as snippet".*

- **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 elements" option that is shown in figure 9.30.

### 9.1.12 Supported data flows

The current version of the workflow framework supports single-sample workflows. This means processing one sample through various analysis steps. When it comes to comparative analysis, this has to be done outside the workflow.

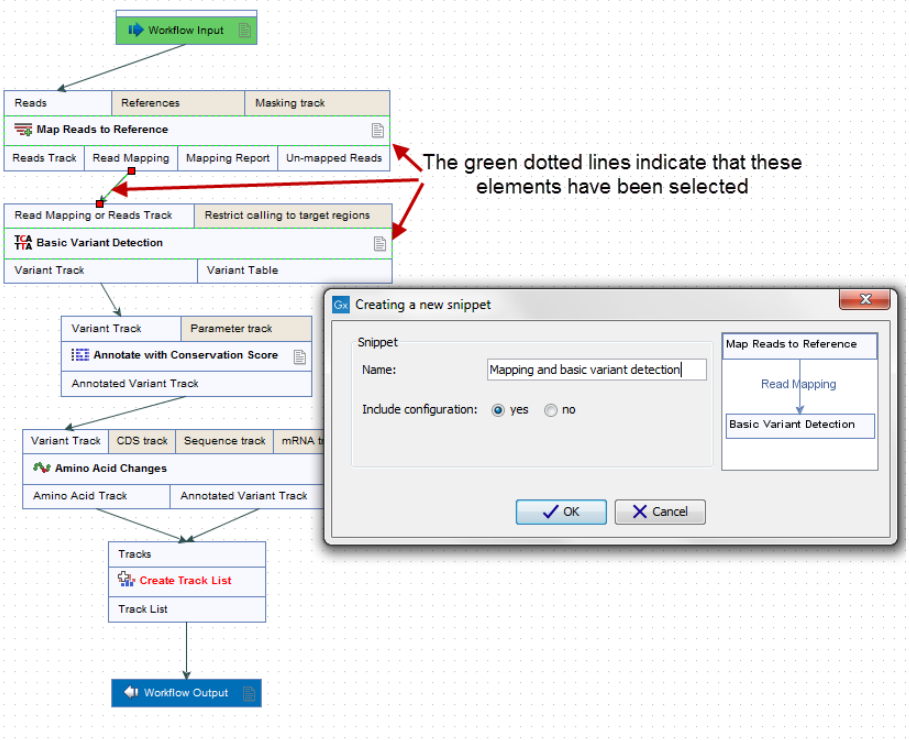



Figure 9.26: *In the "Create a new snippet" dialog you can name the snippet and select whether or not you would like to include the configuration. In the right-hand side of the dialog you can see the elements that are included in the snippet.*

A typical example that would explain how this works is a trio analysis study where you want to compare variants found in a child with those from the mother and father. For this, you would create a workflow including mapping, variant detection, variant annotation and maybe some quality control. All three samples would be processed through this workflow in batch mode (see section 9.3). At the end, you can manually create a track list with all the relevant tracks (reads and variants) and run the trio analysis tool manually.

Since all the comparative tools are relatively quick, the bulk of the computation work can usually be incorporated into the workflow which can take care of the more tedious parts of the manual work involved.

## 9.2 Distributing and installing workflows

Once the workflow has been configured, you can use the **Run** button (see section 9.1.8) to process data through the workflow, but the real power of the workflow is its ability to be distributed and installed in the **Toolbox** alongside the other tools that come with the *CLC Genomics Workbench*, as well as the ability to install the same workflow on a *CLC Genomics Server*. The mechanism for distributing the workflow is a workflow installer file which can be created from the workflow editor and distributed and installed in any Workbench or Server.

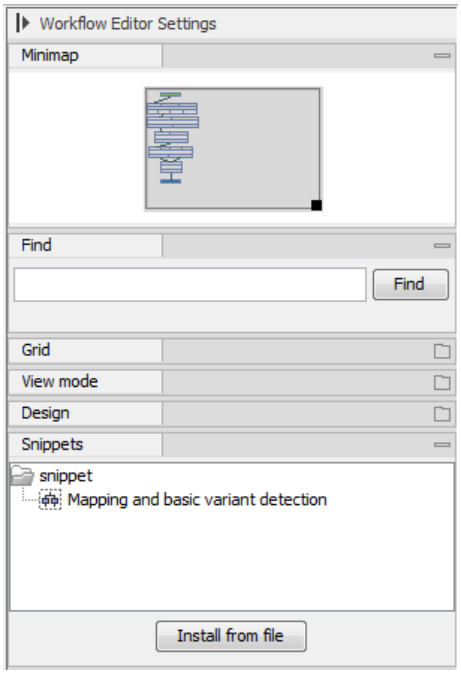



Figure 9.27: *When a snippet is installed, it appears in the Side Panel under the "Snippets" tab.*

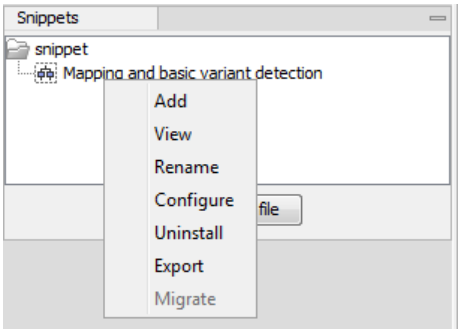



Figure 9.28: *Right-clicking on an installed snippet brings up a range of different options.*

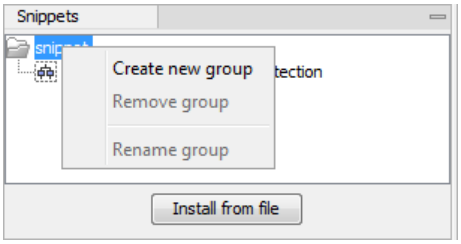



Figure 9.29: *Right-clicking on the snippet top-level folder makes it possible to manipulate the groups.*

### 9.2.1 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 with information from a CLC bio workflow in figure 9.31).

**Author information** Providing name, email and organization of the author of the workflow. This will be visible for users installing the workflow and will enable them to look up the source of the workflow any time. The organization name is important because it is part of the

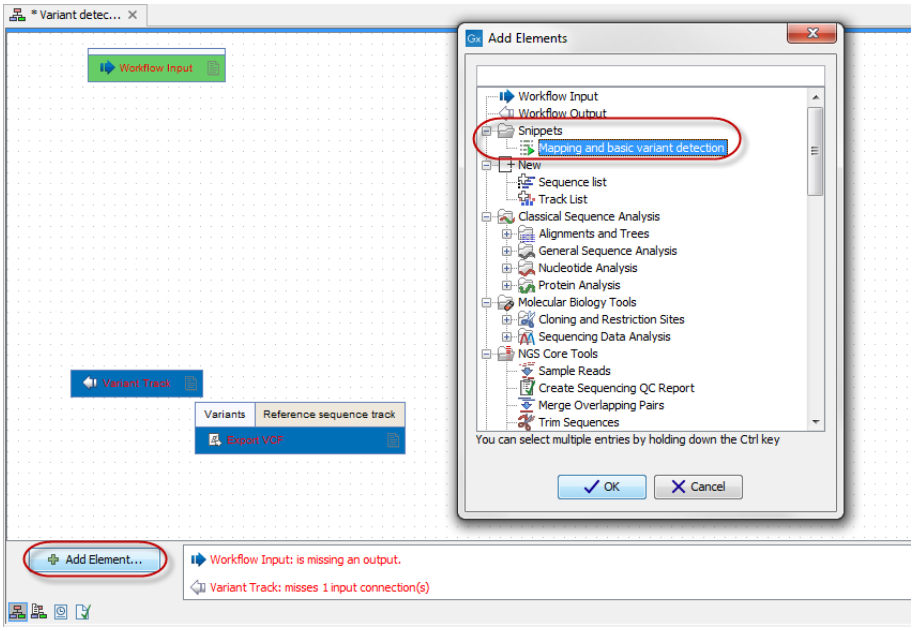



Figure 9.30: *Snippets can be added to a workflow in the workflow editor using the 'Add Elements' button found in the lower left corner.*

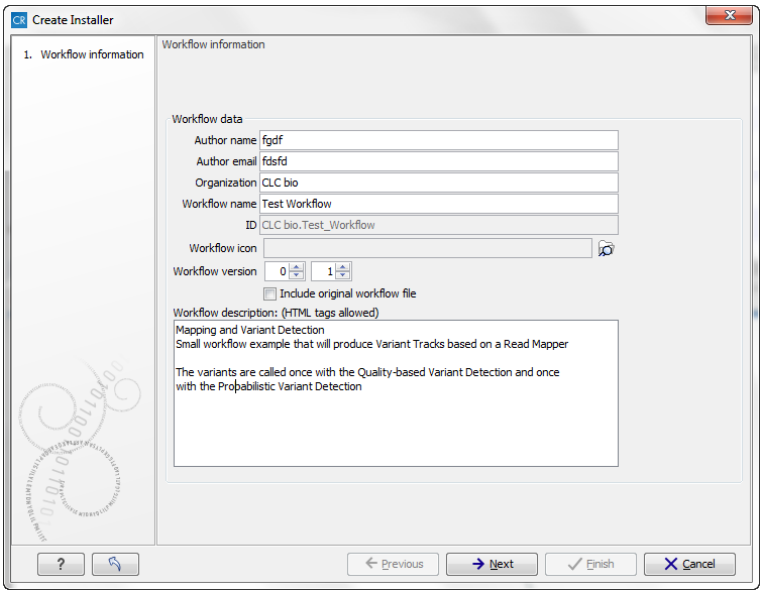



Figure 9.31: *Workflow information for the installer.*

workflow id (see more in section 9.2.3)

**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 9.2.3). 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. 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.

**Include original workflow file** This will include the design file to be included with the installer. Once the workflow is installed in a workbench, you can extract the original workflow file and modify it.

**Workflow description** Provide a textual description of the workflow. This will be displayed for users when they have installed the workflow. Simple HTML tags are allowed (should be HTML 3.1 compatible, see http://www.w3.org/TR/REC-html32).

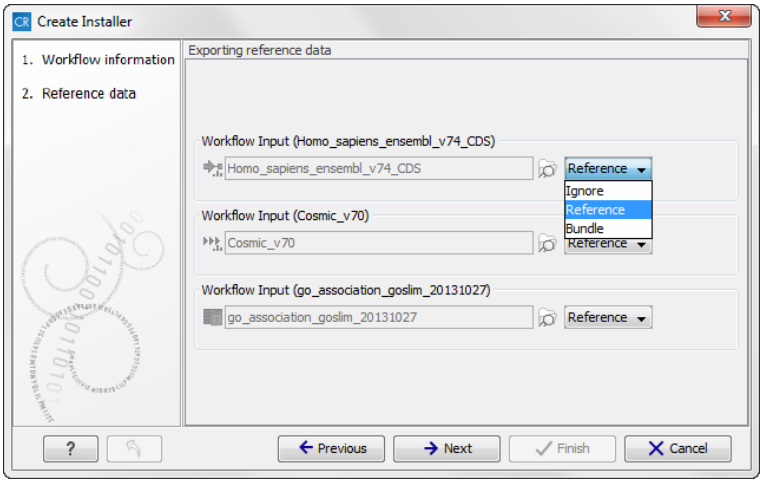



Figure 9.32: *Bundling data with the workflow installer.*

If you configured any of the workflow elements with data, clicking **Next** will give you the following three options (see figure 9.32):

- **Ignore** This will exclude these reference data from the workflow.
- **Reference** This option can be used to include reference data from a shared CLC_References directory in a workflow without bundling the reference data with the workflow. Instead he reference data is included in the workflow by pointing at the shared CLC_References directory. This is particularly useful when working with large reference data.
- **Bundle** Includes data in the workflow by bundling the reference data with the workflow. **Note!** Bundling data should only be used to bundle small data sets with the workflow installer.

Click **Next** and you will be asked to specify where to install the workflow (figure 9.33). 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 writeable persistence location. Should this location not suit your needs, you can always move it afterwards, using the normal persistence operations.

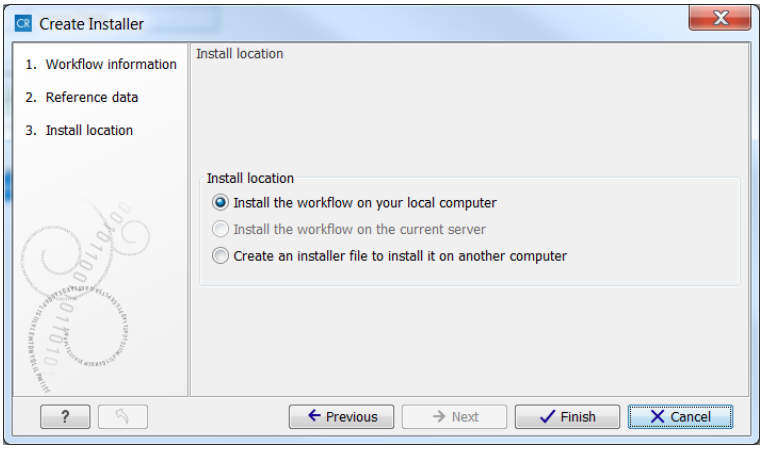



Figure 9.33: *Select whether the workflow should be installed on your local computer or on the current server. A third option is to create an installer file (.cpw) that can be installed on another computer.*

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 9.34) 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.

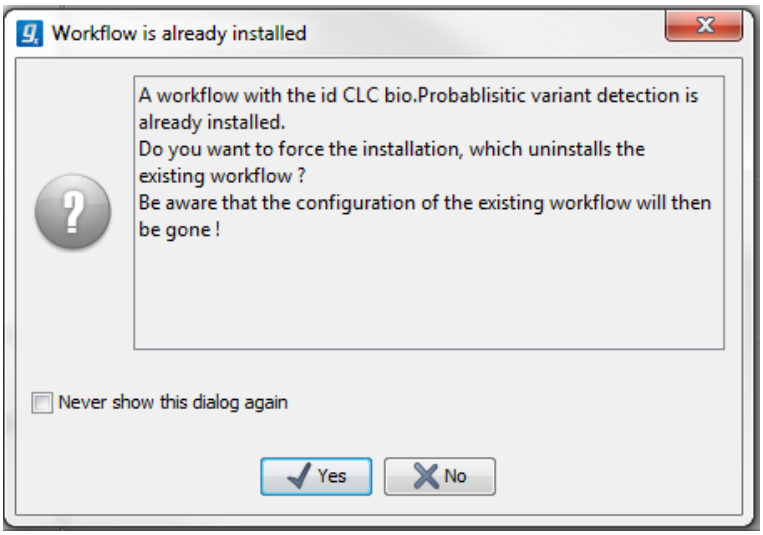



Figure 9.34: *Select whether you wish to force the installation of the workflow or keep the original workflow.*

### 9.2.2 Installing a workflow

Workflows are installed in the workflow manager (for information about installing a workflow on the *CLC Genomics Server*, please see the user manual at http://www.clcbio.com/usermanuals):

**Help | Manage Workflows ( )**

or press the "Workflows" button ( ) in the toolbar and then select "Manage Workflow..." ( ).

This will display a dialog listing the installed workflows. To install an existing workflow, click **Install from File** and select a workflow .cpw file .

Once installed, it will appear in the workflow manager as shown in figure 9.35.

If the workflow was bundled with data, installing it on the workbench will ask you for a location to put the bundled data. Installing a workflow with bundled data on a server, the data will be put in a folder created in the first writeable persistence location.

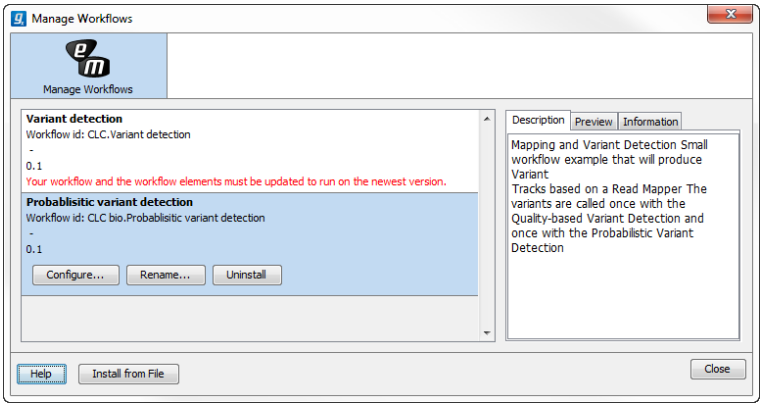



Figure 9.35: *Workflows available in the workflow manager. Note the alert on the "Variant detection" workflow, that means that this workflow needs to be updated.*

Click **Configure** and you will be presented with a dialog listing all the reference data that need to be selected. An example is shown in figure 9.36.

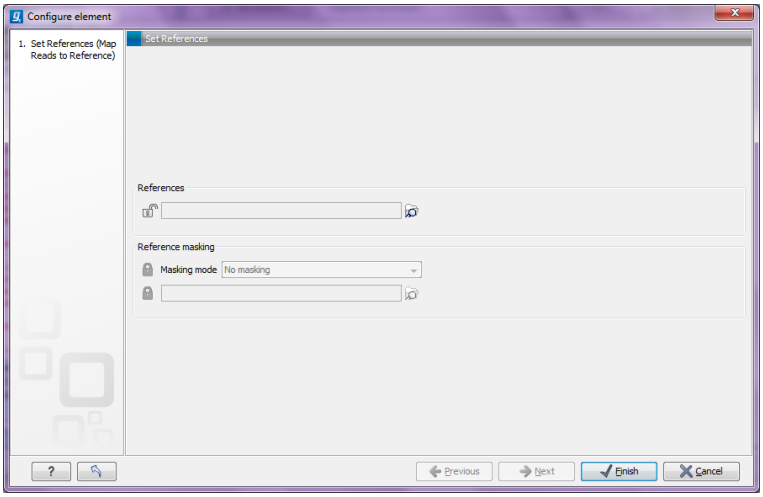



Figure 9.36: *Configuring parameters for the workflow.*

This dialog also allows you to further lock parameters of the workflow (see more about locking in

section 9.1.3).

If the workflow is intended to be executed on a server as well, it is important to select reference data that is located on the server.

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.

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 figure 9.31), the **Preview** shows a graphical representation of the workflow (figure 9.37), and finally you can get **Information** about the workflow (figure 9.38).

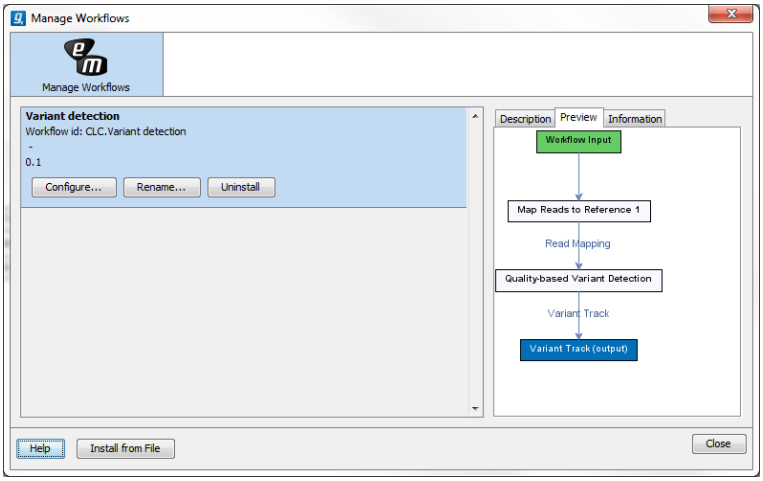



Figure 9.37: *Preview of the workflow.*

The "Information" field (figure 9.38) contains the following:

**Build id** The date followed by the time

**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 version (a technical number that can be used for troubleshooting)

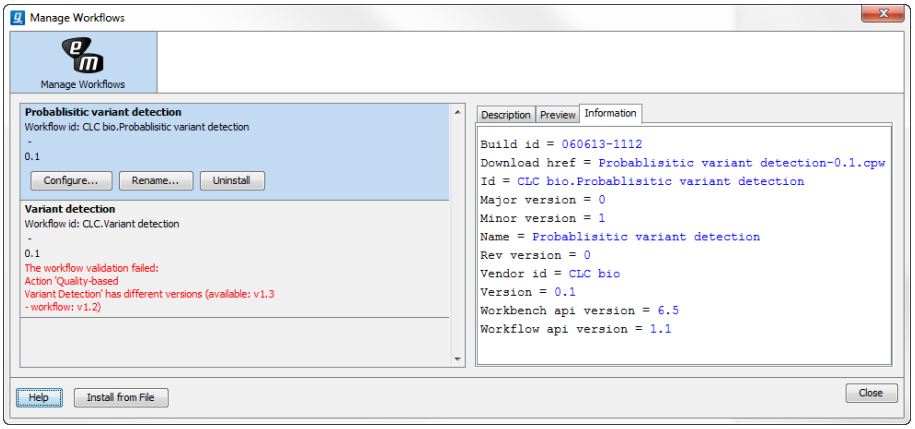



Figure 9.38: *With "Manage Workflows" it is possible to configure, rename and uninstall workflows.*

### 9.2.3 Workflow identification and versioning

A workflow has a version. The version is used to make it easy to distribute an improved version of the same workflow. To do this, create a new installer with an incremented version number. In order to install a new and updated version, the old one has to be uninstalled.

The way the *CLC Genomics Workbench* checks whether a workflow already exists in a previous version is by looking at the workflow id. The id is a combination of the organization name and the name of the workflow itself as it is shown in the dialog shown in figure 9.31. Once installed this information is also available in the workflow manager (in figure 9.35 this is `CLC bio.Simple variant detection and annotation-1.2`).

If you create two different workflows with the same name and using the same organization name when creating the installer, they cannot both be installed.

### 9.2.4 Automatic update of workflow elements

When new versions of the *CLC Genomics Workbench* are released, some of the tools that are part of a workflow may change. When this happens, the workflow may no longer be valid. This will happen both to the workflow configurations saved in the **Navigation Area** and the installed workflows.

When a workflow is opened from the **Navigation Area**, an editor will appear, if tools used in the workflow have been updated (see figure 9.39).

Updating a workflow means that the tools in your workflow is updated with the most recent version of these particular tools. To update your workflow, press the **OK** button at the bottom of the page.

There may be situations where it is important for you to keep the workflow in its original form. This could be the case if you have used a workflow to generate results for a publication. In such cases it may be necessary for you to be able to go back to the original workflow to e.g. repeat an analysis.

You have two options to keep the old workflow:

- If you do not wish to update the workflow at all, press the **Cancel** button. This will keep the workflow unchanged. However, the next time you open the workflow, you will again be

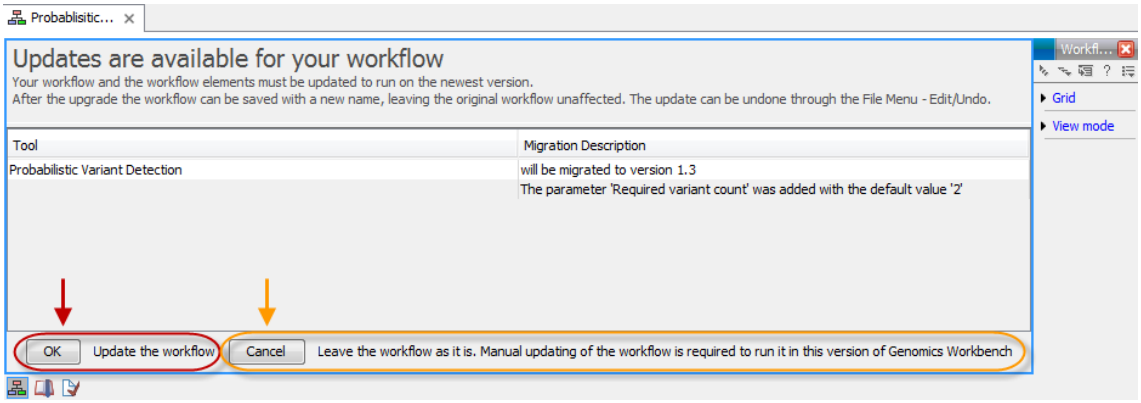



Figure 9.39: *When updates are available an editor appears with information about which tools should be updated. Press "OK" to update the workflow. The workflow must be updated to be able to run the workflow on the newest version of the Workbench.*

asked whether you wish to update the workflow. Please note that only updated workflows can run on the newest versions of the Workbench.

- Another option is to update the workflow and save the updated workflow with a new name. This will ensure that the old workflow is kept rather than being overwritten.

**Note!** In cases where new parameters have been added, these will be used with their default settings.

If you have used the toolbar "Workflow" button ( ) and "Manage Workflow..." ( ) to access a specific workflow in order to e.g. change the workflow configuration or are going to use the "Install from File" function, a button labeled "Update..." will appear whenever tools have been changed and the workflow needs to be updated (figure 9.40). When you click the button labeled "Update...", your workflow will be updated and the existing workflow will be overwritten.

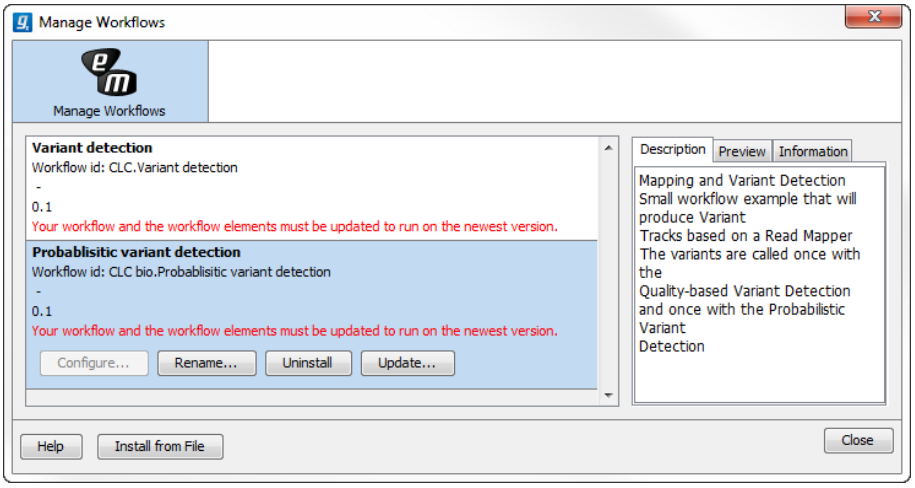



Figure 9.40: *Workflow migration.*

## 9.3 Executing a workflow

Once installed and configured, a workflow will appear in the **Toolbox** under **Workflows** ( ). If an icon was provided with the workflow installer this will also be shown (see figure 9.41).

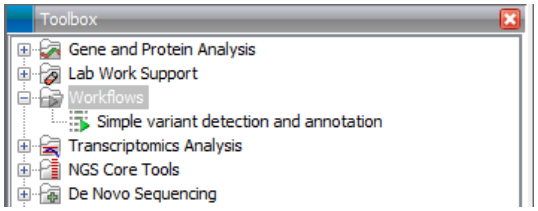



Figure 9.41: *A workflow is installed and ready to be used.*

The workflow is executed just as any other tool in the **Toolbox** by double-clicking or selecting it in the menu (or with the shortcut Ctrl + Enter). This will open a dialog where you provide input data and with options to run the workflow in batch mode (see section 8.1). In the last page of the dialog, you can preview all the parameters of the workflow, as well as the input data, before clicking "Next" to choose where to save the output, and then "Finish" to execute the workflow.

If you are connected to a *CLC Genomics Server*, you will be presented with the option to run the workflow locally on the Workbench or on the Server. When you are selecting where to run the workflow, you should also see a message should there be any missing configurations. There are more details about running Workflows on the Server in the Server manual (`http://www.clcsupport.com/clcgenomicsserver/current/admin/index.php?manual=Workflows.html`).

When the workflow is started, you can see the log file with detailed information from each step in the process.

If the workflow is not properly configured, you will see that in the dialog when the workflow is started [2].

## 9.4 Open copy of installed workflow

A copy of an installed and configured workflow found in the **Toolbox** under **Workflows** ( ) can be opened in the View Area by clicking once and then right-clicking on the name of the installed workflow in the toolbox (figure **??**).

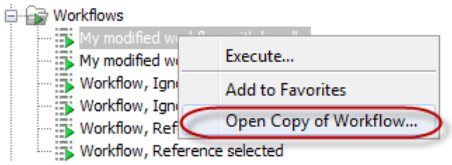



Figure 9.42: *A copy of an installed workflow can be opened from the Toolbox. The copied workflow will open in the View Area.*

An example of a copy of a workflow that has been opened in the **View Area** is shown in figure 9.43.

[2] 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 was created (see section 9.2.3)

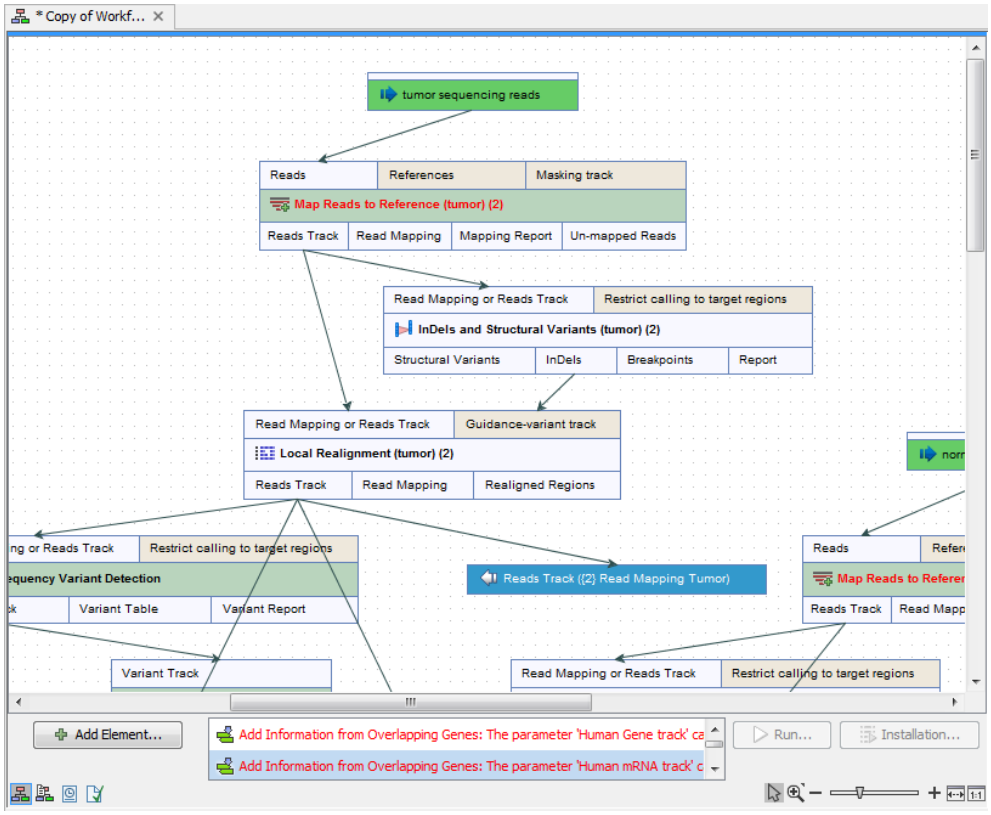



Figure 9.43: *A copy of an installed workflow after it has been opened in the View Area.*

# Part III

# Basic sequence analysis

# Chapter 10

# Viewing and editing sequences

**Contents**


**10.1 View sequence . . . 177**
10.1.1 Sequence settings in Side Panel . . . 178
10.1.2 Restriction sites in the Side Panel . . . 184
10.1.3 Selecting parts of the sequence . . . 185
10.1.4 Editing the sequence . . . 186
10.1.5 Sequence region types . . . 186
**10.2 Circular DNA . . . 186**
10.2.1 Using split views to see details of the circular molecule . . . 188
10.2.2 Mark molecule as circular and specify starting point . . . 188
**10.3 Working with annotations . . . 189**
10.3.1 Viewing annotations . . . 189
10.3.2 Adding annotations . . . 193
10.3.3 Edit annotations . . . 195
10.3.4 Removing annotations . . . 196
**10.4 Element information . . . 197**
**10.5 View as text . . . 198**
**10.6 Sequence Lists . . . 198**
10.6.1 Graphical view of sequence lists . . . 199
10.6.2 Sequence list table . . . 200
10.6.3 Extract sequences from sequence list . . . 201


*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.

## 10.1 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 20.2).

### 10.1.1 Sequence settings in Side Panel

Each view of a sequence has a **Side Panel** located at the right side of the view (see figure 10.1.

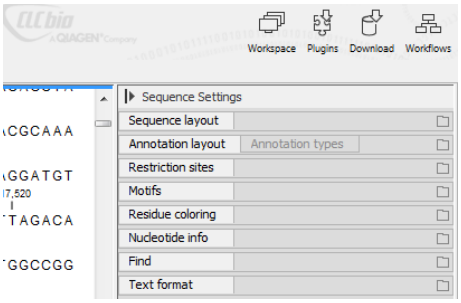



Figure 10.1: *Overview of the Side Panel which is always shown to the right of a view.*

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 10.3.1.

#### Restriction sites

See section 10.1.2.

#### Motifs

See section 14.9.1.

### Residue coloring

These preferences make it possible to color both the residue letter and set a background color for the residue.

- **Non-standard residues.** For nucleotide sequences this will color the residues that are not C, G, A, T or U. For amino acids only B, Z, and X are colored as non-standard residues.
  - **Foreground color.** Sets the color of the letter. Click the color box to change the color.
  - **Background color.** Sets the background color of the residues. Click the color box to change the color.
- **Rasmol colors.** Colors the residues according to the Rasmol color scheme.
  See http://www.openrasmol.org/doc/rasmol.html
  - **Foreground color.** Sets the color of the letter. Click the color box to change the color.
  - **Background color.** Sets the background color of the residues. Click the color box to change the color.
- **Polarity colors (only protein).** Colors the residues according to the following categories:
  - **Green** neutral, polar
  - **Black** neutral, nonpolar
  - **Red** acidic, polar
  - **Blue** basic ,polar
  - As with other options, you can choose to set or change the coloring for either the residue letter or its background:
    - **Foreground color.** Sets the color of the letter. Click the color box to change the color.
    - **Background color.** Sets the background color of the residues. Click the color box to change the color.
- **Trace colors (only DNA).** Colors the residues according to the color conventions of chromatogram traces: A=green, C=blue, G=black, and T=red.
  - **Foreground color.** Sets the color of the letter.
  - **Background color.** Sets the background color of the residues.

### Nucleotide info

These preferences only apply to nucleotide sequences.

- **Color space encoding.** Lets you define a few settings for how the colors should appear.

  **Infer encoding** This is used if you want to display the colors for non-color space sequence (e.g. a reference sequence). The colors are then simply inferred from the sequence.

  **Show corrections** This is only relevant for mapping results - it will show where the mapping process has detected color errors. An example of a color error is shown in figure 25.21.

**Hide unaligned** This option determines whether color for the unaligned ends of reads should be displayed. It also controls whether colors should be shown for gaps. The idea behind this is that these color dots will interfere with the color alignment, so it is possible to turn them off.

- **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 10.1.3.
    - **+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 15.5.
  - **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 18.1.
- **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 6.6).
    - **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 6.6).
    - **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 22.2.3 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 16.5.2.

- **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. Below this option you can choose to search for translations as well. Sequences annotated with coding regions often have the translation specified which can lead to undesired results.

- **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 (see section 10.3.2). 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, mapping results and BLAST results.

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.

### 10.1.2 Restriction sites in the Side Panel

Please see section 19.3.1.

### 10.1.3 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 10.1.2.)

#### 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.

### 10.1.4 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 19.1.4.

**Note** When editing annotated nucleotide sequences, the annotation content is not updated automatically (but its position is). Please refer to section 10.3.3 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.

### 10.1.5 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 10.2 is an example of three regions with separate colors.

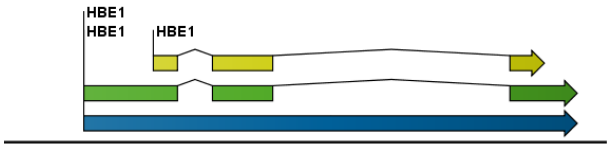



Figure 10.2: *Three regions on a human beta globin DNA sequence (HUMHBB).*

Figure 10.3 shows an artificial sequence with all the different kinds of regions.

## 10.2 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" ( )**

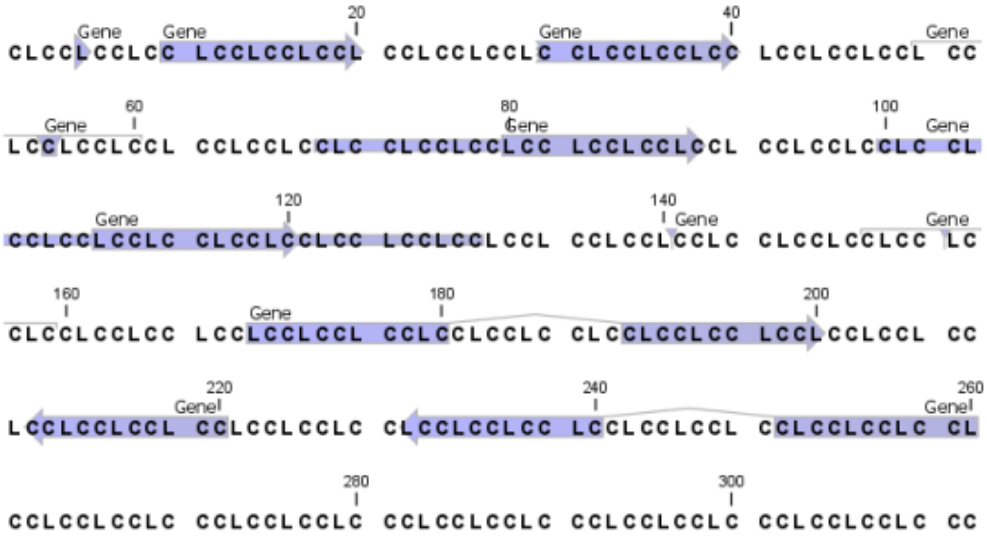



Figure 10.3: *Region #1: A single residue, Region #2: A range of residues including both endpoints, Region #3: A range of residues starting somewhere before 30 and continuing up to and including 40, Region #4: A single residue somewhere between 50 and 60 inclusive, Region #5: A range of residues beginning 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.*

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 10.4.

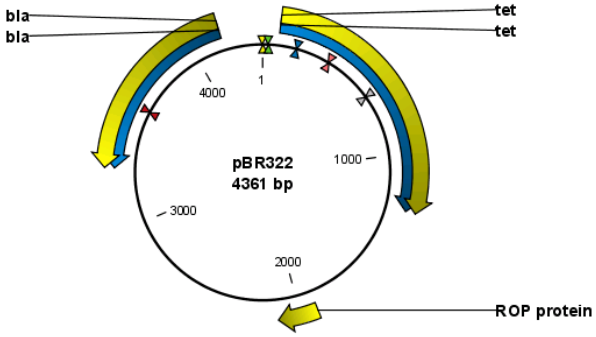



Figure 10.4: *A molecule shown in a circular view.*

This view of the sequence shares some of the properties of the linear view of sequences as described in section 10.1, 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.

### 10.2.1 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 10.5.

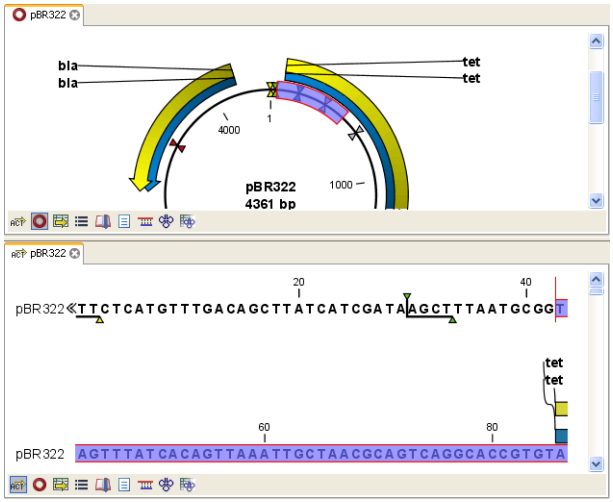



Figure 10.5: *Two views showing the same sequence. The bottom view is zoomed in.*

**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.

### 10.2.2 Mark molecule as circular and specify starting point

You can mark a DNA molecule as circular by right-clicking its name in either the sequence view or the circular view. In the right-click menu you can also make a circular molecule linear. A circular molecule displayed in the normal sequence view, will have the sequence ends marked with a ï¿½.

The starting point of a circular sequence can be changed by:

**make a selection starting at the position that you want to be the new starting point | right-click the selection | Move Starting Point to Selection Start**

**Note!** This can only be done for sequence that have been marked as circular.

## 10.3 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 13.5.2), and atom groups defined on the structure transferred to sequence annotations or vica versa (section 13.5.3).
- You can manually add annotations to a sequence (described in the section 10.3.2).

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 14.1.

**Note!** Annotations are included if you export the sequence in GenBank, Swiss-Prot, EMBL or CLC format. When exporting in other formats, annotations are not preserved in the exported file.

### 10.3.1 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 10.6 shows an annotation displayed on a sequence.

The various sequence views listed in section 10.3.1 have different default settings for showing annotations. However, they all have two groups in the **Side Panel** in common:

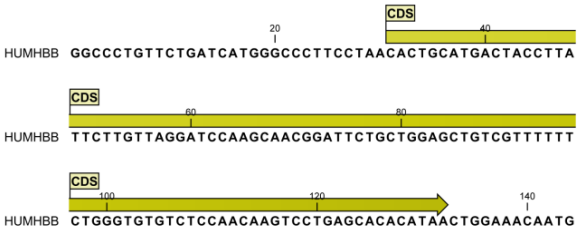



Figure 10.6: *An annotation showing a coding region on a genomic dna sequence.*

- **Annotation Layout**
- **Annotation Types**

The two groups are shown in figure 10.7.

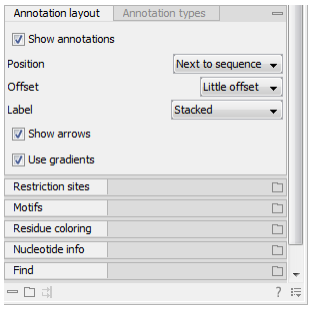



Figure 10.7: *The annotation layout in the Side Panel. The annotation types can be shown by clicking on the "Annotation types" tab.*

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).
- **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 10.8).

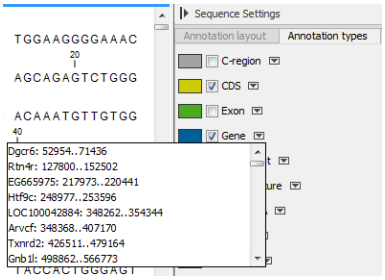



Figure 10.8: *Browsing the gene annotations on a sequence.*

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.

### 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 10.9).

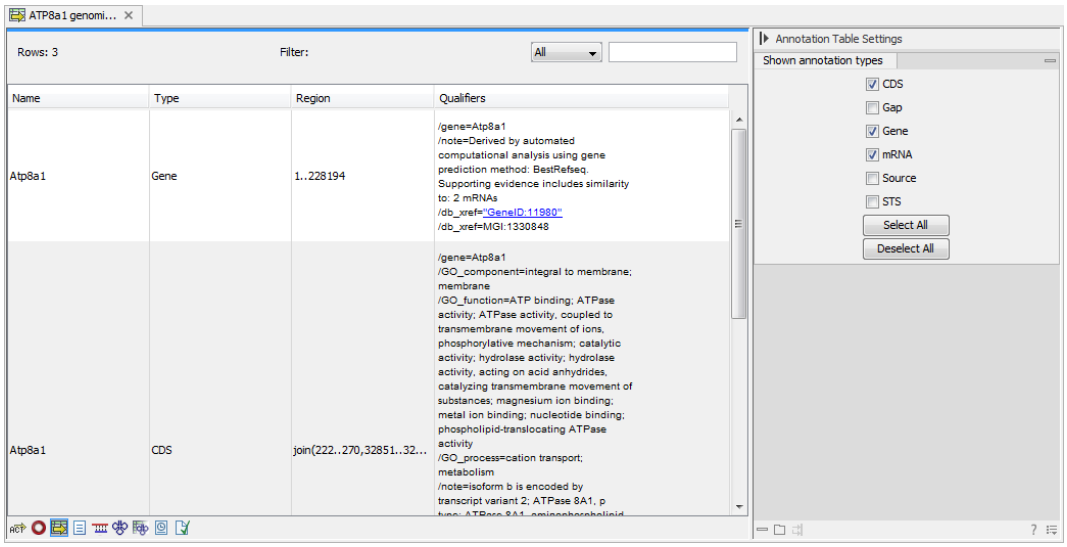



Figure 10.9: *A table showing annotations on the sequence.*

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 10.3.2).

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 10.3.2).

### 10.3.2 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 10.10.

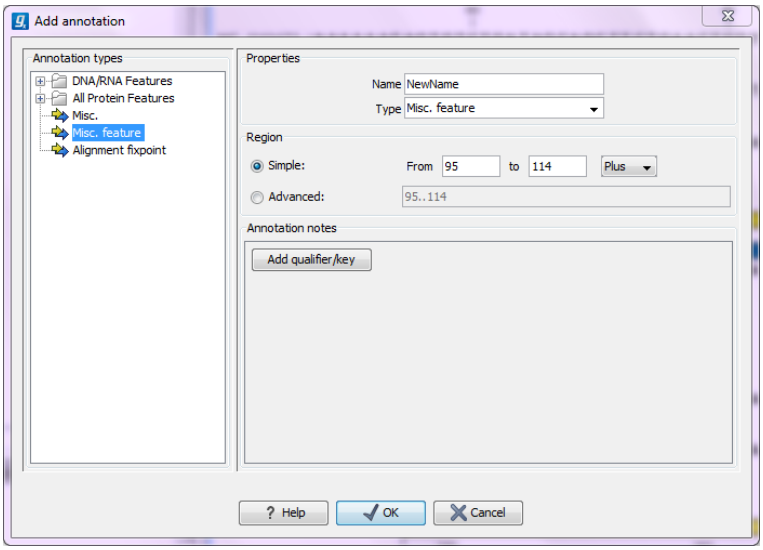



Figure 10.10: *The Add Annotation dialog.*

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.

[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

(Whether the name is actually shown depends on the **Annotation Layout** preferences, see section 10.3.1).

- **Type.** Reflects the left-hand part of the dialog as described above. You can also choose directly in this list or type your own annotation type.
- **Region.** If you have already made a selection, this field will show the positions of the selection. You can modify the region further using the conventions of DDBJ, EMBL and 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 10.3.1) 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. "www.clcbio.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.

### 10.3.3 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 10.10, 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 10.11.

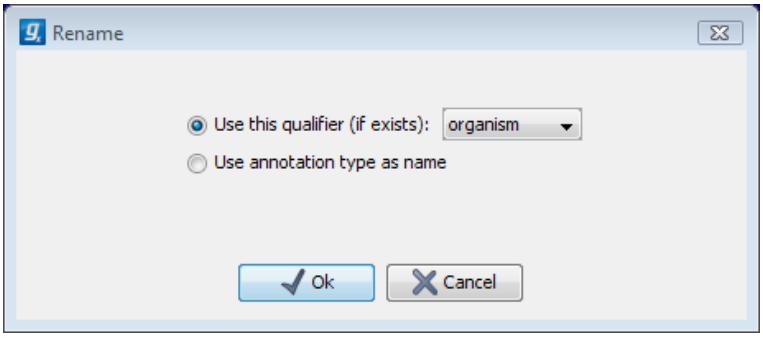



Figure 10.11: *The Advanced Rename dialog.*

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 10.12.

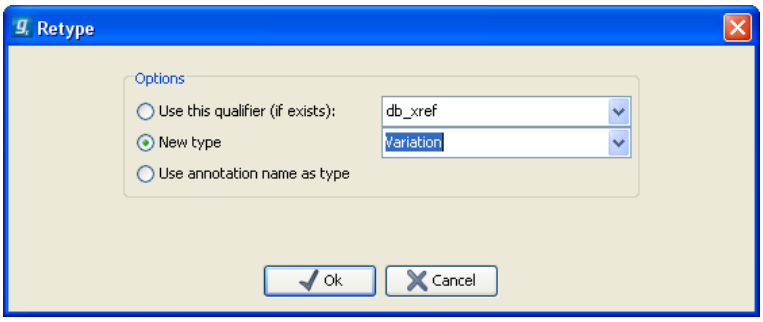



Figure 10.12: *The Advanced Retype dialog.*

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).

### 10.3.4 Removing annotations

Annotations can be hidden using the **Annotation Types** preferences in the **Side Panel** to the right of the view (see section 10.3.1). 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**

## 10.4 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 10.13.




Figure 10.13: *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.

- **Name.** The name of the sequence which is also shown in sequence views and in the **Navigation Area**.
- **Description.** A description of 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 the history (section 7) 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.

The information available depends on the origin of the sequence. Sequences downloaded from database like NCBI and UniProt (see section 11) have this information. On the other hand, some sequence formats like fasta format do not contain this information.

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.

Note that for other kinds of data, the **Element info** will only have **Name** and **Description**.

## 10.5 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 10.4.)

In the **Side Panel**, you find a search field for searching the text in the view.

## 10.6 Sequence Lists

The **Sequence List** shows a number of sequences in a tabular format or it can show the sequences together in a normal sequence view.

Having sequences in a sequence list can help organizing sequence data.

Sequence lists are generated automatically when you import files containing more than one sequence. Sequence lists may also be created as the output from particular Workbench tool including database searches.

See (chapter 11.1).

**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** | **right-click the elements** | **New** | **Sequence List ( )**

Alternatively, you can launch this took via the menu system:

**File** | **New** | **Sequence List ( )**

This opens the **Sequence List** Wizard:

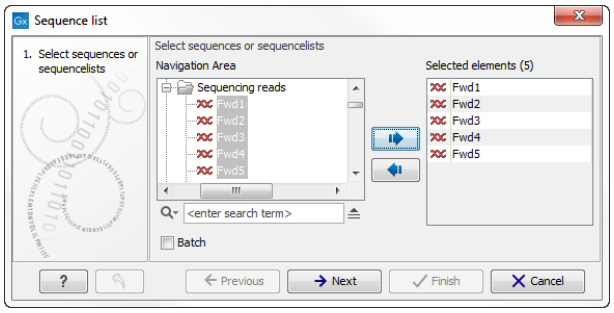



Figure 10.14: *A Sequence List dialog.*

The dialog allows you to select more sequences to include in the list, or to remove already chosen sequences from the list.

Clicking **Finish** opens the sequence list. It can be saved by clicking **Save** ( ) or by dragging the tab of the view into the **Navigation Area**.

Opening a Sequence list is done by:

**right-click the sequence list in the Navigation Area** | **Show ( )** | **Graphical Sequence List ( ) OR Table ( )**

The two different views of the same sequence list are shown in split screen in figure 10.15.

### 10.6.1 Graphical view of sequence lists

The graphical view of sequence lists is almost identical to the view of single sequences (see section 10.1). The main difference is that you now can see more than one sequence in the same view.

However, you also have a few extra options for sorting, deleting and adding sequences:

- 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**.

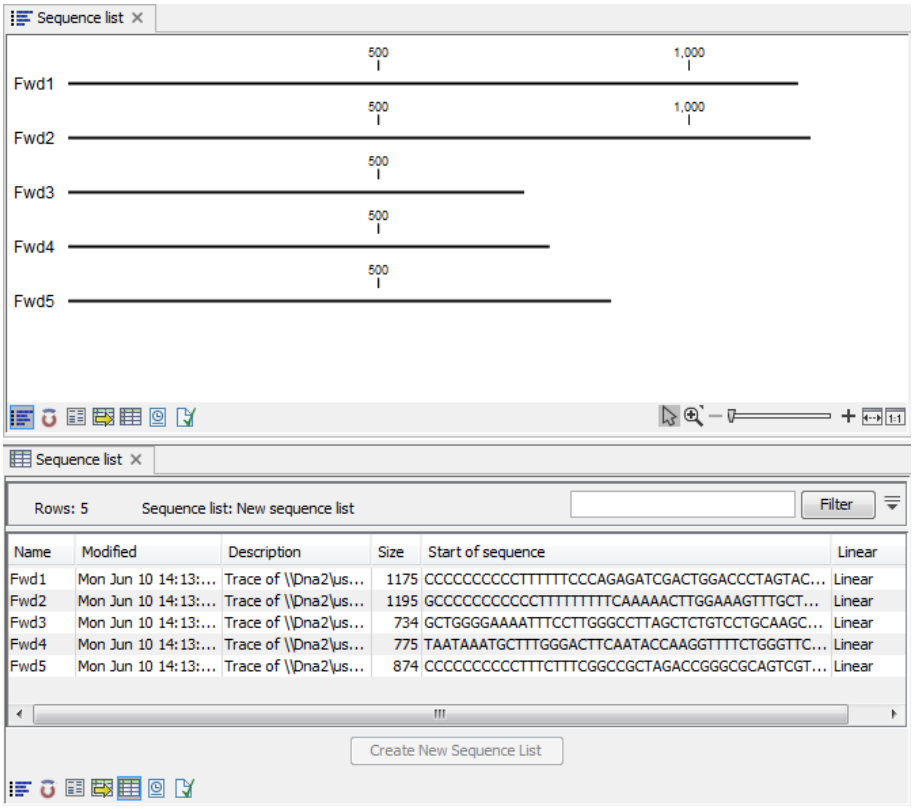



Figure 10.15: *A sequence list containing multiple sequences can be viewed in either a table or in a graphical sequence list. The graphical view is useful for viewing annotations and the sequence itself, while the table view provides other information like sequence lengths, and the number of sequences in the list (number of Rows reported).*

### 10.6.2 Sequence list table

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** ( ).

You can also create a subset of the sequence list:

**select the relevant sequences** | **right-click** | **Create New Sequence List**

This will create a new sequence list, which only includes the selected sequences.

Learn more about tables in Appendix 8.3.

### 10.6.3 Extract sequences from sequence list

Sequences can be extracted from a sequence list when the sequence list is opened in tabular view. One or more sequences can be dragged (with the mouse) directly from the table into the **Navigation Area**. This allows you to extract specific sequences from the entire list. Another option is to extract all sequences found in the list. This can be done with the **Extract Sequences** tool:

**Toolbox** | **Classical Sequence Analysis** (  ) | **General Sequence Analysis** (  )|. **Extract Sequences** ()

A description of how to use the **Extract Sequences** tool can be found in section 14.2.

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

# Chapter 11

# Data download

**Contents**


**11.1 GenBank search . . . . . 202**
11.1.1 GenBank search options . . . . . 203
11.1.2 Handling of GenBank search results . . . . . 204
11.1.3 Save GenBank search parameters . . . . . 205
**11.2 UniProt (Swiss-Prot/TrEMBL) search . . . . . 206**
11.2.1 UniProt search options . . . . . 206
11.2.2 Handling of UniProt search results . . . . . 207
11.2.3 Save UniProt search parameters . . . . . 208
**11.3 Search for structures at NCBI . . . . . 208**
11.3.1 Structure search options . . . . . 209
11.3.2 Handling of NCBI structure search results . . . . . 210
11.3.3 Save structure search parameters . . . . . 211
**11.4 Download reference genome data . . . . . 212**
11.4.1 Selecting data types for download . . . . . 212
11.4.2 Cytogenetic ideograms . . . . . 214
**11.5 Sequence web info . . . . . 215**
11.5.1 Google sequence . . . . . 215
11.5.2 NCBI . . . . . 215
11.5.3 PubMed References . . . . . 216
11.5.4 UniProt . . . . . 216
11.5.5 Additional annotation information . . . . . 216


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

## 11.1 GenBank search

This section describes searches for sequences in GenBank - the **NCBI Entrez** database. The NCBI search view is opened in this way (figure 11.1):

**Download | Search for Sequences at NCBI ( )**

or **Ctrl + B (⌘ + B on Mac)**

This opens the following view:

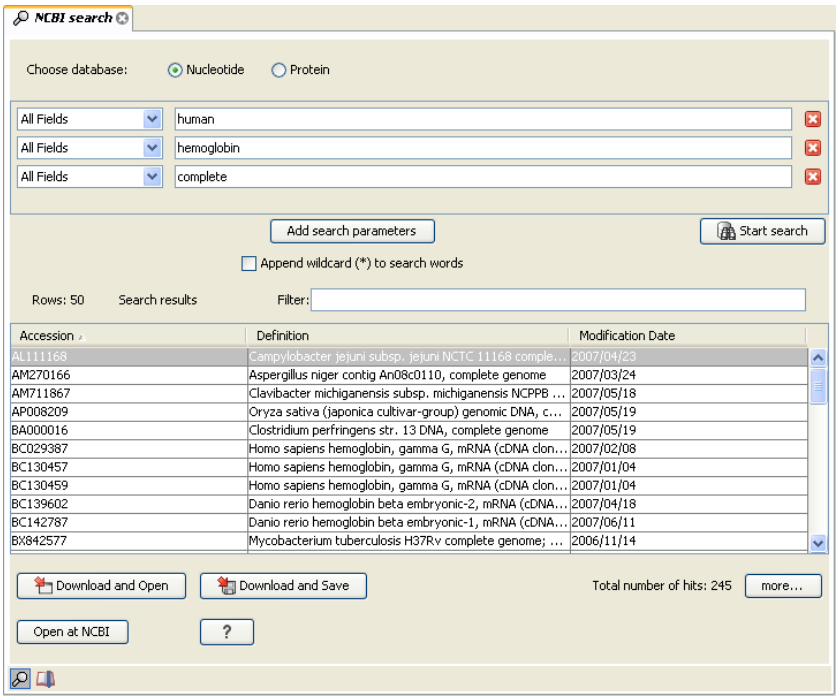



Figure 11.1: *The GenBank search view.*

### 11.1.1 GenBank search options

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

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

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 NCBI database at the same time.
- **Organism**. Text.
- **Description**. Text.
- **Modified Since**. Between 30 days and 10 years.

- **Gene Location**. Genomic DNA/RNA, Mitochondrion, or Chloroplast.
- **Molecule**. Genomic DNA/RNA, mRNA or rRNA.
- **Sequence Length**. Number for maximum or minimum length of the sequence.
- **Gene Name**. Text.

The search parameters are the most recently used. The **All fields** allows searches in all parameters in the NCBI 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. You can also write e.g. `CD9 NOT homo sapiens` in **All fields**.

**Note!** The 'Feature Key' option is only available in GenBank when searching for nucleotide sequences. 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**.

**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.

### 11.1.2 Handling of GenBank 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. This 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 sequence hit is represented by text in three columns:

- Accession.
- Description.
- Modification date.
- 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, doesn't save the sequence.
- Download and save, lets you choose location for saving sequence.
- Open at NCBI, searches the sequence at NCBI'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 GenBank 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 GenBank search results using right-click menu

You may also select one or more sequences from the list and download using the right-click menu (see figure 11.2). Choosing **Download and Save** lets you select a folder where the sequences are saved when they are downloaded. Choosing **Download and Open** opens a new view for each of the selected sequences.

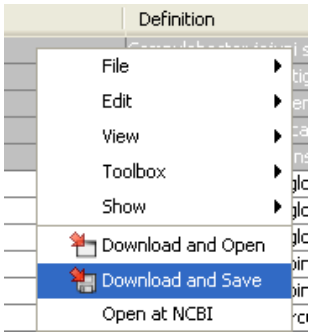



Figure 11.2: *By right-clicking a search result, it is possible to choose how to handle the relevant sequence.*

#### Copy/paste from GenBank search results

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

To copy/paste files into the **Navigation Area**:

> **select one or more of the search results | Ctrl + C (⌘ + C on Mac) | select a 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.

### 11.1.3 Save GenBank 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.

## 11.2 UniProt (Swiss-Prot/TrEMBL) search

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 11.3) is opened in this way:

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

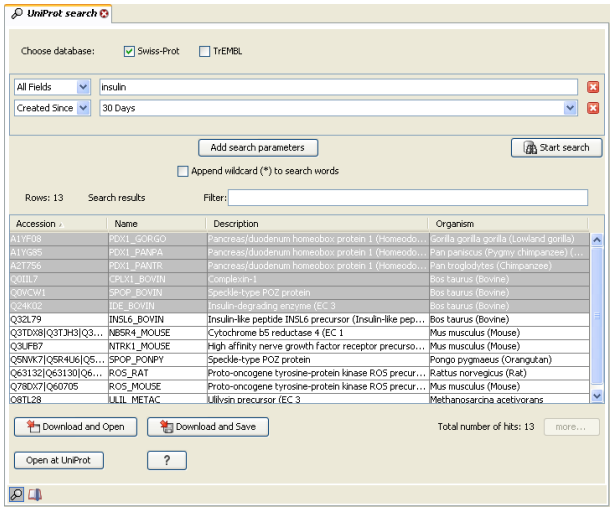



Figure 11.3: *The UniProt search view.*

### 11.2.1 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 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.

### 11.2.2 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 (see figure 11.2). 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.

### 11.2.3 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.

## 11.3 Search for 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 13.2.

The NCBI search view is opened in this way:

> **Download** | **Search for structures at NCBI** ()

or **Ctrl + B (⌘ + B on Mac)**

This opens the view shown in figure 11.4:

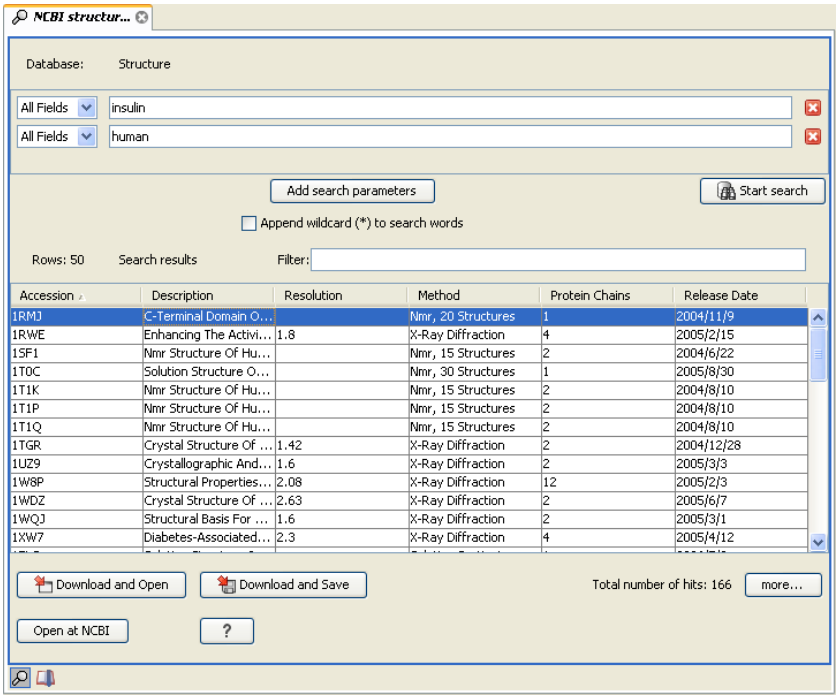



Figure 11.4: *The structure search view.*

### 11.3.1 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.

### 11.3.2 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 11.5). 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.

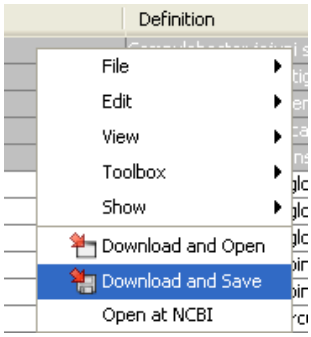



Figure 11.5: *By right-clicking a search result, it is possible to choose how to handle the relevant structure.*

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.

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.

### 11.3.3 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.

## 11.4 Download reference genome data

The *CLC Genomics Workbench* offers an easy way of retrieving popular reference data sources such as genes, variant annotations and genome sequences as tracks. The data itself is not provided or hosted by CLC bio. CLC bio only provides an easy way to retrieve data that should otherwise have been downloaded and imported:

**Download | Download Reference Genome Data ( )**

This opens the dialog shown in figure 11.6:

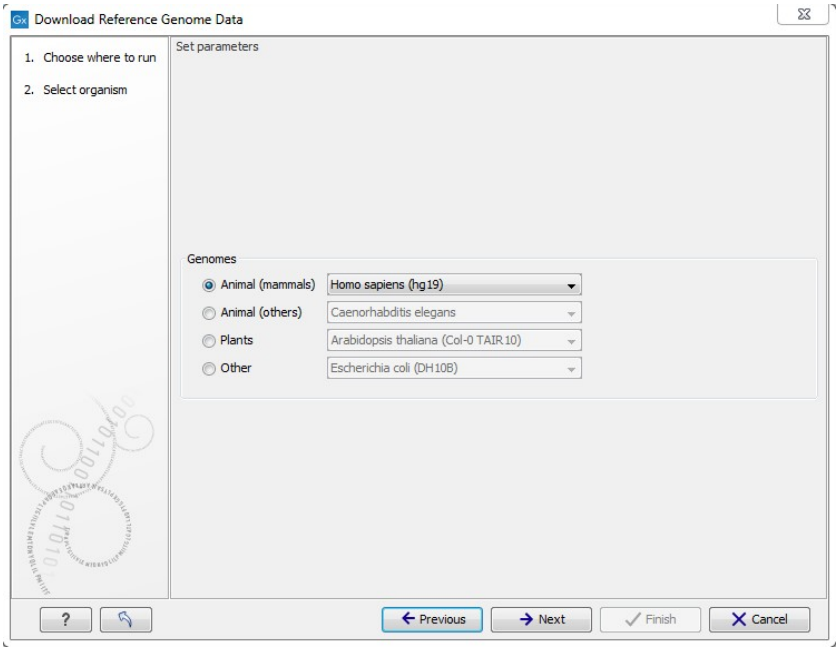



Figure 11.6: *Selecting an organism for download.*

Select the organism of interest and click **Next**. The list of organisms is dynamically updated by CLC bio independent of Workbench versions, so you will always see the most recent list of organisms.

The list just shows a selection of some of the popular organisms – if you do not find what you are looking for, there is always the possibility to download and import the data as well.

### 11.4.1 Selecting data types for download

Once you have clicked **Next**, you will be asked whether you wish to download the genome sequence or whether you already have it available as shown in figure 11.7:

If you do not already have an existing sequence imported into the *CLC Genomics Workbench*, it will be downloaded automatically from Ensembl. If you already have a reference sequence, it

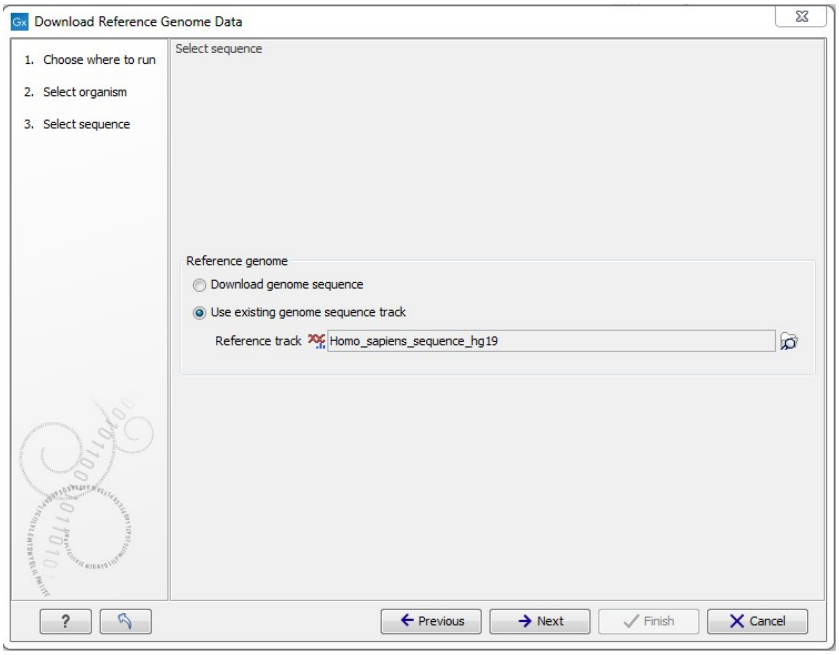



Figure 11.7: *Selecting a reference genome sequence if available.*

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.

Clicking **Next** allows you to select which types of annotation data you wish to download as shown in figure 11.8: .

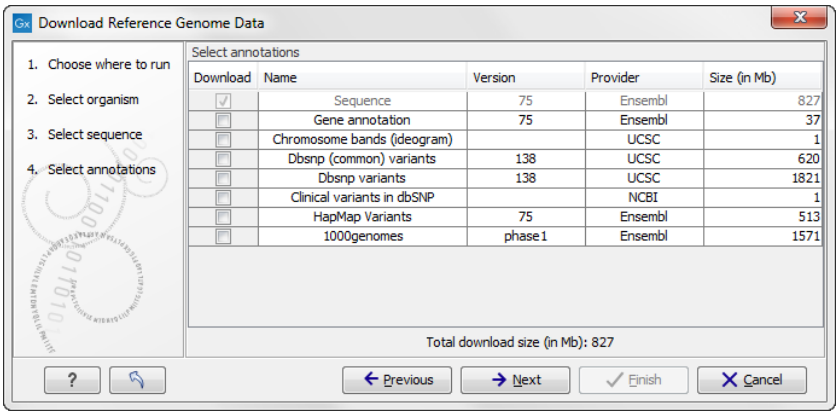



Figure 11.8: *Selecting different types of annotation data for human hg19.*

This step is different from organism to organism and depends on the data sources that CLC bio has included for download. In the example in figure 11.8 showing hg19 build of the human genome, a lot of variant data is available from e.g. dbSNP or COSMIC. Please note that for human data, a difference between the UCSC genome build and Ensembl/NCBI exists, which means that variants downloaded from UCSC will not be annotated on the mitochondrial genome when using this download tool (see `http://genome.ucsc.edu/cgi-bin/hgGateway?db=hg19`).

Both the data provider and the size of the data is listed in the dialog, and at the bottom you can see the size of all the selected downloads. 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, notably larger.

All data downloaded with this tool will be tracks (either sequence tracks or various kinds of annotation tracks). The Ensemble 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.

### 11.4.2 Cytogenetic ideograms

An ideogram, also called a cytogenetic ideogram, is a chromosome map with numbered banding patterns that shows the relationship between the two chromosome arms and the centromere. Users of the UCSC Genome Browser (`http://genome.ucsc.edu/`) are most likely familiar with the use of cytogenetic ideogram in the UCSC Genome Browser. Ideograms can be imported into the Workbench (from UCSC) and be used in track lists to provide an overview of the analyzed data in context of the individual chromosomes. An example is shown in figure 11.9.

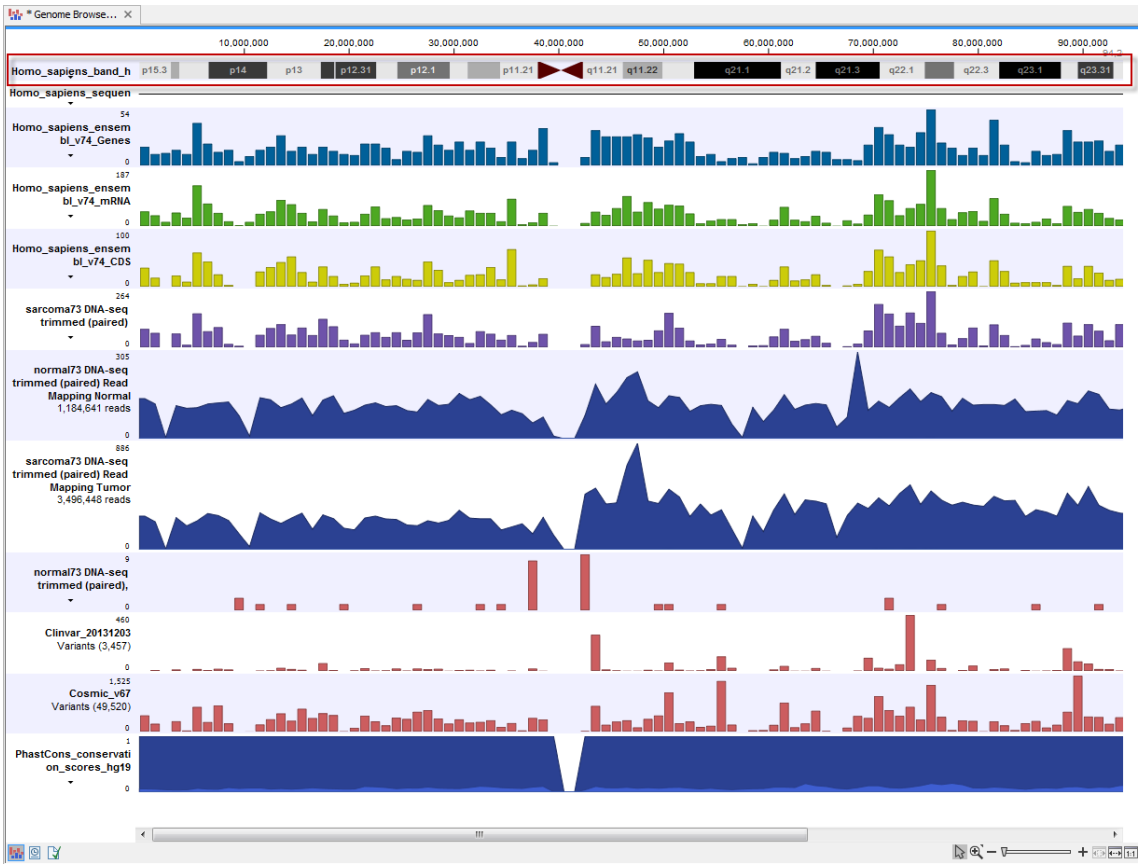



Figure 11.9: *The ideogram is particularly useful when used in combination with other tracks in a track list. In this figure the ideogram is highlighted with a red box.*

The ideogram can be imported into *CLC Genomics Workbench* via the download function found in the upper right corner of the Workbench.

**Download | Download Reference Genome Data ( ) | select "Chromosome bands (ideogram)"**

For a description of how to do this please refer to section 11.4. The imported ideograms can be

found in the **Navigation Area** at the destination that was specified in the download wizard. The ideograms are always accompanied by a reference sequence (see figure 11.10).

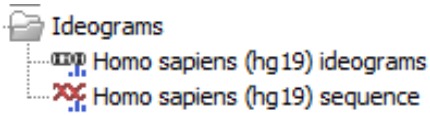



Figure 11.10: *The ideograms and the reference sequence in the Navigation Area.*

## 11.5 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 10.5). 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 11.11):

**Open a sequence or a sequence list | Right-click the name of the sequence | Web Info ( ) | select the desired search function**

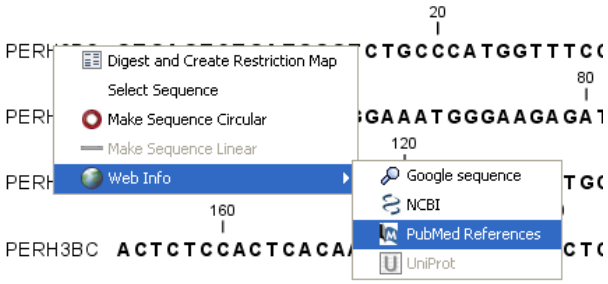



Figure 11.11: *Open webpages with information about this sequence.*

This will open your computer's default browser searching for the sequence that you selected.

### 11.5.1 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.

### 11.5.2 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.

### 11.5.3 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.

### 11.5.4 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.

### 11.5.5 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 12

# BLAST search

## Contents


**12.1 Running BLAST searches . . . . . 218**
12.1.1 BLAST at NCBI . . . . . 219
12.1.2 BLAST a partial sequence against NCBI . . . . . 222
12.1.3 BLAST against local data . . . . . 222
12.1.4 BLAST a partial sequence against a local database . . . . . 225
**12.2 Output from BLAST searches . . . . . 226**
12.2.1 Graphical overview for each query sequence . . . . . 226
12.2.2 Overview BLAST table . . . . . 226
12.2.3 BLAST graphics . . . . . 228
12.2.4 BLAST table . . . . . 229
12.2.5 Extracting a consensus sequence from a BLAST result . . . . . 232
**12.3 Local BLAST databases . . . . . 232**
12.3.1 Make pre-formatted BLAST databases available . . . . . 232
12.3.2 Download NCBI pre-formatted BLAST databases . . . . . 232
12.3.3 Create local BLAST databases . . . . . 233
**12.4 Manage BLAST databases . . . . . 235**
12.4.1 Migrating from a previous version of the Workbench . . . . . 236
**12.5 Bioinformatics explained: BLAST . . . . . 236**
12.5.1 Examples of BLAST usage . . . . . 236
12.5.2 Searching for homology . . . . . 237
12.5.3 How does BLAST work? . . . . . 237
12.5.4 Which BLAST program should I use? . . . . . 238
12.5.5 Which BLAST options should I change? . . . . . 239
12.5.6 Explanation of the BLAST output . . . . . 241
12.5.7 I want to BLAST against my own sequence database, is this possible? . 241
12.5.8 What you cannot get out of BLAST . . . . . 244
12.5.9 Other useful resources . . . . . 244

*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 12.5.

Figure 12.9 shows an example of a BLAST result in the *CLC Genomics Workbench*.

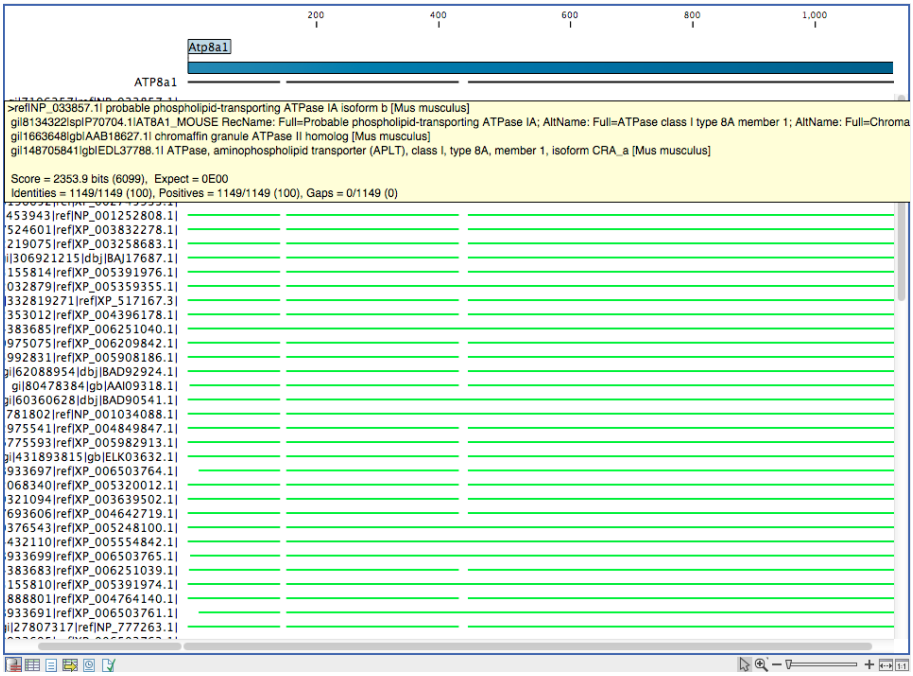



Figure 12.1: *Display of the output of a BLAST search. At the top is there a graphical representation of BLAST hits with tool-tips showing additional information on individual hits. Below is a tabular form of the BLAST results.*

## 12.1 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.

### 12.1.1 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 12.2

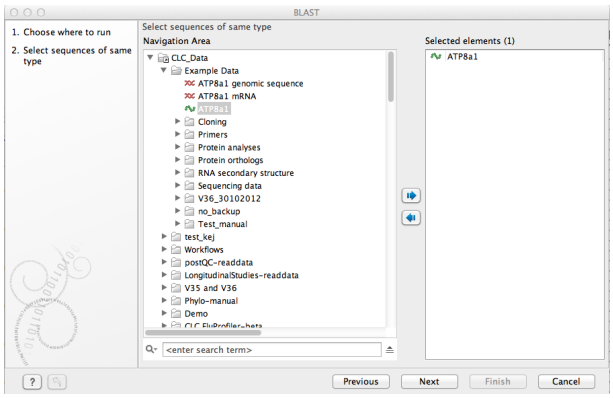



Figure 12.2: *Choose one or more sequences to conduct a BLAST search with.*

Select one or more sequences of the same type (either DNA or protein) and click **Next**.

In this dialog, you choose which type of BLAST search to conduct, and which database to search against. See figure 12.3. 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 D. Here you can also read how to add additional databases available the NCBI to the list provided in the dropdown menu.

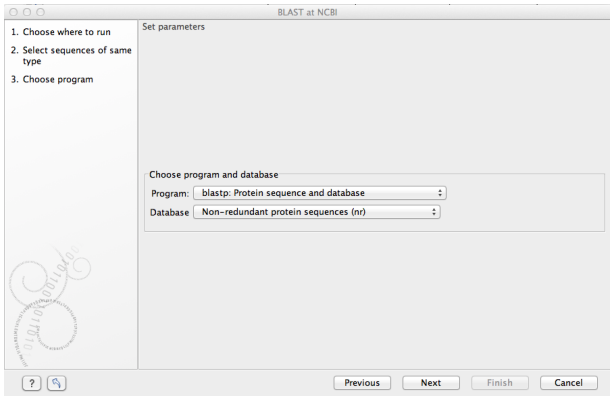



Figure 12.3: *Choose a BLAST Program and a database for the search.*

**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 12.4, allows you to choose parameters to tune your BLAST search, to meet your requirements.

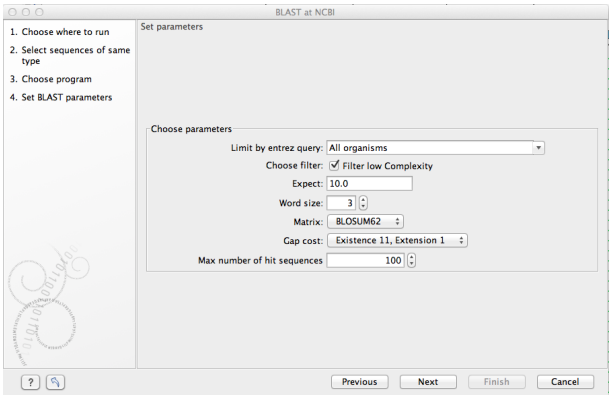



Figure 12.4: *Parameters that can be set before submitting a BLAST search.*

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.

- **Choose filter.** You can choose to apply **Low-complexity**. 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 default value is 10, meaning that under the circumstances of this search, 10 matches are expected to be found merely by chance according to the stochastic model of Karlin and Altschul (1990). Details of how E-values are calculated can be found at the NCBI: http://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-1.html If the E-value ascribed to a match is greater than the EXPECT threshold, the match will not be reported. Lower EXPECT 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 of E less than one 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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

### 12.1.2 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 12.3 and the rest of the options are the same as when performing a BLAST search with a full sequence.

### 12.1.3 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, the *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 12.3.3)
- You can add pre-formatted databases (see section 12.3.1)
- 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 12.5:

Select one or more sequences of the same type (DNA or protein) and click **Next**.

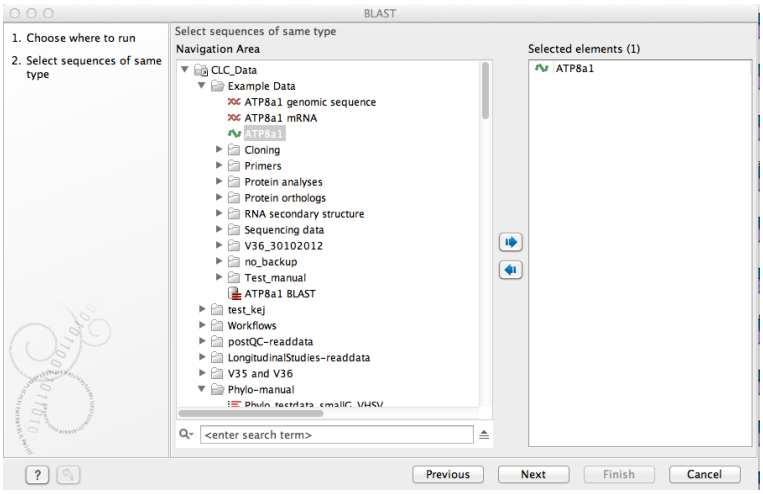



Figure 12.5: *Choose one or more sequences to conduct a BLAST search.*

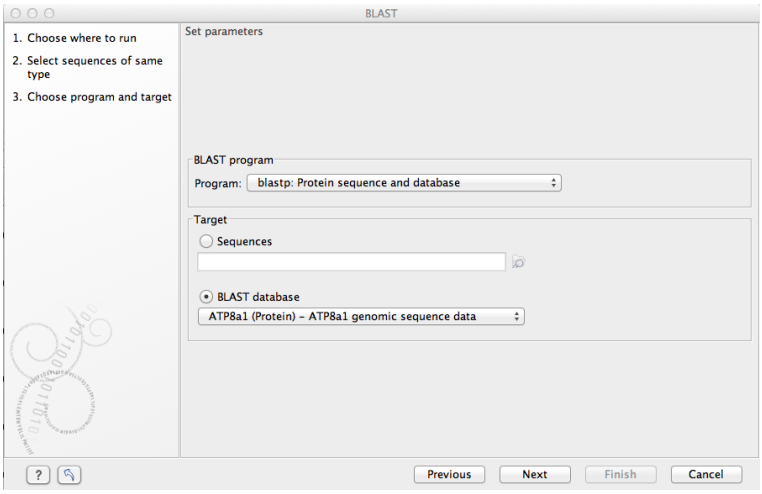



Figure 12.6: *Choose a BLAST program and a target database.*

This opens the dialog seen in figure 12.6:

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.
- **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 13.1.3).

Click **Next**.

This dialog allows you to adjust the parameters to meet the requirements of your BLAST search (figure 12.7).

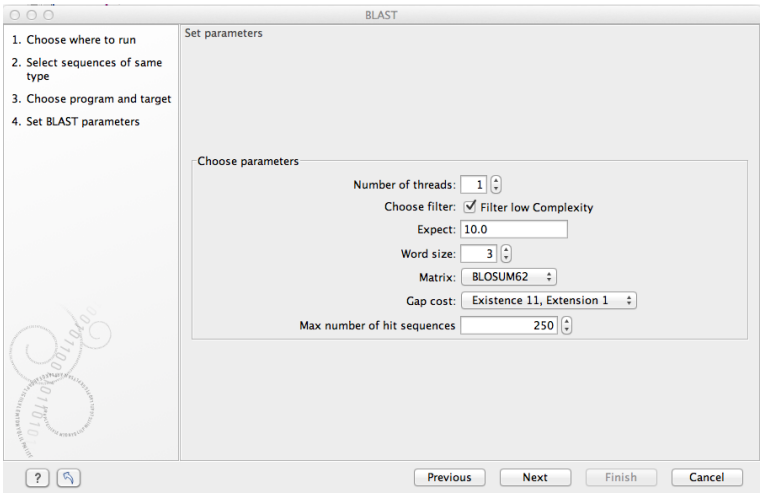



Figure 12.7: *Parameters that can be set before submitting a local BLAST search.*

- **Number of threads.** You can specify the number of threads, which should be used if your Workbench is installed on a multi-threaded system.
- **Choose filter.** You can choose to apply **Low-complexity**. 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 default value is 10, meaning that under the circumstances of this search, 10 matches are expected to be found merely by chance according to the stochastic model of Karlin and Altschul (1990). Details of how E-values are calculated can be found at the NCBI: `http://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-1.html` If the E-value ascribed to a match is greater than the EXPECT threshold, the match will not be reported. Lower EXPECT

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 of E less than one 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.

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 12.3.3.

- **BLAST Database**. Select a database already available in one of your designated BLAST database folders. Read more in section 12.4.

When a database or a set of sequences has been selected, click **Next**.

This opens the dialog seen in figure 12.8:

See section 12.1.1 for information about these limitations.

### 12.1.4 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:

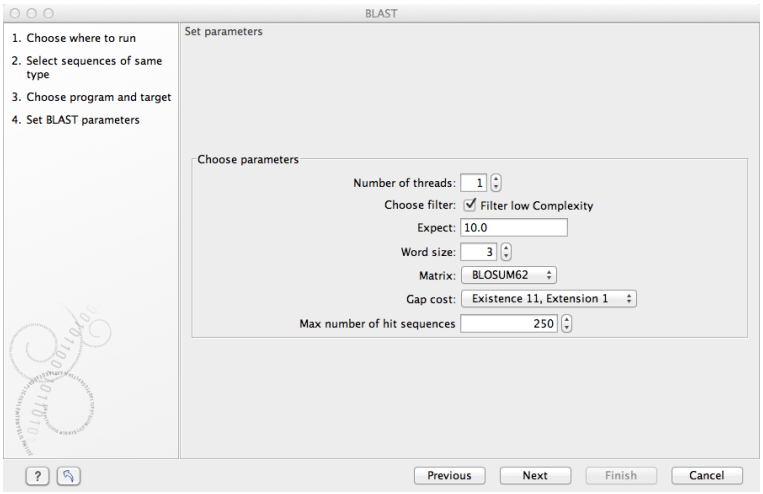



Figure 12.8: *Examples of parameters that can be set before submitting a BLAST search.*

**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 12.6 and the rest of the options are the same as when performing a BLAST search with a full sequence.

## 12.2 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 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 12.2.2.

### 12.2.1 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 12.9. In cases where only one sequence was entered into a BLAST search, such a graphical overview is the default output.

Figure 12.9 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.

### 12.2.2 Overview BLAST table

In the overview BLAST table for a multi-sequence blast search, as shown in figure 12.10, there is one row for each query sequence. Each row represents the BLAST result for this query sequence.

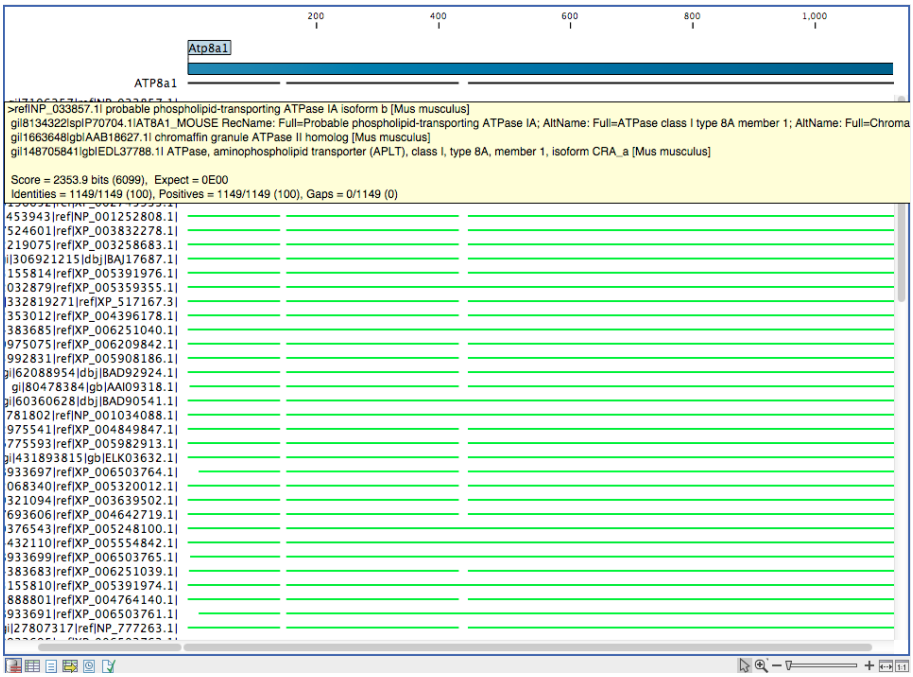



Figure 12.9: *Default display of the output of a BLAST search for one query sequence. At the top is there a graphical representation of BLAST hits with tool-tips showing additional information on individual hits.*

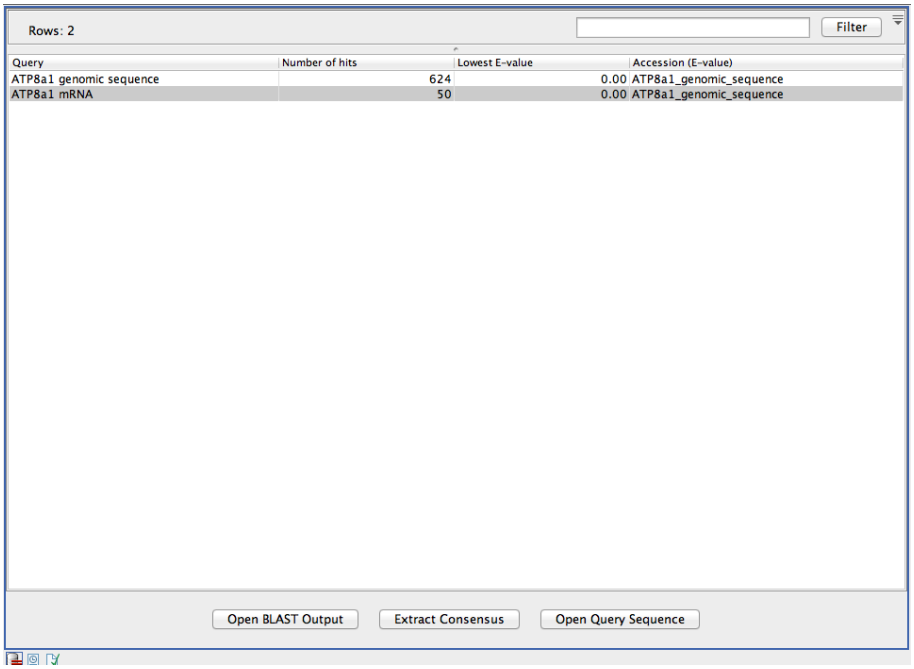



Figure 12.10: *An overview BLAST table summarizing the results for a number of query sequences.*

Double-clicking a row will open the BLAST result for this query sequence, allowing more detailed investigation of the result. You can also select one or more rows and click the **Open BLAST Output** button at the bottom of the view. Consensus sequence can be extracted by clicking the **Extract Consensus** button at the bottom. Clicking the **Open Query Sequence** will open a sequence list with the selected query sequences. This can be useful in work flows where BLAST is used as a filtering mechanism where you can filter the table to include e.g. sequences that

have a certain top hit and then extract those.

In the overview table, the following information is shown:

- Query: Since this table displays information about several query sequences, the first column is the name of the query sequence.
- Number of hits: The number of hits for this query sequence.
- For the following list, the value of the best hit is displayed together with accession number and description of this hit, 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 hit length
  - Accession (hit length)
  - Description (hit 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.

### 12.2.3 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.

- **Compactness.** You can control the level of sequence detail to be displayed:
  - **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.

- **BLAST layout.** You can 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 10.1.

Some of the information available in the tooltips 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.

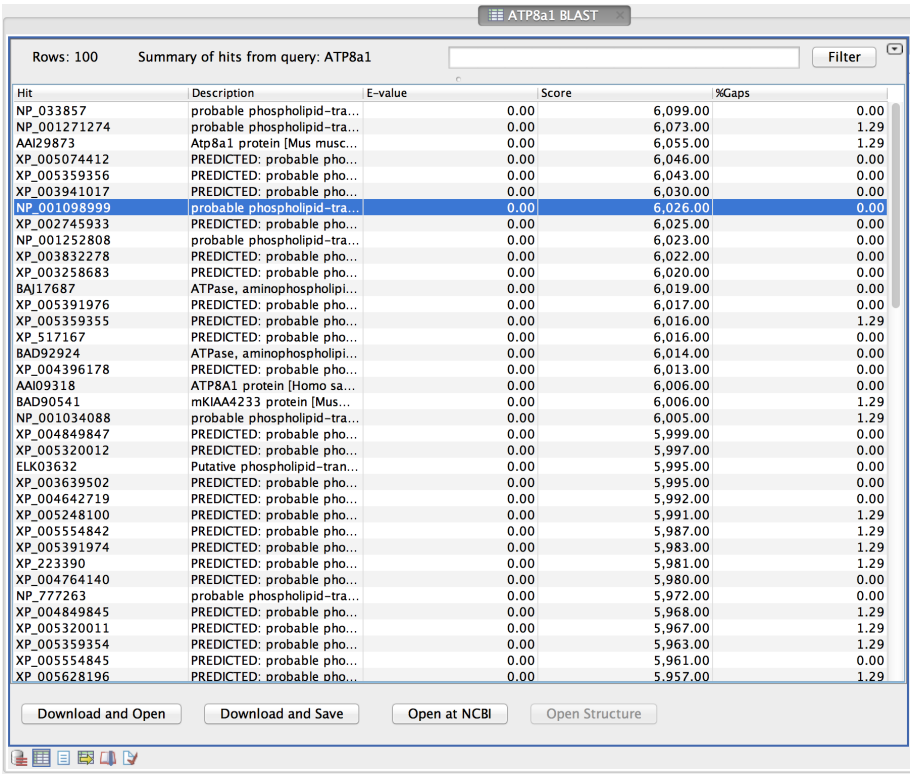
### 12.2.4 BLAST table

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 Table button ( ) at the bottom of the view**

Figure 12.11 is an example of a BLAST Table.




| Hit | Description | E-value | Score | %Gaps |
|---|---|---|---|---|
| NP_033857 | probable phospholipid-tra... | 0.00 | 6,099.00 | 0.00 |
| NP_001271274 | probable phospholipid-tra... | 0.00 | 6,073.00 | 1.29 |
| AAI29873 | Atp8a1 protein [Mus musc... | 0.00 | 6,055.00 | 1.29 |
| XP_005074412 | PREDICTED: probable pho... | 0.00 | 6,046.00 | 0.00 |
| XP_005359356 | PREDICTED: probable pho... | 0.00 | 6,043.00 | 0.00 |
| XP_003941017 | PREDICTED: probable pho... | 0.00 | 6,030.00 | 0.00 |
| NP_001098999 | probable phospholipid-tra... | 0.00 | 6,026.00 | 0.00 |
| XP_002745933 | PREDICTED: probable pho... | 0.00 | 6,025.00 | 0.00 |
| NP_001252808 | probable phospholipid-tra... | 0.00 | 6,023.00 | 0.00 |
| XP_003832278 | PREDICTED: probable pho... | 0.00 | 6,022.00 | 0.00 |
| XP_003258683 | PREDICTED: probable pho... | 0.00 | 6,020.00 | 0.00 |
| BAJ17687 | ATPase, aminophospholipi... | 0.00 | 6,019.00 | 0.00 |
| XP_005391976 | PREDICTED: probable pho... | 0.00 | 6,017.00 | 0.00 |
| XP_005359355 | PREDICTED: probable pho... | 0.00 | 6,016.00 | 1.29 |
| XP_517167 | PREDICTED: probable pho... | 0.00 | 6,016.00 | 0.00 |
| BAD92924 | ATPase, aminophospholipi... | 0.00 | 6,014.00 | 0.00 |
| XP_004396178 | PREDICTED: probable pho... | 0.00 | 6,013.00 | 0.00 |
| AAI09318 | ATP8A1 protein [Homo sa... | 0.00 | 6,006.00 | 0.00 |
| BAD90541 | mKIAA4233 protein [Mus... | 0.00 | 6,006.00 | 1.29 |
| NP_001034088 | probable phospholipid-tra... | 0.00 | 6,005.00 | 1.29 |
| XP_004849847 | PREDICTED: probable pho... | 0.00 | 5,999.00 | 0.00 |
| XP_005320012 | PREDICTED: probable pho... | 0.00 | 5,997.00 | 0.00 |
| ELK03632 | Putative phospholipid-tran... | 0.00 | 5,995.00 | 0.00 |
| XP_003639502 | PREDICTED: probable pho... | 0.00 | 5,995.00 | 0.00 |
| XP_004642719 | PREDICTED: probable pho... | 0.00 | 5,992.00 | 0.00 |
| XP_005248100 | PREDICTED: probable pho... | 0.00 | 5,991.00 | 1.29 |
| XP_005554842 | PREDICTED: probable pho... | 0.00 | 5,987.00 | 1.29 |
| XP_005391974 | PREDICTED: probable pho... | 0.00 | 5,983.00 | 1.29 |
| XP_223390 | PREDICTED: probable pho... | 0.00 | 5,981.00 | 1.29 |
| XP_004764140 | PREDICTED: probable pho... | 0.00 | 5,980.00 | 0.00 |
| NP_777263 | probable phospholipid-tra... | 0.00 | 5,972.00 | 0.00 |
| XP_004849845 | PREDICTED: probable pho... | 0.00 | 5,968.00 | 1.29 |
| XP_005320011 | PREDICTED: probable pho... | 0.00 | 5,967.00 | 1.29 |
| XP_005359354 | PREDICTED: probable pho... | 0.00 | 5,963.00 | 1.29 |
| XP_005554845 | PREDICTED: probable pho... | 0.00 | 5,961.00 | 0.00 |
| XP_005628196 | PREDICTED: probable pho... | 0.00 | 5,957.00 | 1.29 |

Figure 12.11: *Display of the output of a BLAST search in the tabular view. The hits can be sorted by the different columns, simply by clicking the column heading.*

The BLAST 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.
- **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.
- **Hit start.** Shows the start position in the hit sequence
- **Hit end.** Shows the end position in the hit sequence.

- **Hit length.** The length of the hit.
- **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 hit 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 hit sequence.
- **%Identity.** Shows the percentage of identical residues in the query and hit sequence.
- **Positive.** Shows the number of similar but not necessarily identical residues in the query and hit sequence.
- **%Positive.** Shows the percentage of similar but not necessarily identical residues in the query and hit sequence.
- **Gaps.** Shows the number of gaps in the query and hit sequence.
- **%Gaps.** Shows the percentage of gaps in the query and hit sequence.
- **Query Frame/Strand.** Shows the frame or strand of the query sequence.
- **Hit Frame/Strand.** Shows the frame or strand of the hit sequence.

In the **BLAST table** view you can handle the hit 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 hit sequence contain structure information, the sequence is opened in a text view or a 3D view (3D view in *CLC Main Workbench* or *CLC Genomics Workbench*).

The hits 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 12.2.3 so that selecting a hit in the table will make a selection on the corresponding sequence in the graphical view.

### 12.2.5 Extracting a consensus sequence from a BLAST result

It is possible to batch extract a consensus sequence from a BLAST result. This is described in more detail in section 25.8.

## 12.3 Local BLAST databases

BLAST databases on your local system can be made available for searches via your *CLC Genomics Workbench*, (section 12.3.1). To make adding databases even easier, you can download pre-formatted BLAST databases from the NCBI from within your *CLC Genomics Workbench*, (section 12.3.2). You can also easily create your own local blast databases from sequences within your *CLC Genomics Workbench*, (section 12.3.3).

### 12.3.1 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 12.4). All the files that comprise a given BLAST database must be included. This may be as few as three files, but can be more. See figure 12.12.
- Add the location where your BLAST databases are stored using the BLAST database manager (see section 12.4). See figure 12.16.

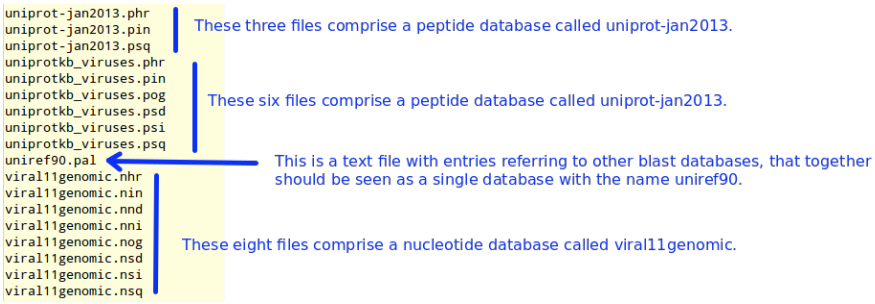



Figure 12.12: *BLAST databases are made up of several files. The exact number varies and depends on the tool used to build the databases as well as how large the database is. Large databases will be split into the number of volumes and there will be several files per volume. If you have made your BLAST 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.*

### 12.3.2 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 *CLC Genomics Workbench*.

You must be connected to the internet to use this tool.

To download a database, go to:

**Toolbox** | **BLAST** ( )| **Download BLAST Databases** ( )

A window like the one in figure 12.13 pops up showing you the list of databases available for download.

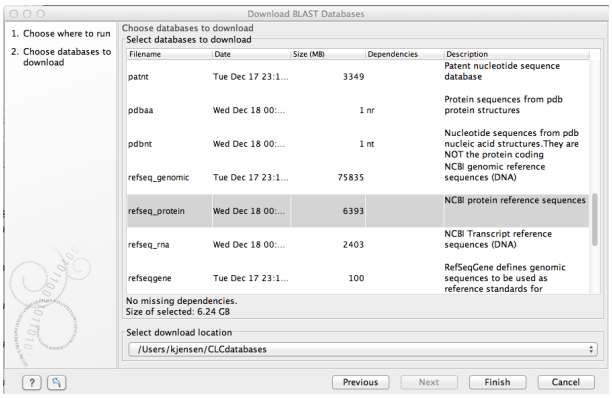



Figure 12.13: *Choose from pre-formatted BLAST databases at the NCBI available for download.*

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 12.4).

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.

### 12.3.3 Create local BLAST databases

In the *CLC Genomics Workbench* 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 12.1.3).

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 12.3.1.

To create a BLAST database, go to:

**Toolbox** | **BLAST** ( )| **Create BLAST Database** ( )

This opens the dialog seen in figure 12.14.

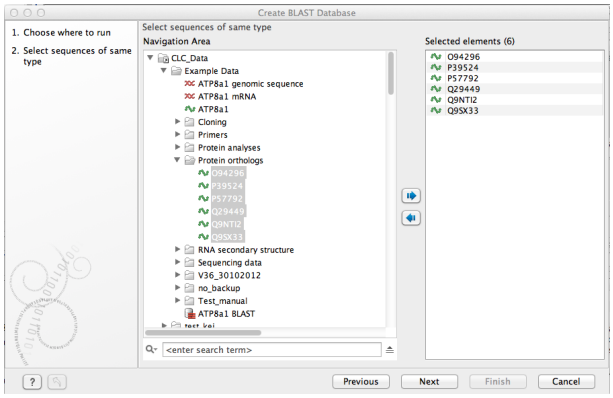



Figure 12.14: *Add sequences for the BLAST database.*

Select sequences or sequence lists you wish to include in your database and click **Next**.

In the next dialog, shown in figure 12.15, 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.** You can add more details to describe the contents of the database.
- **Location.** You can select 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 12.4.

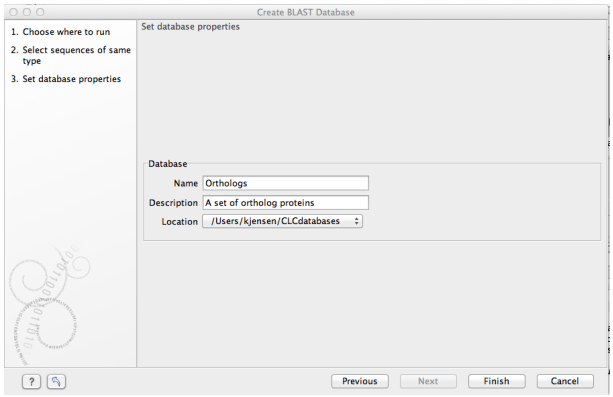



Figure 12.15: *Providing a name and description for the database, and the location to save the files to.*

Click **Finish** to create the BLAST database. Once the process is complete, the new database will be available in the **Manage BLAST Databases** dialog, see section 12.4, and when running local BLAST (see section 12.1.3).

## 12.4 Manage BLAST databases

The BLAST databases available as targets for running local BLAST searches (see section 12.1.3) can be managed through the Manage BLAST Databases dialog (see figure 12.16):

**Toolbox** | **BLAST** ( )| **Manage BLAST Databases** ( )

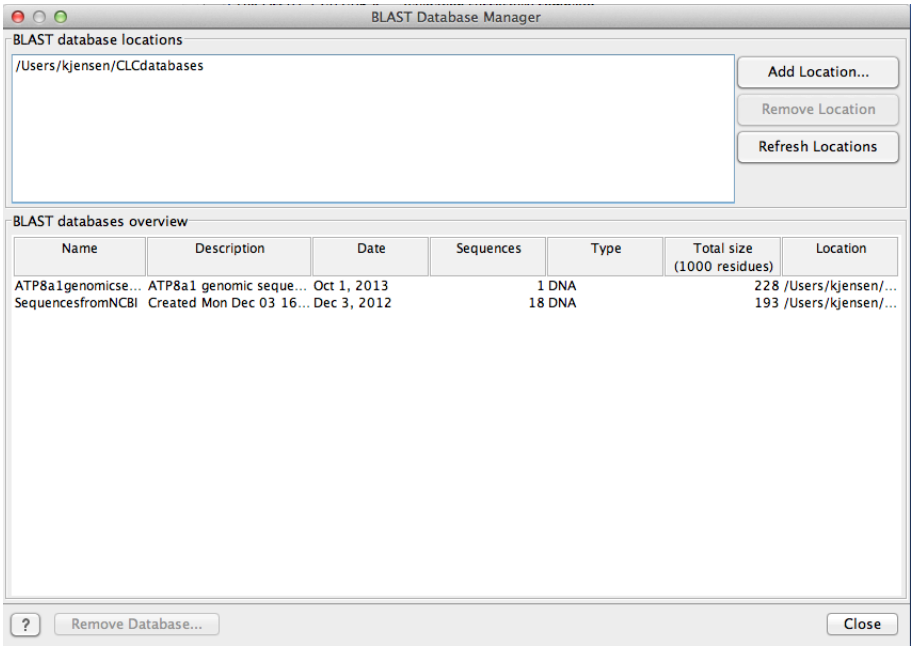



Figure 12.16: *Overview of available BLAST databases.*

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 12.3.3, 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 on Linux or Mac systems.

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.

### 12.4.1 Migrating from a previous version of the Workbench

In versions released before 2011, the BLAST database management was very different from this. In order to migrate from the older versions, please add the folders of the old BLAST databases as locations in the BLAST database manager (see section 12.4). The old representations of the BLAST databases in the **Navigation Area** can be deleted.

If you have saved the BLAST databases in the default folder, they will automatically appear because the default database location used in *CLC Genomics Workbench 8.0* is the same as the default folder specified for saving BLAST databases in the old version.

## 12.5 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, a BLAST search identifies homologous sequences by searching one or more databases usually hosted by NCBI (`http://www.ncbi.nlm.nih.gov/`), on the query sequence of interest [McGinnis and Madden, 2004].

BLAST is an open source program and anyone can download and change the program code. This has also given rise to a number of BLAST derivatives; WU-BLAST is probably the most commonly used [Altschul and Gish, 1996].

BLAST is highly scalable and comes in a number of different computer platform configurations which makes usage on both small desktop computers and large computer clusters possible.

### 12.5.1 Examples of BLAST usage

BLAST can be used for a lot of different purposes. A few of them are mentioned below.

- **Looking for species.** If you are sequencing DNA from unknown species, BLAST may help identify the correct species or homologous species.
- **Looking for domains.** If you BLAST a protein sequence (or a translated nucleotide sequence) BLAST will look for known domains in the query sequence.

- **Looking at phylogeny.** You can use the BLAST web pages to generate a phylogenetic tree of the BLAST result.
- **Mapping DNA to a known chromosome.** If you are sequencing a gene from a known species but have no idea of the chromosome location, BLAST can help you. BLAST will show you the position of the query sequence in relation to the hit sequences.
- **Annotations.** BLAST can also be used to map annotations from one organism to another or look for common genes in two related species.

### 12.5.2 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.

After the BLAST search the user will receive a report specifying found homologous sequences and their local alignments to the query sequence.

### 12.5.3 How does BLAST work?

BLAST identifies homologous sequences using a heuristic method which initially finds short matches between two sequences; thus, the method does not take the entire sequence space into account. After initial match, BLAST attempts to start local alignments from these initial matches. This also means that BLAST does not guarantee the optimal alignment, thus some sequence hits may be missed. In order to find optimal alignments, the Smith-Waterman algorithm should be used (see below). In the following, 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 12.17 for an illustration of words in a protein sequence.

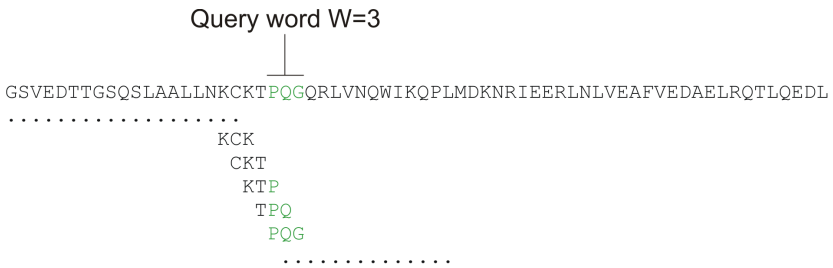



Figure 12.17: *Generation of exact BLAST words with a word size of W=3.*

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 12.17). 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 12.18). The default scoring matrix for blastp is BLOSUM62 (for explanation of scoring matrices, see www.clcbio.com/be). The compilation of exact words and neighborhood words is then used to match against the database sequences.

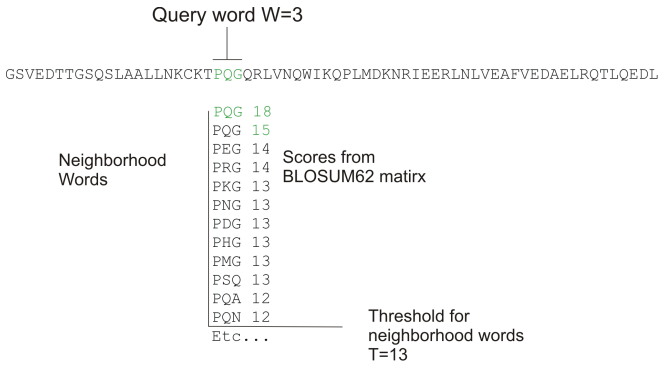



Figure 12.18: *Neighborhood BLAST words based on the BLOSUM62 matrix. Only words where the threshold* T *exceeds 13 are included in the initial seeding.*

After initial finding of words (seeding), the BLAST algorithm will extend the (only 3 residues long) alignment in both directions (see figure 12.19). Each time the alignment is extended, an alignment score is increases/decreased. When the alignment score drops below a predefined threshold, the extension of the alignment stops. This ensures that the alignment is not extended to regions where only very poor alignment between the query and hit sequence is possible. If the obtained alignment receives a score above a certain threshold, it will be included in the final BLAST result.



```
Query: 325 SLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEA     365
           +LA++L+   TP G R++ +W+  P+ D   + ER   + A
Sbjct: 290 TLASVLDCTVTPMGSRMLKRWLHMPVRDTRVLLERQQTIGA     330
```

Figure 12.19: *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 12.20.

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.

### 12.5.4 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:

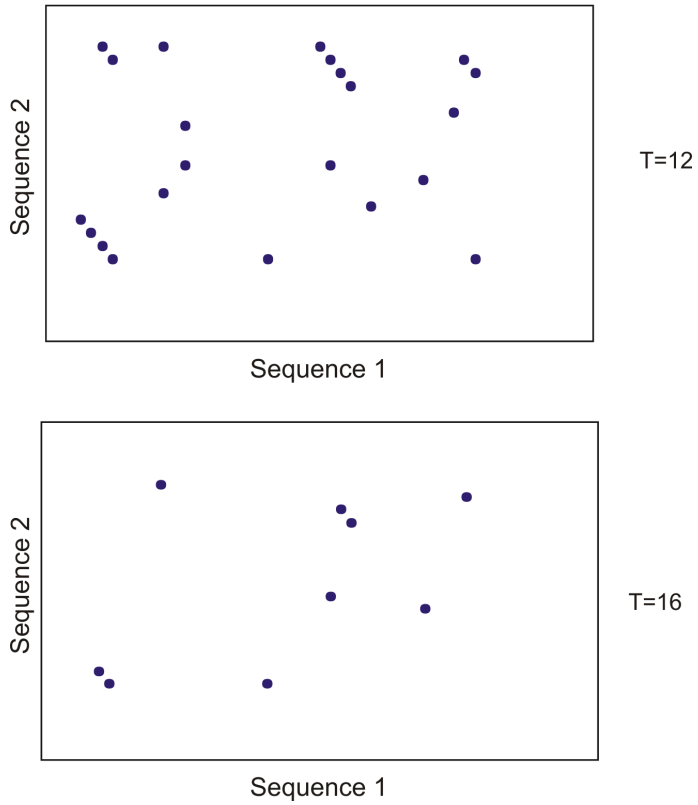



Figure 12.20: *Each dot represents a word match. Increasing the threshold of* T *limits the search space significantly.*

| 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.

### 12.5.5 Which BLAST options should I change?

The NCBI BLAST web pages and the BLAST command line tool offer a number of different options which can be changed in order to obtain the best possible result. Changing these parameters

can have a great impact on the search result. It is not the scope of this document to comment on all of the options available but merely the options which can be changed with a direct impact on the search result.

### The E-value

The *expect value*(E-value) can be changed in order to limit the number of hits to the most significant ones. The lower the E-value, the better the hit. The E-value is dependent on the length of the query sequence and the size of the database. For example, an alignment obtaining an E-value of 0.05 means that there is a 5 in 100 chance of occurring by chance alone.

E-values are very dependent 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. The default threshold for the E-value on the BLAST web page is 10. Increasing this value will most likely generate more hits. Below are some rules of thumb which can be used as a guide but should be considered with common sense.

- **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 protein is matched to the database.
- **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 homologue but it is a gray area.
- **E-value > 1** Proteins are most likely not related
- **E-value > 10** Hits are most likely junk 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 the 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

Change of 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.

Fortunately, the optimal search options for finding short, nearly exact matches can already be found on the BLAST web pages http://www.ncbi.nlm.nih.gov/BLAST/.

#### 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. See *Bioinformatics Explained* on scoring matrices on http://www.clcbio.com/be/. The default scoring matrix for blastp is BLOSUM62.

### 12.5.6 Explanation of the BLAST output

The BLAST output comes in different flavors. On the NCBI web page the default output is html, and the following description will use the html output as example. Ordinary text and xml output for easy computational parsing is also available.

The default layout of the NCBI BLAST result is a graphical representation of the hits found, a table of sequence identifiers of the hits together with scoring information, and alignments of the query sequence and the hits.

The graphical output (shown in figure 12.21) gives a quick overview of the query sequence and the resulting hit sequences. The hits are colored according to the obtained alignment scores.

The table view (shown in figure 12.22) provides more detailed information on each hit and furthermore acts as a hyperlink to the corresponding sequence in GenBank.

In the alignment view one can manually inspect the individual alignments generated by the BLAST algorithm. This is particularly useful for detailed inspection of the sequence hit found(sbjct) and the corresponding alignment. In the alignment view, all scores are described for each alignment, and the start and stop positions for the query and hit sequence are listed. The strand and orientation for query sequence and hits are also found here.

In most cases, the table view of the results will be easier to interpret than tens of sequence alignments.

### 12.5.7 I want to BLAST against my own sequence database, is this possible?

It is possible to download the entire BLAST program package and use it on your own computer, institution computer cluster or similar. This is preferred if you want to search in proprietary sequences or sequences unavailable in the public databases stored at NCBI. The downloadable BLAST package can either be installed as a web-based tool or as a command line tool. It is

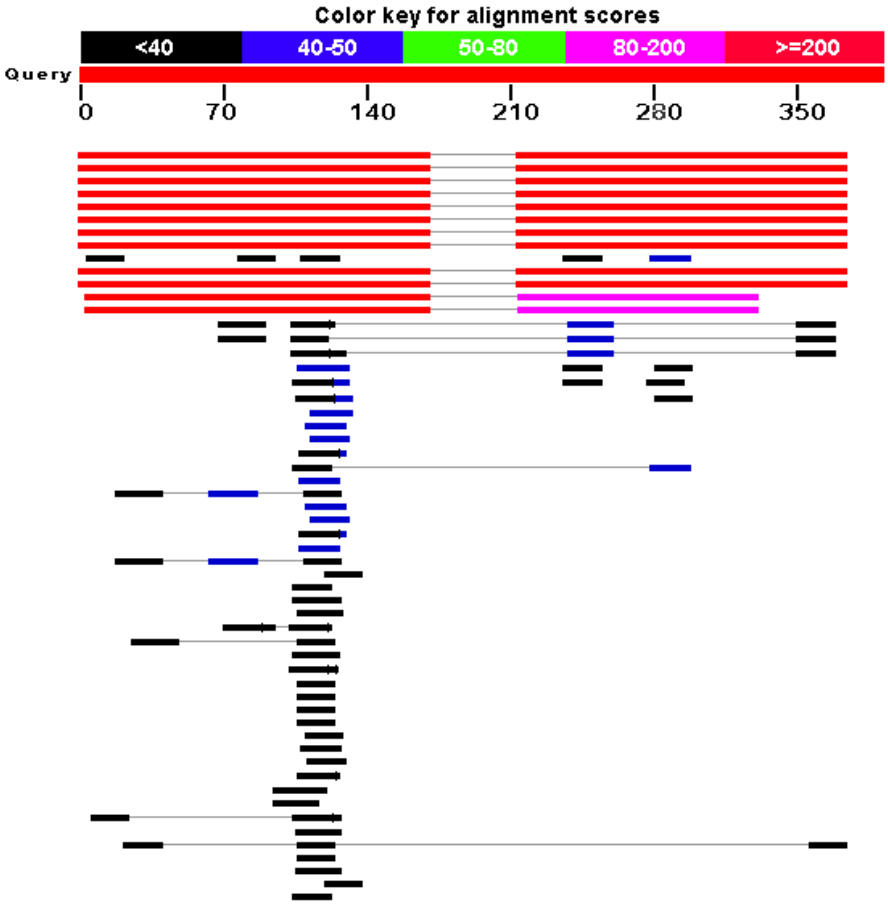



Figure 12.21: *BLAST graphical view. A simple graphical overview of the hits found aligned to the query sequence. The alignments are color coded ranging from black to red as indicated in the color label at the top.*

Sequences producing significant alignments:
(Click headers to sort columns)

| Accession | Description | Max score | Total score | Query coverage | E value | Max ident | Links |
|---|---|---|---|---|---|---|---|
| **Transcripts** | | | | | | | |
| NM_174886.1 | Homo sapiens TGFB-induced factor (TALE family homeobox) (TGIF), | 339 | 563 | 85% | 1e-90 | 100% | U E G M |
| NM_173210.1 | Homo sapiens TGFB-induced factor (TALE family homeobox) (TGIF), | 339 | 563 | 85% | 1e-90 | 100% | U E G M |
| NM_173209.1 | Homo sapiens TGFB-induced factor (TALE family homeobox) (TGIF), | 339 | 563 | 85% | 1e-90 | 100% | U E G M |
| NM_173211.1 | Homo sapiens TGFB-induced factor (TALE family homeobox) (TGIF), | 339 | 563 | 85% | 1e-90 | 100% | U E G M |
| NM_173207.1 | Homo sapiens TGFB-induced factor (TALE family homeobox) (TGIF), | 339 | 563 | 85% | 1e-90 | 100% | U E G M |
| NM_173208.1 | Homo sapiens TGFB-induced factor (TALE family homeobox) (TGIF), | 339 | 563 | 85% | 1e-90 | 100% | U E G M |
| NM_170695.2 | Homo sapiens TGFB-induced factor (TALE family homeobox) (TGIF), | 339 | 563 | 85% | 1e-90 | 100% | U E G M |
| NM_003244.2 | Homo sapiens TGFB-induced factor (TALE family homeobox) (TGIF), | 339 | 563 | 85% | 1e-90 | 100% | U E G M |
| NM_003246.2 | Homo sapiens thrombospondin 1 (THBS1), mRNA | 38.2 | 38.2 | 4% | 7.2 | 100% | U E G M |
| NM_177965.2 | Homo sapiens chromosome 8 open reading frame 37 (C8orf37), | 38.2 | 38.2 | 4% | 7.2 | 100% | U E G M |
| **Genomic sequences** [show first] | | | | | | | |
| NT_010859.14 | Homo sapiens chromosome 18 genomic contig, reference assembly | 339 | 602 | 85% | 1e-90 | 100% | |
| NW_926940.1 | Homo sapiens chromosome 18 genomic contig, alternate assembly | 339 | 602 | 85% | 1e-90 | 100% | |
| NT_011109.15 | Homo sapiens chromosome 19 genomic contig, reference assembly | 262 | 375 | 73% | 3e-67 | 94% | |
| NW_927217.1 | Homo sapiens chromosome 19 genomic contig, alternate assembly | 262 | 375 | 73% | 3e-67 | 94% | |

Figure 12.22: *BLAST table view. A table view with one row per hit, showing the accession number and description field from the sequence file together with BLAST output scores.*

available for a wide range of different operating systems.

The BLAST package can be downloaded free of charge from the following location `http://www.ncbi.nlm.nih.gov/BLAST/download.shtml`

Pre-formatted databases are available from a dedicated BLAST ftp site `ftp://ftp.ncbi.nlm.nih.gov/blast/db/`. Moreover, it is possible to download programs/scripts from the same site enabling automatic download of changed BLAST databases. Thus it is possible to schedule a nightly update of changed databases and have the updated BLAST database stored locally or on a shared network drive at all times. Most BLAST databases on the NCBI site are updated on

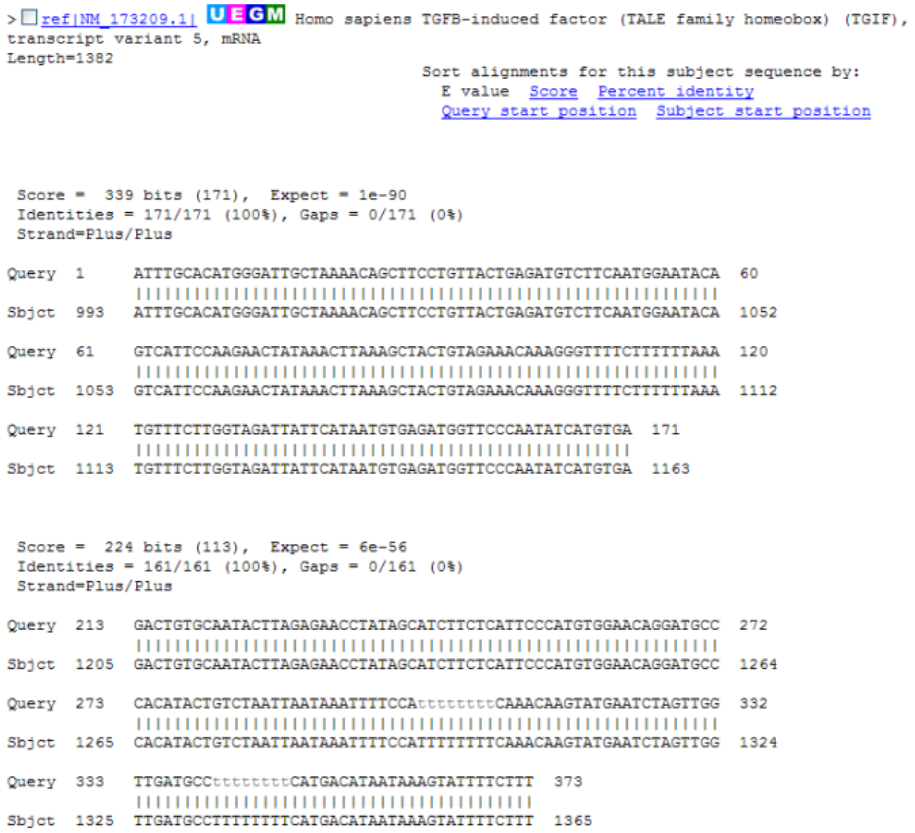



Figure 12.23: *Alignment view of BLAST results. Individual alignments are represented together with BLAST scores and more.*

a daily basis to include all recent sequence submissions to GenBank.

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 12.24). It is also much easier to batch download a selection of hit sequences for further inspection.




Figure 12.24: *Snippet of alignment view of BLAST results from CLC Main Workbench. Individual alignments are represented directly in a graphical view. The top sequence is the query sequence and is shown with a selection of annotations.*

### 12.5.8 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.

### 12.5.9 Other useful resources

The BLAST web page hosted at NCBI
http://www.ncbi.nlm.nih.gov/BLAST

Download pages for the BLAST programs
http://www.ncbi.nlm.nih.gov/BLAST/download.shtml

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/

Explanation of scoring/substitution matrices and more
http://www.clcbio.com/be/

### Creative Commons License

All CLC bio's scientific articles are licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 2.5 License. You are free to copy, distribute, display, and use the work for educational purposes, under the following conditions: You must attribute the work in its original form and "CLC bio" has to be clearly labeled as author and provider of the work. You may not use this work for commercial purposes. You may not alter, transform, nor build upon this work.




See http://creativecommons.org/licenses/by-nc-nd/2.5/ for more information on how to use the contents.

# Chapter 13

# 3D Molecule Viewer

## Contents


**13.1 Importing molecule structure files . . . . . 246**
13.1.1 From the Protein Data Bank . . . . . 247
13.1.2 From your own file system . . . . . 247
13.1.3 BLAST search against the PDB database . . . . . 248
13.1.4 Import issues . . . . . 249
**13.2 Viewing molecular structures in 3D . . . . . 250**
13.2.1 Moving and rotating . . . . . 250
13.2.2 Troubleshooting 3D graphics errors . . . . . 251
13.2.3 Updating old structure files . . . . . 252
**13.3 Customizing the visualization . . . . . 252**
13.3.1 Visualization styles and colors . . . . . 252
13.3.2 Project settings . . . . . 258
**13.4 Snapshots of the molecule visualization . . . . . 260**
**13.5 Tools for linking sequence and structure . . . . . 260**
13.5.1 Show sequence associated with molecule . . . . . 260
13.5.2 Link sequence or sequence alignment to structure . . . . . 260
13.5.3 Transfer annotations between sequence and structure . . . . . 262
**13.6 Protein structure alignment . . . . . 263**
13.6.1 The Align Protein Structure dialog box . . . . . 264
13.6.2 Example: alignment of calmodulin . . . . . 264
13.6.3 The Align Protein Structure algorithm . . . . . 266


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

## 13.1 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** (13.1.1)
- **From your own file system** (13.1.2)
- **Using BLAST search against the PDB database** (13.1.3)

### 13.1.1 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 13.1). The search hits will appear in the table below the search field.

Select the molecule structure of interest and click on the button labeled "Download and Open" (see figure 13.1) 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.

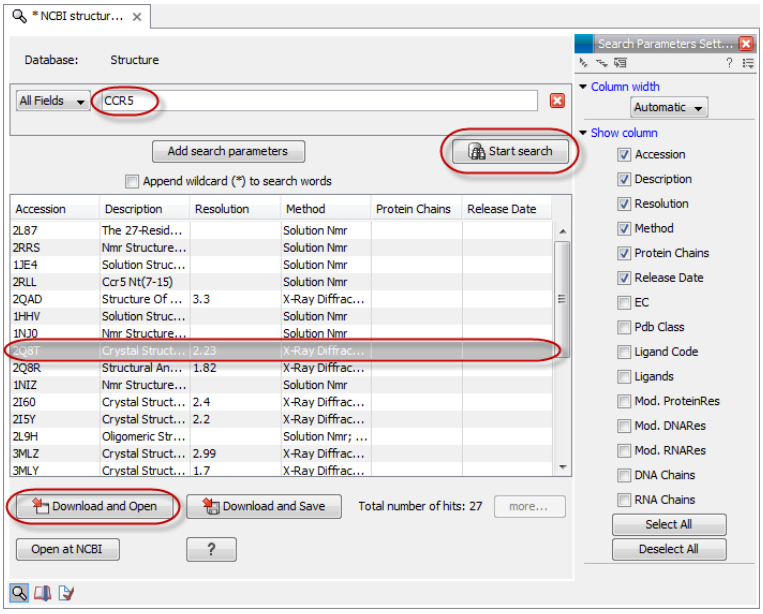



Figure 13.1: *Download protein structure from the Protein Data Bank. It is possible to open a structure file directly from the output of the search by clicking the "Download and Open" button or by double clicking directly on the relevant row.*

### 13.1.2 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 13.2). 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**.

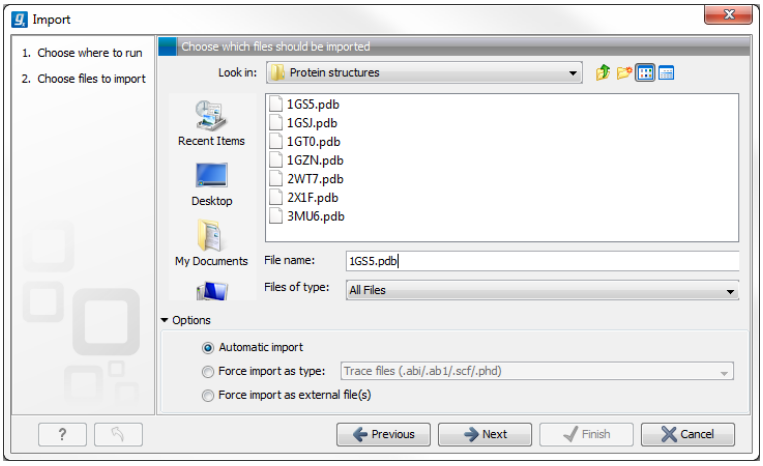



Figure 13.2: *A PDB file can be imported using the "Standard Import" function.*

### 13.1.3 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 13.3).

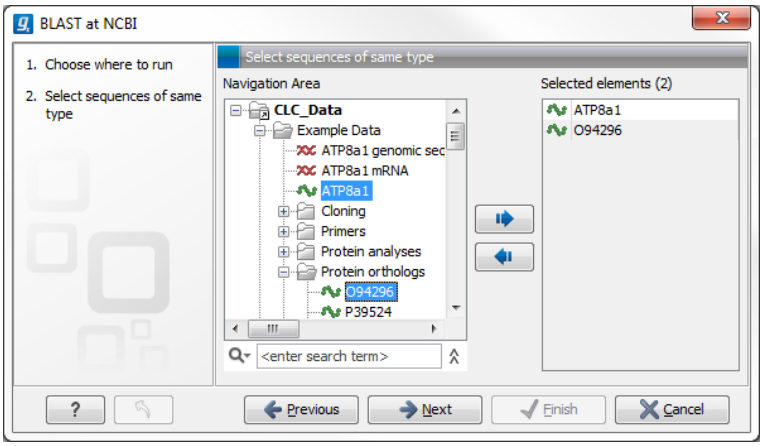



Figure 13.3: *Select the input sequence of interest. In this example a protein sequence for ATPase class I type 8A member 1 and an ATPase ortholog from S. pombe have been selected.*

Click **Next** and choose program and database (figure 13.4). 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.

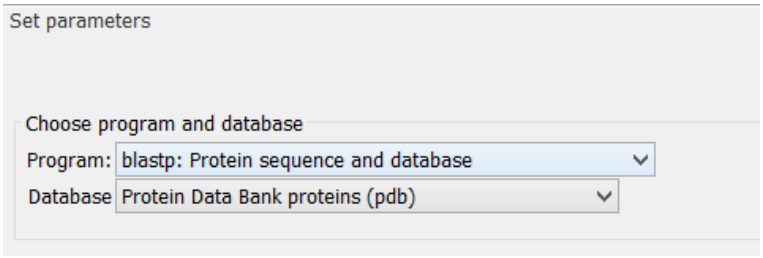



Figure 13.4: *Select database and program.*

Please refer to section 12.1.1 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.

**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 13.5). 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**.

### 13.1.4 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 13.6).

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.

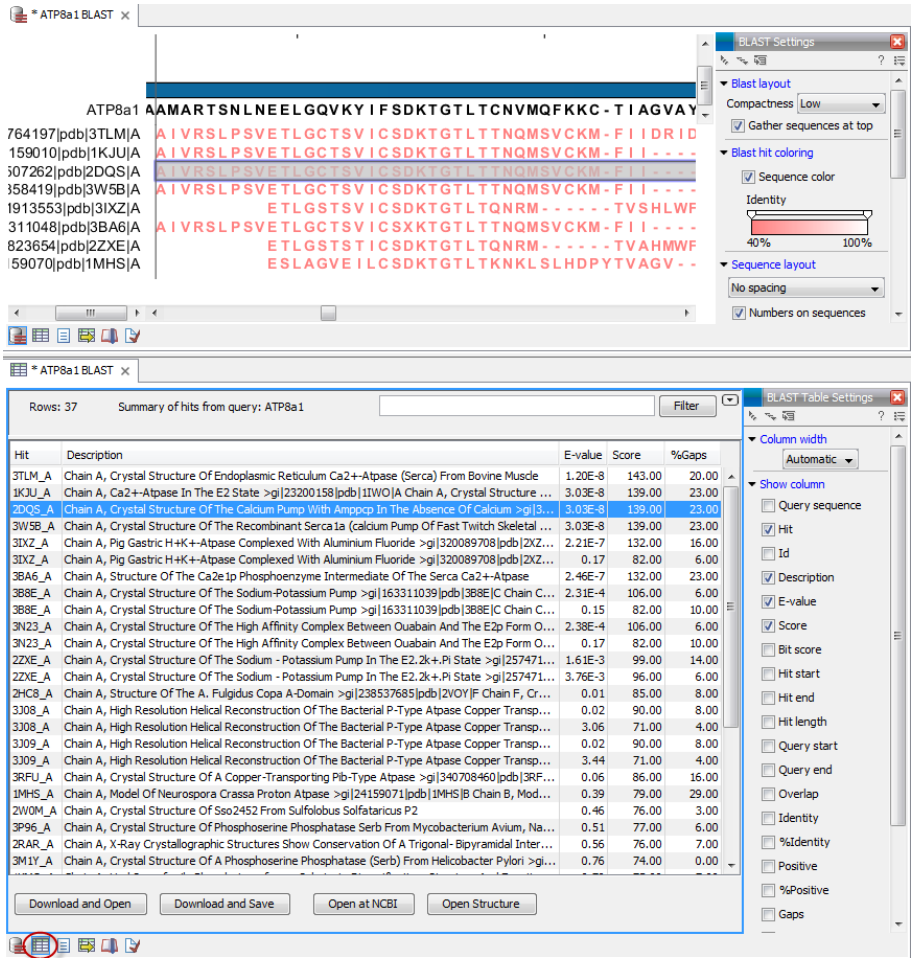



Figure 13.5: *Top: The output from "BLAST at NCBI". Bottom: The "BLAST table". One of the protein sequences has been selected. This activates the four buttons under the table. Note that the table and the BLAST Graphics are linked, this means that when a sequence is selected in the table, the same sequence will be highlighted in the BLAST Graphics view.*

## 13.2 Viewing molecular structures in 3D

An example of a 3D structure that has been opened as a **Molecule Project** is shown in figure 13.7.

### 13.2.1 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.

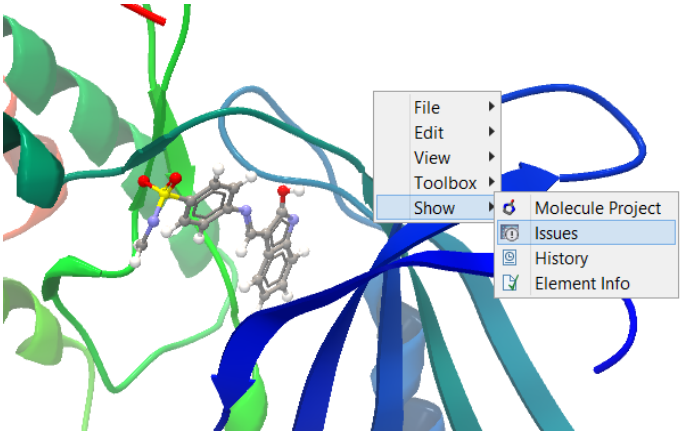



Figure 13.6: *At the bottom of the Molecule Project it is possible to switch to the "Show Issues" view by clicking on the "table-with-exclamation-mark" icon.*

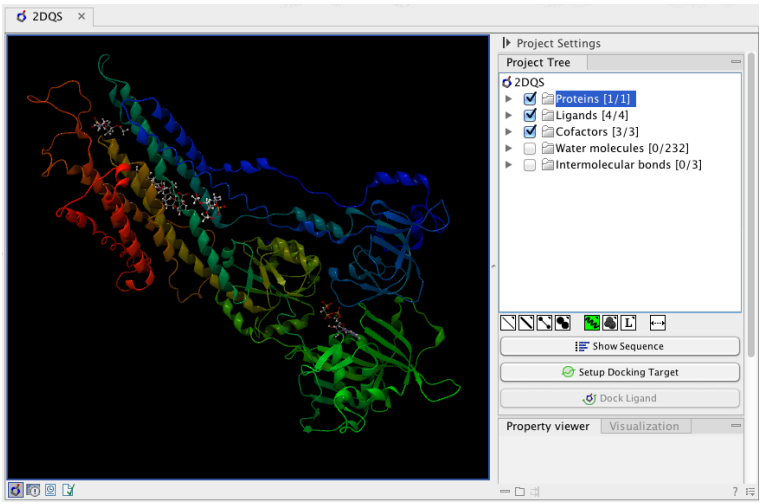



Figure 13.7: *3D view of a calcium ATPase. All molecules in the PDB file are shown in the Molecule Project. The Project Tree in the right side of the window lists the involved molecules.*

### 13.2.2 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.

### 13.2.3 Updating old structure files

A completely redesign of the 3D Molecule Viewer was released in August 2013. It is therefore necessary to update older structure files. To update existing structure files, double click on the name in the **Navigation Area**. This will bring up the dialog shown in figure 13.8, which via the "Download from PDB..." button gives access to downloading the specific structure in PDB format.

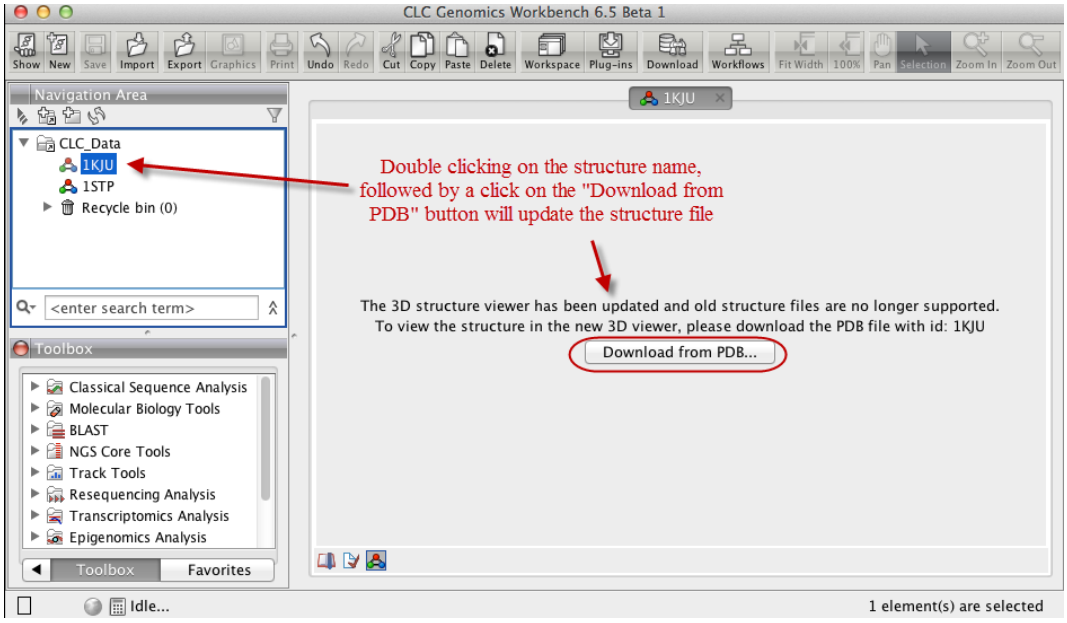



Figure 13.8: *Old structure files are not supported by the new 3D Molecule Viewer and must be updated.*

## 13.3 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.

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**.

**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 on the entry.

### 13.3.1 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 visualization style icons:

- **Color by Element** Classic CPK coloring based on atom type (e.g. oxygen red, carbon gray, hydrogen white, nitrogen blue, sulfur yellow).
- **Color by Temperature** For PDB files, this is based on the b-factors. For structure models created with tools in a CLC workbench, this is based on an estimate of the local model quality. The color scale goes from blue (0) over white (50) to red (100). The b-factors as well as the local model quality estimate are measures of uncertainty or disorder in the atom position; the higher the number, the higher the uncertainty.
- **Color Carbons by Entry** Each entry (molecule or atom group) is assigned its own specific color. Only carbon atoms are colored by the specific color, other atoms are colored by element.
- **Color by Entry** Each entry (molecule or atom group) is assigned its own specific color.
- **Custom Color** The user selects a molecule color from a palette.
- **Custom Carbon Color** The user selects a molecule color from a palette. Only carbon atoms are colored by the specific color, other atoms are colored by element.

#### Backbone

()

For the molecules in the Proteins and Nucleic Acids categories, the backbone structure can be visualized in a schematic rendering, highlighting the secondary structure elements for proteins and matching base pairs for nucleic acids. The backbone visualization can be combined with any of the atom-level visualizations.

Five color schemes are available for backbone structures:

- **Color by Residue Position** Rainbow color scale going from blue over green to yellow and red, following the residue number.
- **Color by Type** For proteins, beta sheets are blue, helices red and loops/coil gray. For nucleic acids backbone ribbons are white while the individual nucleotides are indicated in green (T/U), red (A), yellow (G), and blue (C).
- **Color by Backbone Temperature** For PDB files, this is based on the b-factors for the C$\alpha$ 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 making a custom atom group that includes all atoms from the molecules (see section 13.3.1)

### 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 13.9).

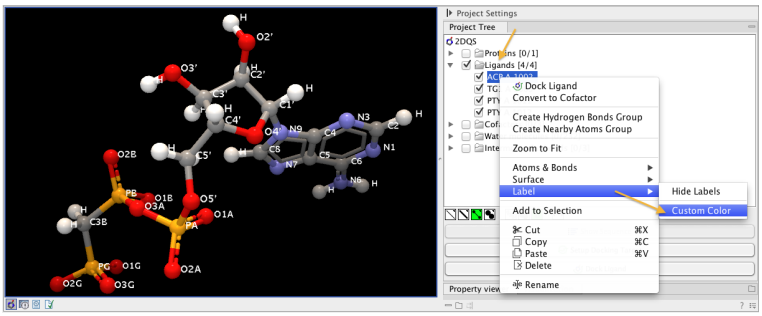



Figure 13.9: *The color of the labels can be adjusted in two different ways. Either directly using the label button by right clicking the button, or by right clicking on the molecule or category of interest in the Project Tree.*

- For proteins and nucleic acids, each residue is labelled with the PDB name and number.
- For ligands, each atom is labelled 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 labelled with the PDB name and number.
- For atom groups not including protein atoms, each atom is labelled with the atom name as given in the input.

Labels can be removed again by clicking on the label button.

### 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 13.10). Double-clicking an entry in the **Project Tree** will have the same effect.

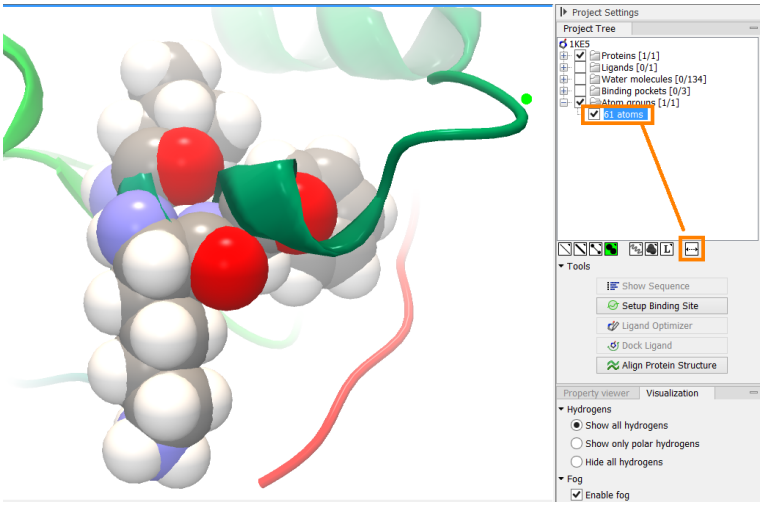



Figure 13.10: *The "Fit to screen" button can be used to bring a particular molecule or category of molecules in focus.*

### Custom atom groups

In some situations it may be relevant 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 done by making an atom group. Atoms groups can be created in different ways, as listed below. 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 molecules in the **Project Tree**.

Atom groups can be deleted using the context menu or the 'Delete' key, when they are selected in the Project Tree.

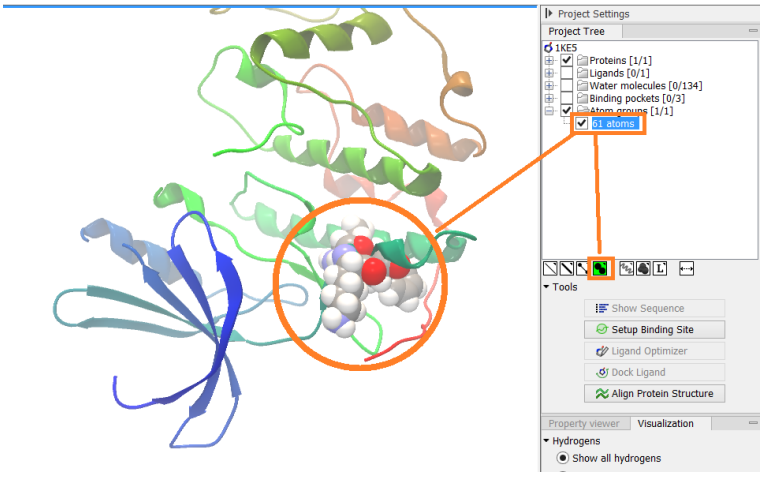



Figure 13.11: *An atom group that has been highlighted by adding a unique visualization style.*

### How to create atom groups

Atom groups can be created in different ways. 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**.

### Convert a selection to an atom group

To convert the current selection into an atom group, select the Current entry in the **Project Tree**, and from the right-click context menu choose 'Create Group from Selection' to make an atom group with exactly the selected atoms, or choose 'Create Group from Selection plus Context' to include whole residues or molecules in the atom group.

Note that an atom group can be renamed. Select the atom group in the Project Tree and invoke the right-click context menu. Here, the Rename option is found.

- **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. The atom selection can then be converted to an atom group as described above.
- **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, whole molecules can be removed from the current selection via the context menu. The atom selection can then be converted to an atom group as described above.
- **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 visible atoms inside the sphere will be selected. The status bar (lower right corner) will show the radius of the sphere. The current atom selection can then be converted to an atom group as described above.

- **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 13.5.1). A split-view will appear with a sequence editor for each of the sequence data types (Protein, DNA, RNA) (figure 13.12). If you go to the protein (or nucleic acid) sequence that was opened with the "Show Sequence" button and select a region in the sequence, the selected residues will show up as the "Current" selection in the 3D view and the Project Tree view. The atom selection can then be converted to an atom group as described above. Notice that the link between the 3D view and the sequence editor is lost if either window is closed, or if the sequences are modified.

- **Align to Existing Sequence** If a single protein chain is selected, the "Align to Existing Sequence" button can be clicked (section 13.5.2). 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 atom groups can be created as for the "Show Sequence" option.

- **Atom group with nearby atoms** When just one molecule is selected in the Project Tree, the context menu on the molecule entry contains an option to create an atom group comprising molecules and residues within 5 Å of the selected molecule (Create Nearby Atoms Group). This will create an entry in the Atom groups category. A 'Nearby Atoms Group' can also be created from binding pocket entries (binding pockets can only be created in *CLC Drug Discovery Workbench*).

- **Atom group with interacting binding site atoms** If a Binding Site Setup is present in the Project Tree, the context menu on ligands and docking results has a 'Binding Site Interactions' menu. Here, the 'Create Interacting Atoms Group' will create an atom group containing molecules and residues included in the Binding Site Setup, and within 5 Å of the selected molecule. A Binding Site Setup can only be created using the *CLC Drug Discovery Workbench*.

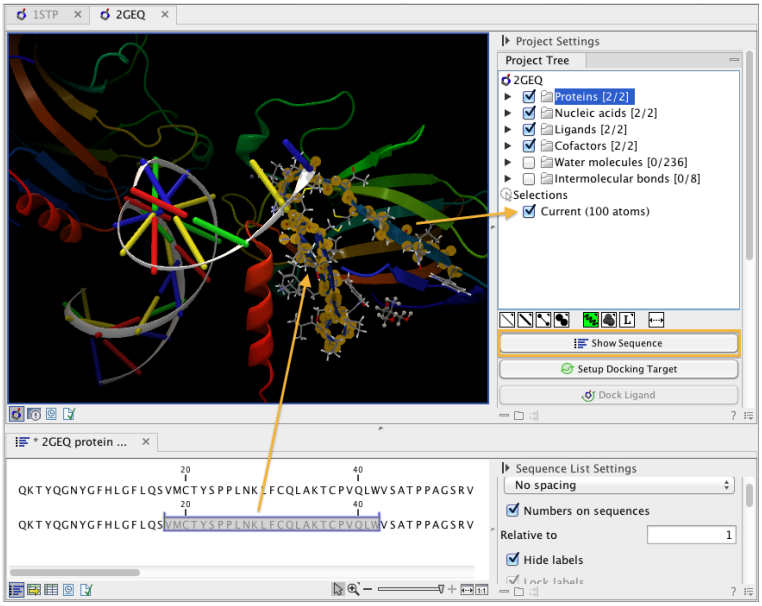



Figure 13.12: *The protein sequence in the split view is linked with the protein structure. This means that when a part of the protein sequence is selected, the same region in the protein structure will be selected.*

### 13.3.2 Project settings

A number of general settings can be adjusted from the **Side Panel**. Personal settings as well as molecule visualizations can be saved by clicking in the lower right corner of the **Side Panel** ( ). This is described in detail in section 4.6.

#### Project Tree Tools

Just below the Project Tree, the following tools are available

- **Show Sequence** Select molecules which have sequences associated (Protein, DNA, RNA) in the Project Tree, and click this button. Then, a split-view will appear with a sequence editor for each of the sequence data types (Protein, DNA, RNA). This is described in section 13.5.1.
- **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 13.5.2.
- **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 13.5.3.
- **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 13.6.

#### 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 27.15.2. 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.

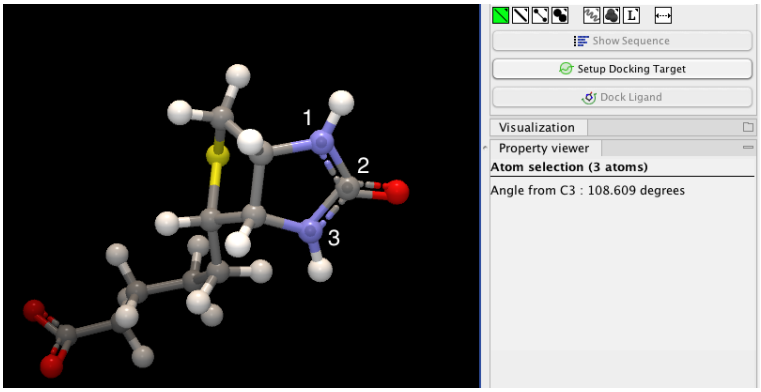



Figure 13.13: *Selecting two, three, or four atoms will display the distance, angle, or dihedral angle, respectively.*

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" four 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.
- **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.

## 13.4 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).

## 13.5 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 13.5.3).

### 13.5.1 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 13.14). 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 13.3.1. 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 13.5.2).

### 13.5.2 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 13.5.1) or Align to Existing Sequence.

Select a single protein chain in the project tree (see figure 13.15). 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.

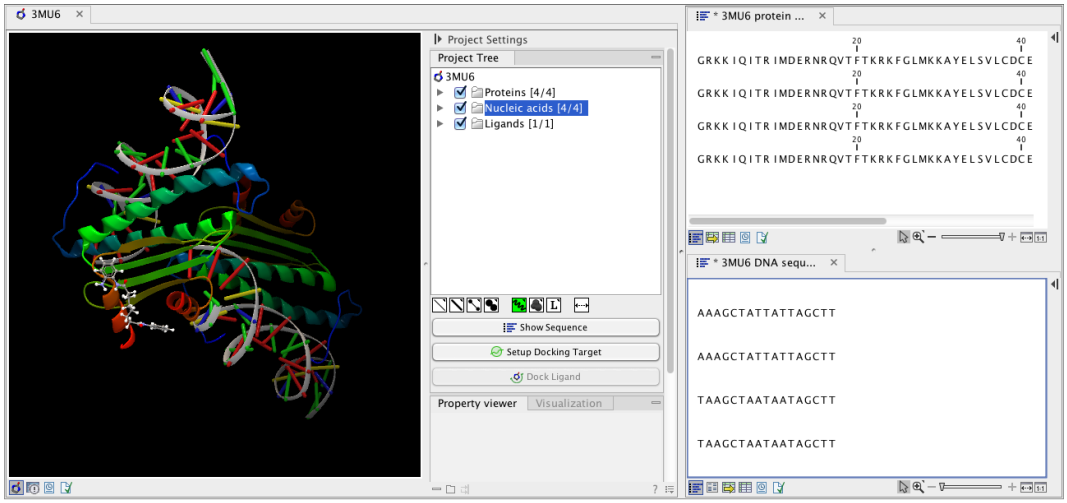



Figure 13.14: *Protein chain sequences and DNA sequences are shown in separate views.*

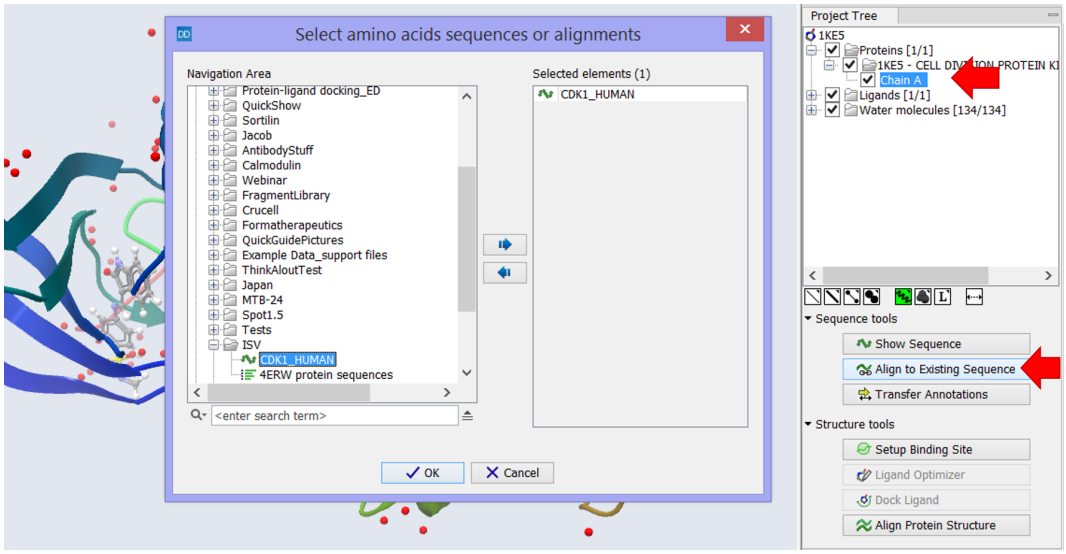



Figure 13.15: *Select a single protein chain in the Project Tree and invoke "Align to Existing Sequence".*

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 13.5.1), 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 20.1) 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 13.5.3). Notice, 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 20.2
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 13.18 and section 20.3.4).

### 13.5.3 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 10.3 and more about atom groups in section 13.3.1.

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 13.5.1) or by mapping to an existing sequence or sequence alignment using the 'Align to Existing Sequence' button (see section 13.5.2).

Invoke the Transfer Annotations dialog by selecting a linked protein chain in the Project Tree and press 'Transfer Annotations' (see figure 13.16).

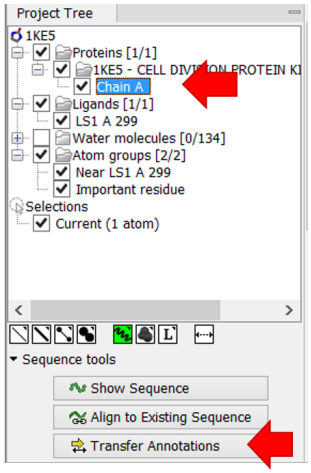



Figure 13.16: *Select a single protein chain in the Project Tree and invoke "Transfer Annotations".*

The dialog contains two tables (see figure 13.17). 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**

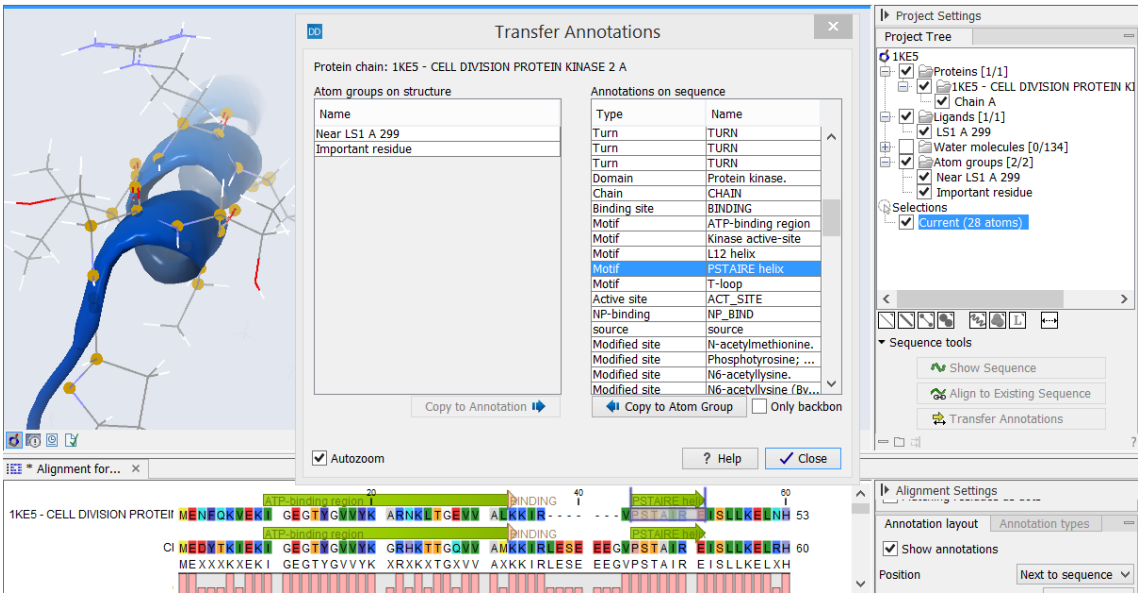



Figure 13.17: *The Transfer Annotations dialog allow you to select annotations listed in the two tables, and copy them from structure to sequence or vice versa.*

In order to undo operations made using the Transfer Annotations dialog, the dialog must first be closed. To undo atom groups added to the structure, activate the 3D view by clicking in it and press Undo in the Toolbar. To undo annotations added to the sequence, activate the sequence view by clicking in it and press Undo in the Toolbar.

### Transfer sequence annotations from aligned sequences

It is only annotations present on the sequence linked to the 3D view that can be transferred to atom groups on the structure. If you wish to transfer annotations that are found on other sequences in a linked sequence alignment, you need first to copy the sequence annotations to the actual sequence linked to the 3D view (the sequence with the same name as the protein structure). This is done by invoking the context menu on the sequence annotation you wish to copy (see figure 13.18 and section 20.3.4 ).

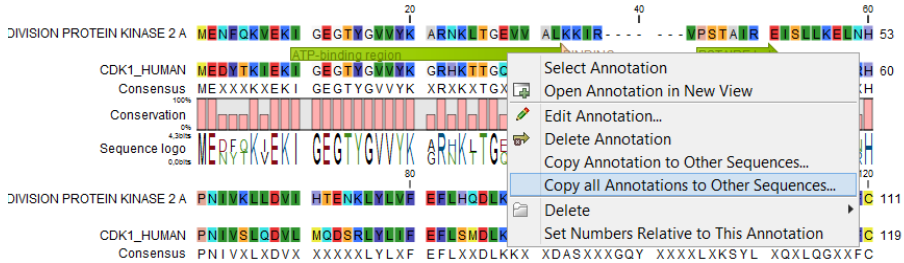



Figure 13.18: *Copy annotations from sequences in the alignment to the sequence linked to the 3D view.*

## 13.6 Protein structure alignment

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 13.19). 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.

### 13.6.1 The Align Protein Structure dialog box

The dialog box contains three fields:

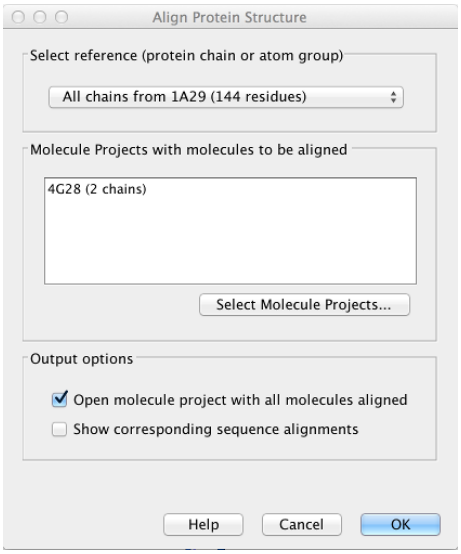



Figure 13.19: *The Align Protein Structure dialog box.*

- **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 alignment between each project and the reference may be output.

### 13.6.2 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 13.19. 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 13.20. 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).

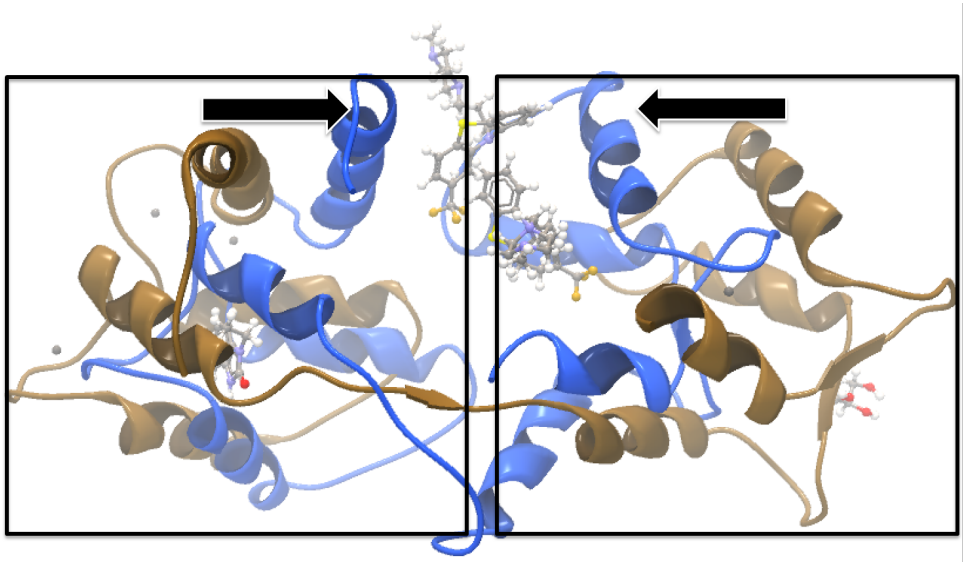


Figure 13.20: *Global alignment of two calmodulin structures (blue and brown). The two domains of calmodulin (shown within black boxes) can undergo large changes in relative orientation. In this case, the different orientation of the domains in the blue and brown structures makes a good global alignment 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 13.21). Using the new atom group as the reference in the alignment dialog leads to the alignment shown in figure 13.22. In addition to the original input proteins, the output now includes two Atom Groups, which contain the atoms on which the alignment was focused. The **History** of the output **Molecule Project** shows that the alignment has 0.9 Å RMSD over the 62 residues.

**Aligning a binding site** Two bound calcium atoms, one from each calmodulin structure, are shown in the black box of figure 13.22. 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 13.23.

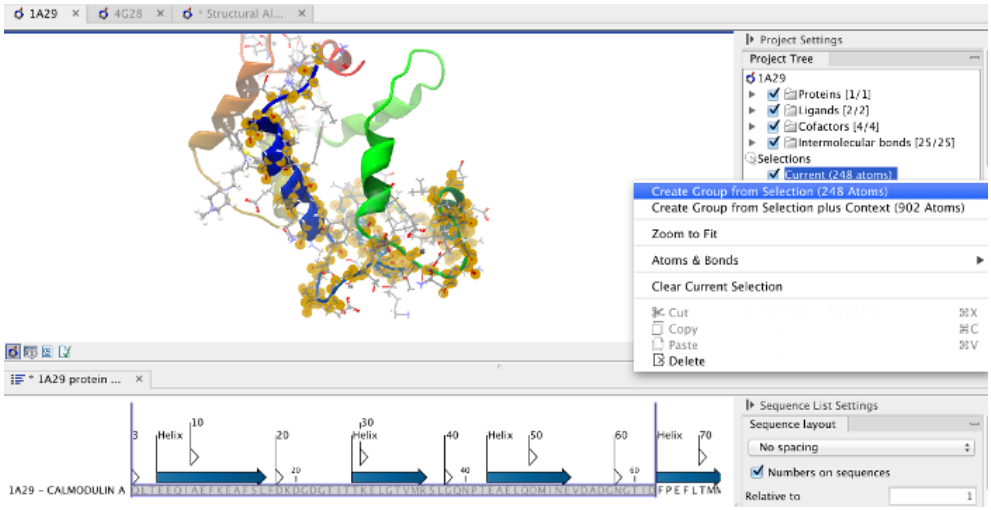



Figure 13.21: *Creation of an atom group containing the N-terminal domain of calmodulin.*

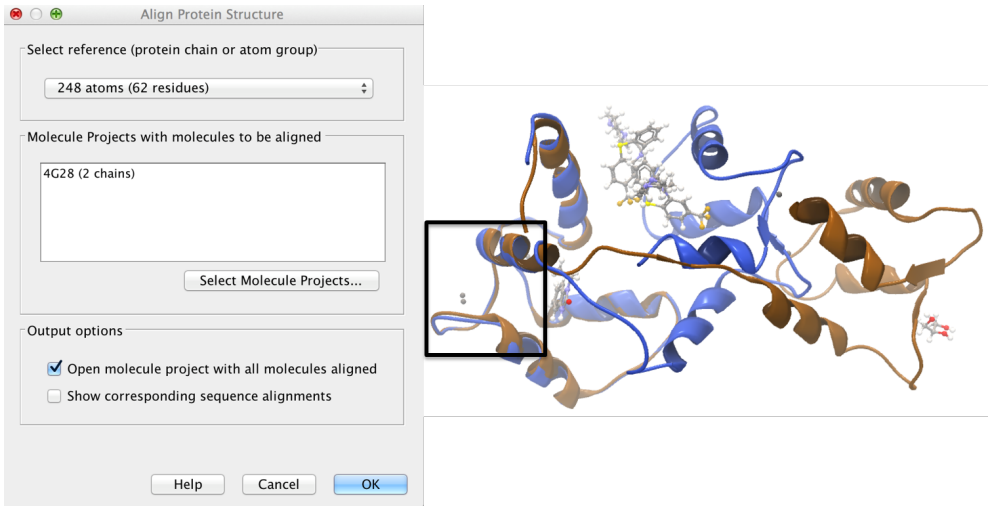



Figure 13.22: *Alignment of the same two calmodulin proteins as in figure 13.20, but this time with a focus on the N-terminal domain. The blue and brown structures are now well-superimposed in the N-terminal region. The black box encloses two calcium atoms that are bound to the structures.*

### 13.6.3 The Align Protein Structure algorithm

Any approach to structure alignment must make a trade-off between alignment length and alignment accuracy. For example, is it better to align 200 amino acids at an RMSD of 3.0 Å or 150 amino acids at an RMSD of 2.5 Å? The Align Protein Structure algorithm determines the answer to this question by taking the alignment with the higher TM-score. For an alignment focused on a protein of length $L$, this is:

$$\text{TM-score} = \frac{1}{L}\sum_i \frac{1}{1+\frac{d_i}{d(L)}^2}$$

where $i$ runs over the aligned pairs of residues, $d_i$ is the distance between the $i^{th}$ such pair, and $d(L)$ is a normalization term that approximates the average distance between two randomly chosen points in a globular protein of length $L$ [Zhang and Skolnick, 2004]. A perfect alignment has a TM-score of 1.0, and two proteins with a TM-score $>$0.5 are often said to show structural

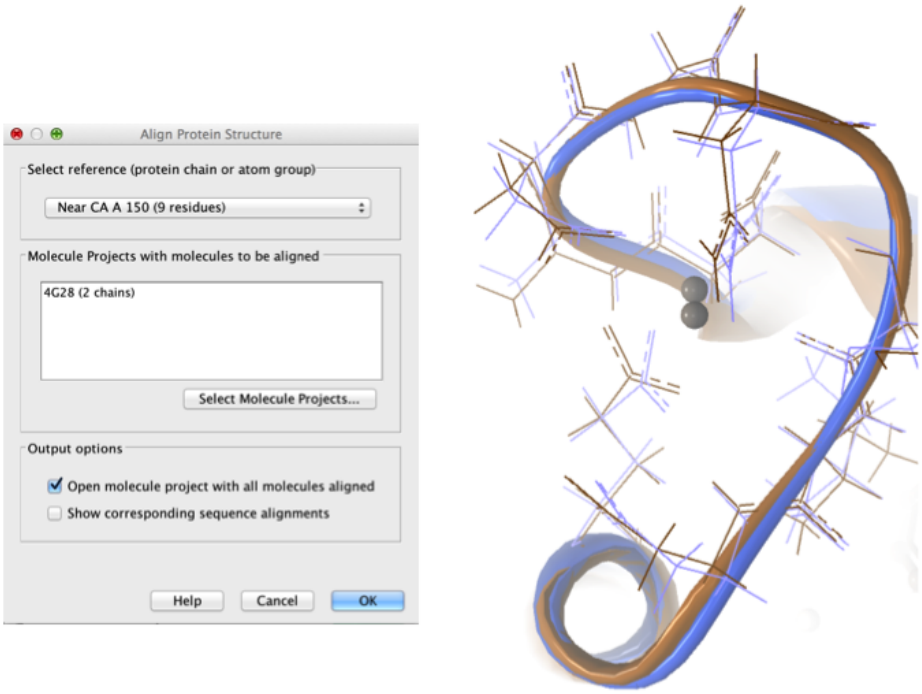



Figure 13.23: *Alignment of the same two calmodulin domains as in figure 13.20, but this time with a focus on the calcium atom within the black box of figure 13.22. The calcium atoms are less than 1 Å apart – compatible with thermal motion encoded in the atoms' temperature factors.*

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).

# Chapter 14

# General sequence analyses

**Contents**


**14.1 Extract Annotations . . . 268**
**14.2 Extract sequences . . . 270**
**14.3 Shuffle sequence . . . 272**
**14.4 Dot plots . . . 274**
14.4.1 Create dot plots . . . 274
14.4.2 View dot plots . . . 275
14.4.3 Bioinformatics explained: Dot plots . . . 277
14.4.4 Bioinformatics explained: Scoring matrices . . . 281
**14.5 Local complexity plot . . . 284**
**14.6 Sequence statistics . . . 285**
14.6.1 Bioinformatics explained: Protein statistics . . . 288
**14.7 Join sequences . . . 291**
**14.8 Pattern discovery . . . 292**
14.8.1 Pattern discovery search parameters . . . 293
14.8.2 Pattern search output . . . 294
**14.9 Motif Search . . . 294**
14.9.1 Dynamic motifs . . . 295
14.9.2 Motif search from the Toolbox . . . 296
14.9.3 Java regular expressions . . . 298
**14.10 Create motif list . . . 300**


*CLC Genomics Workbench* offers different kinds of sequence analyses that apply to both protein and DNA. The analyses are described in this chapter.

## 14.1 Extract Annotations

The **Extract annotations** tool makes it very easy to extract parts of a sequence (or several sequences) based on its annotations. Using a few steps it is possible to:

- extract e.g. all tRNA genes from the *E. coli* genome.

- automatically add flanking regions to the annotated sequences.
- search for specific words in all available annotations.

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** | **Classical Sequence Analysis** (  ) | **General Sequence Analysis** (  )| **Extract Annotations** (  )

This opens the dialog shown in figure 14.1 that asks for either an annotated sequence or an annotation or variant track.

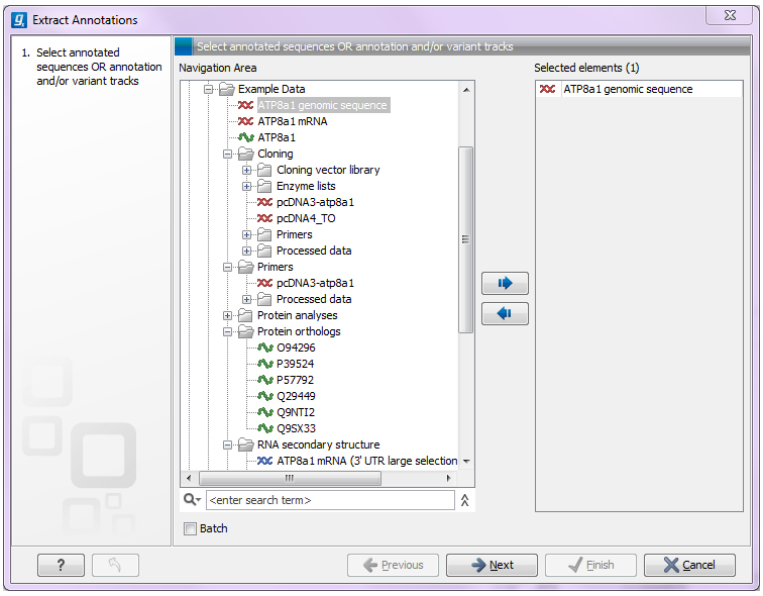



Figure 14.1: *Select one or more annotated sequence or annotation or variant tracks.*

If you selected tracks as input, the next step will ask for a sequence track to use for extracting the annotations.

Click **Next**. At the top of the dialog shown in figure 14.2 you can specify a sequence track (in case a track was selected as input), or which annotations to use if an annotated sequence was selected as input:

- **Search term**. 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. If you e.g. 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, e.g. "MLH1, Human" will find annotations including both "MLH1" and "Human".
- **Annotation types** If only certain types of annotations should be extracted, this can be specified here.

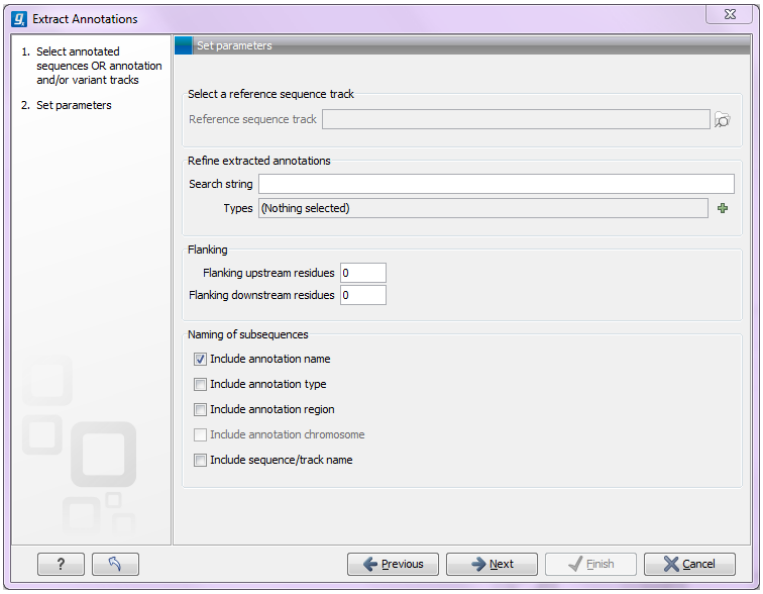



Figure 14.2: *Adjusting parameters for extract annotations.*

The sequence of interest can be extracted with flanking sequences:

- **Flanking upstream residues.** The output will include this number of extra residues at the 5' end of the annotation.
- **Flanking downstream residues.** The output will include this number of extra residues at the 3' end of the annotation.

The sequences that are created can be named after the annotation name, type etc:

- **Include annotation name.** This will use the name of the annotation in the name of the extracted sequence.
- **Include annotation type.** This corresponds to the type chosen above and will put this information in the name of the resulting sequences. This is useful information if you have chosen to extract "All" types of annotations.
- **Include annotation region.** The region covered by the annotation on the original sequence (i.e. not including flanking regions) will be included in the name.
- **Include sequence/track name.** If you have selected more than one sequence as input, this option enables you to discern the origin of the resulting sequences in the list by putting the name of the original sequence into the name of the resulting sequences.

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

## 14.2 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
- BLAST overview tables
- sequence lists
- Contigs and read mappings
- Read mapping tables
- Read mapping tracks
- RNA-Seq mapping results

**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 18.7.6 that describes the function "Extract from Selection" that also can be selected from the right click menu (see figure 14.3).

For extracting a subset of a sequence list, you can highlight the sequences of interest in the table view of the sequence list, right click on the selection and launch the Extract Sequences tool.

The Extract Sequences tool can be launched via the Toolbox menu, by going to:

> **Toolbox** | **Classical Sequence Analysis** () | **General Sequence Analysis** ()| **Extract Sequences** ()

Alternatively, on all the data types listed above except sequence lists, the option to run this tool appears by right clicking in the relevant area; a row in a table or in the read area of mapping data. An example is shown in figure 14.3.

Please note that for mappings, only the read sequences are extracted. Reference and consensus sequences are not extracted using this tool. Similarly, when extracting sequences from BLAST results, the sequence hits are extracted, not the original query sequence or a consensus sequence.

"Note also, 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 (e.g. 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."

The dialog allows you to select the **Destination**. Here you can 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. The exception to this is when working with a sequence list, where choosing to extract to a sequence list would create a copy of the same sequence list. In this case, the other option would generally be chosen. This would then result in the generation of individual sequence objects for each sequence in the sequence list.

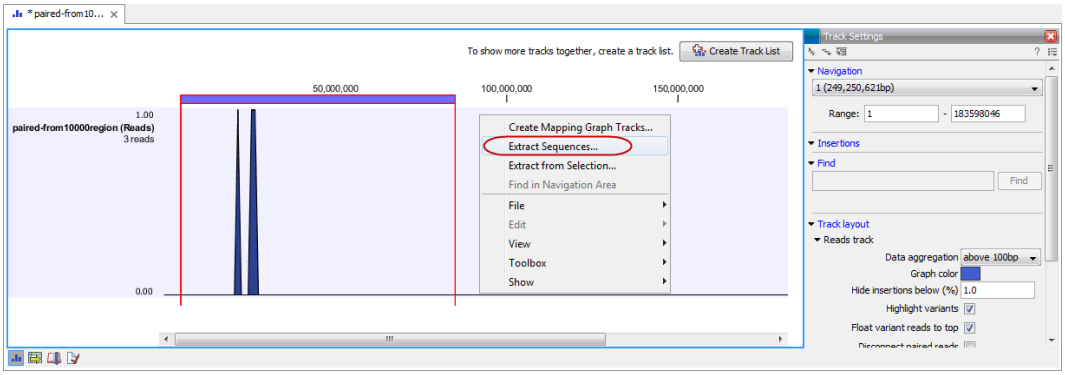



Figure 14.3: *Right click somewhere in the reads track area and select "Extract Sequences".*

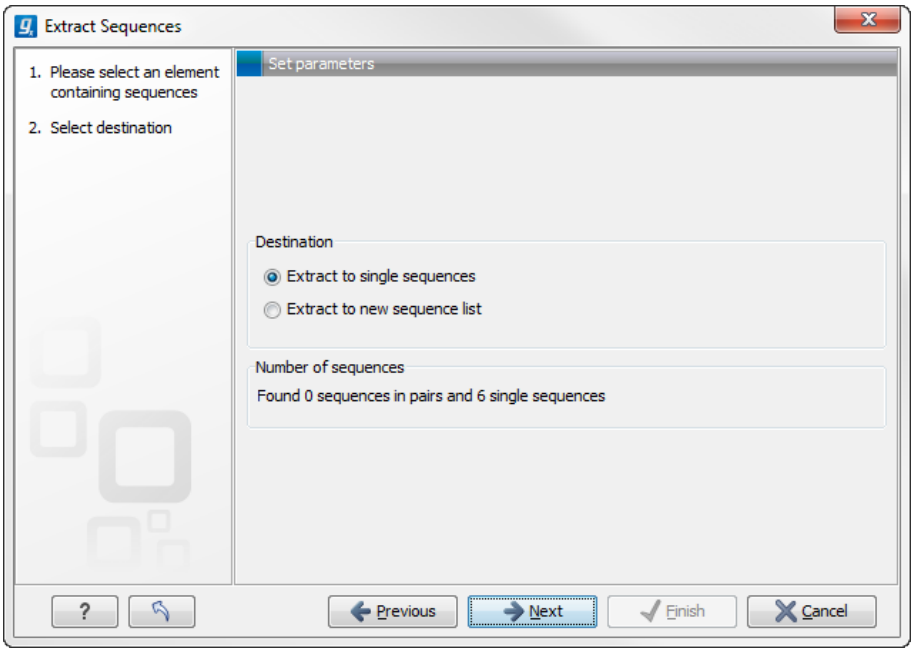



Figure 14.4: *Choosing whether the extracted sequences should be placed in a new list or as single sequences.*

Below these options, in the dialog, you can see the number of sequences that will be extracted.

## 14.3 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 ( )**

This opens the dialog displayed in figure 14.5:

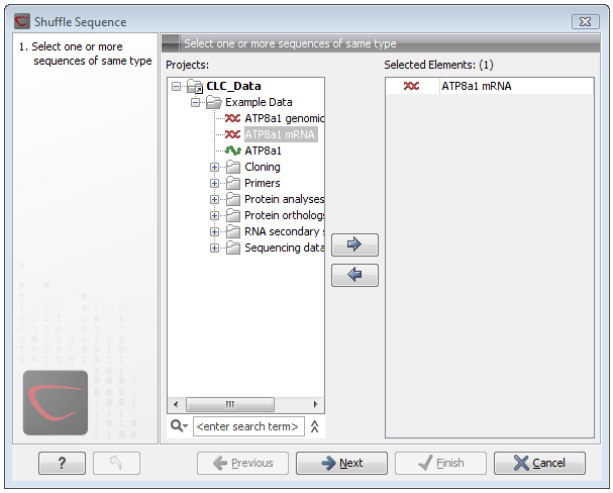



Figure 14.5: *Choosing sequence for shuffling.*

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 14.6:

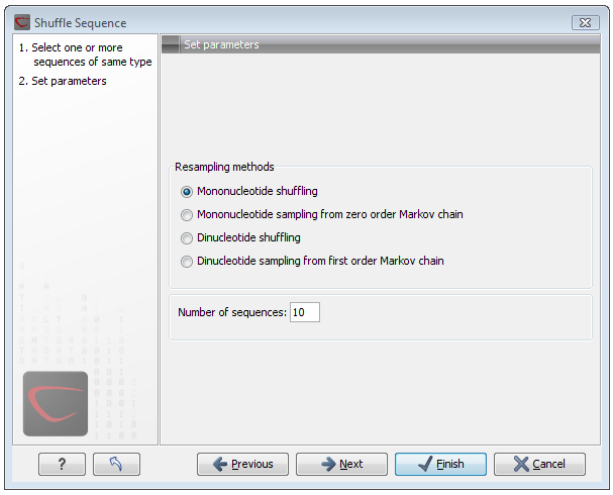



Figure 14.6: *Parameters for shuffling.*

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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

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.

## 14.4 Dot plots

Dot plots provide a powerful visual comparison of two sequences. Dot plots can also be used to compare regions of similarity within a sequence. This chapter first describes how to create and second how to adjust the view of the plot.

### 14.4.1 Create dot plots

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.

To create a dot plot, go to:

> **Toolbox** | **Classical Sequence Analysis** ( ) | **General Sequence Analysis** ( )| **Create Dot Plot** ( )

This opens the dialog shown in figure 14.7.

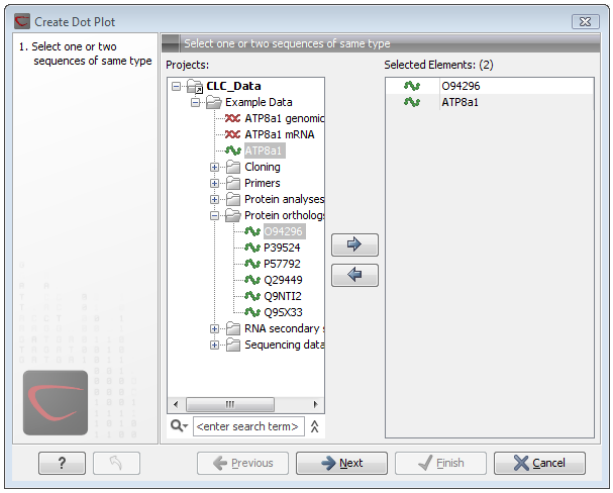



Figure 14.7: *Selecting sequences for the dot plot.*

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 elements from the selected elements. Click **Next** to adjust dot plot parameters. Clicking **Next** opens the dialog shown in figure 14.8.

**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.

#### Adjust 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'.

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

### 14.4.2 View dot plots

A view of a dot plot can be seen in figure 14.9. You can select **Zoom in** ( ) in the Toolbar and click the dot plot to zoom in to see the details of particular areas.

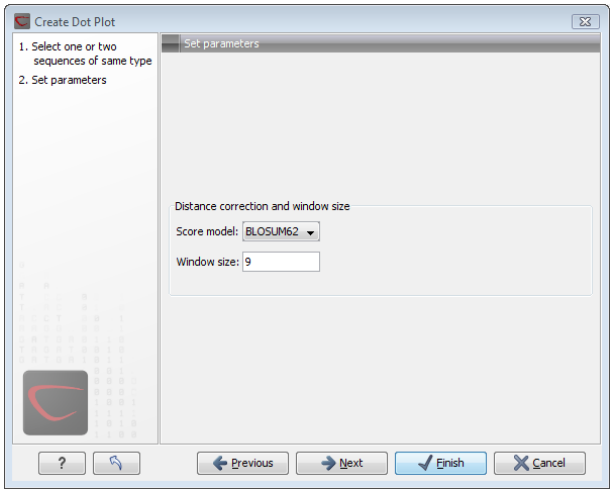



Figure 14.8: *Setting the dot plot parameters.*

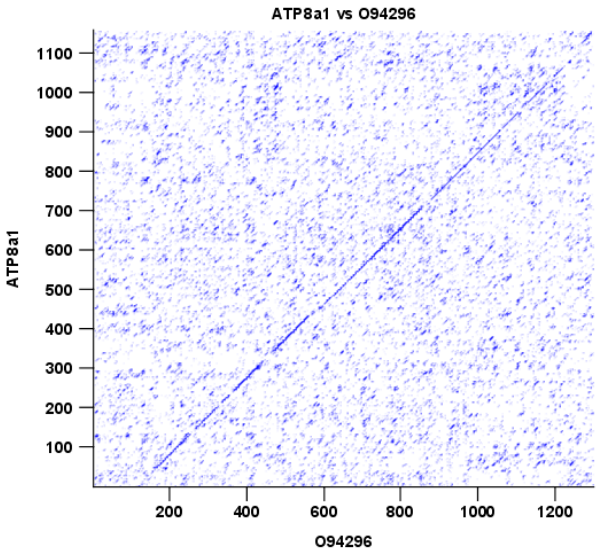



Figure 14.9: *A view is opened showing the dot plot.*

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. (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. (If you choose a color gradient, which includes white). Se figure 14.9.

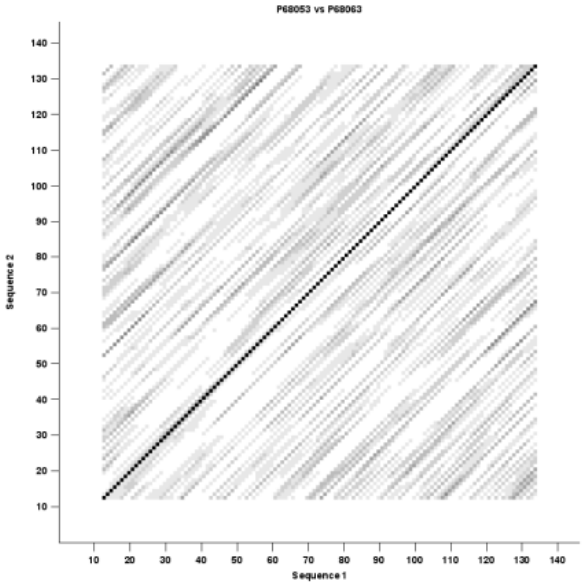



Figure 14.10: *Dot plot with inverted colors, practical for printing.*

### 14.4.3 Bioinformatics explained: Dot plots

#### Realization of 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 14.11 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`.

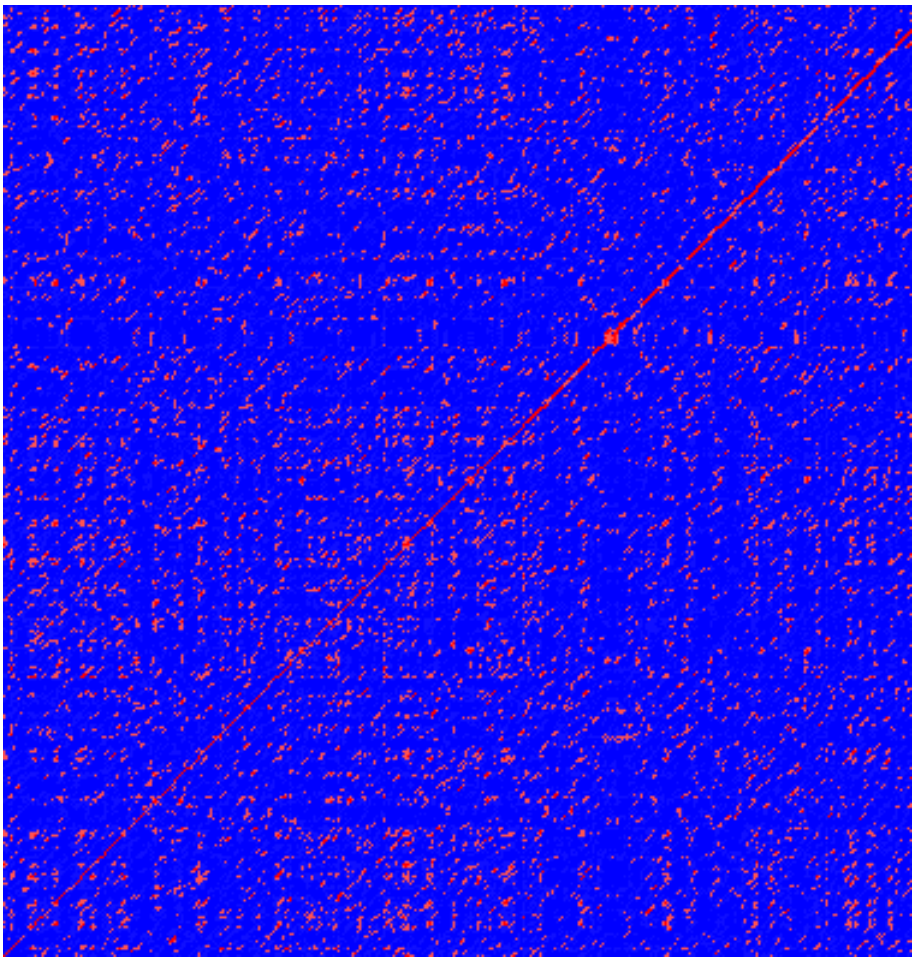


Figure 14.11: *Dot plot of DQ232610 vs. DQ023146 (Influenza A virus nucleoproteins) showing and overall similarity*

**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.

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 14.13 you can see a sequence with repeats.

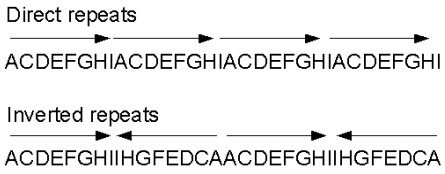



Figure 14.12: *Direct and inverted repeats shown on an amino acid sequence generated for demonstration purposes.*

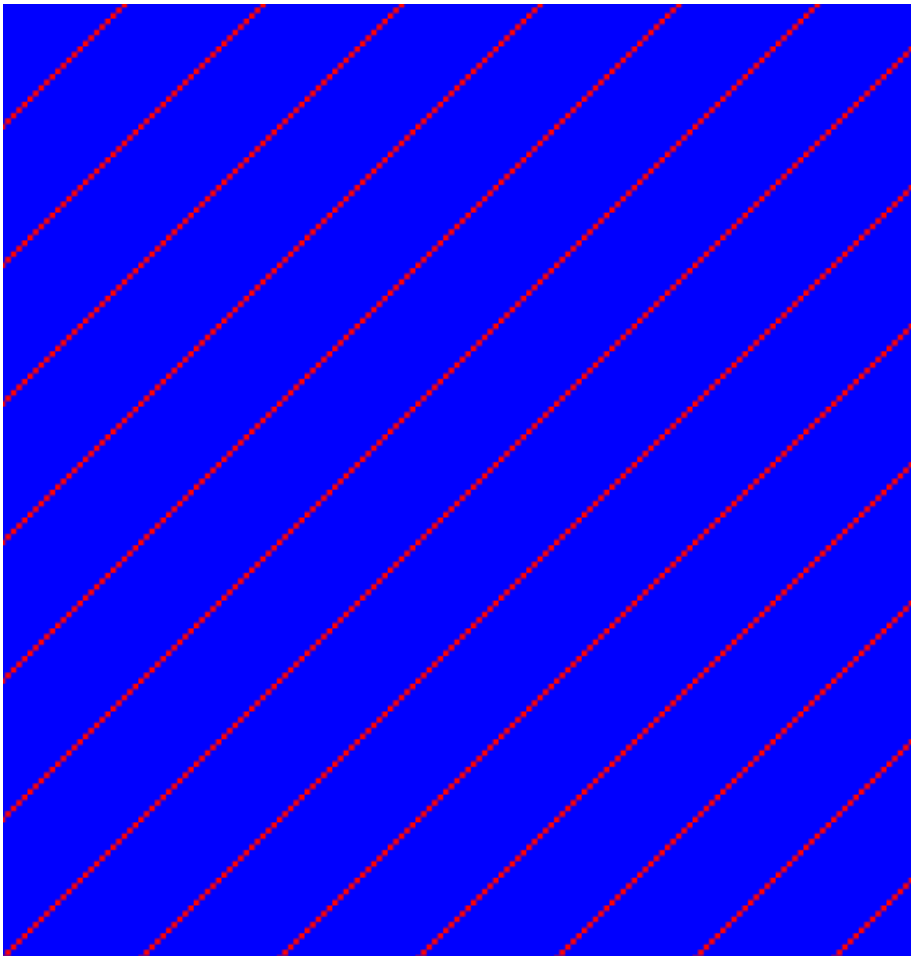


Figure 14.13: *The dot plot of a sequence showing repeated elements. See also figure 14.12.*

### 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 14.14. 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)

### Sequence inversions

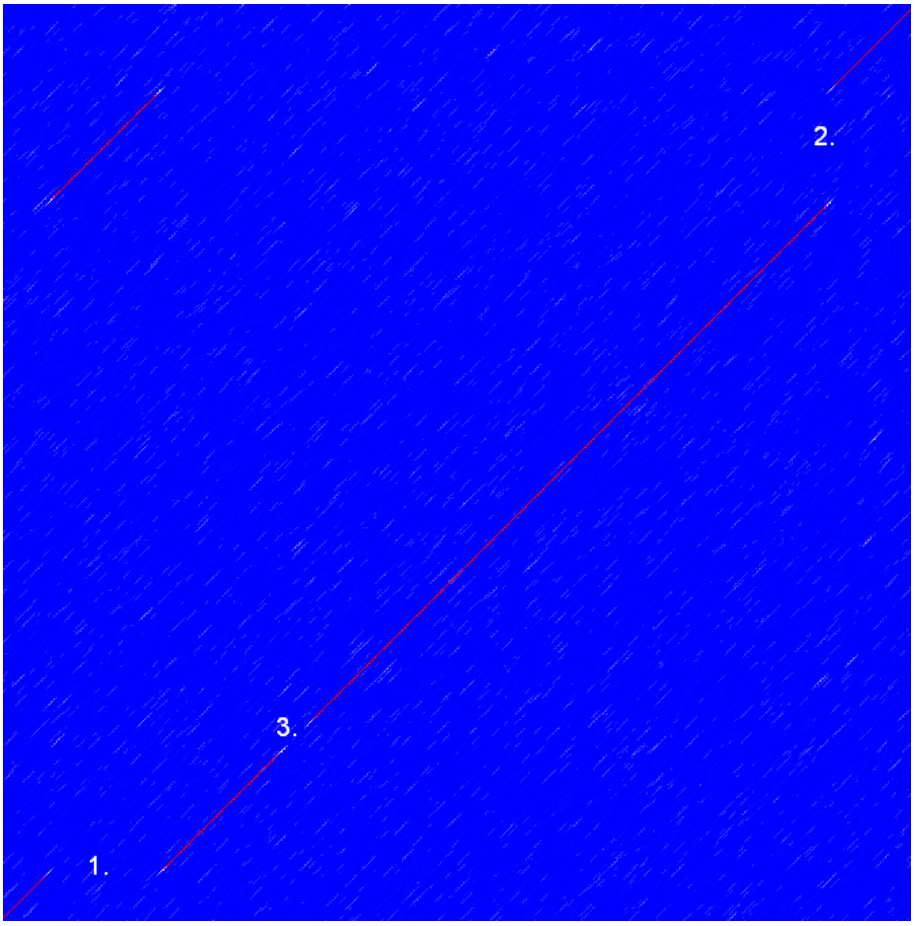



Figure 14.14: *This dot plot show various frame shifts in the sequence. See text for details.*

In dot plots you can see an inversion of sequence as contrary diagonal to the diagonal showing similarity. In figure 14.15 you can see a dot plot (window length is 3) with an inversion.

#### 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 14.16 is a square shows the low-complexity region of this sequence.

#### Creative Commons License

All CLC bio's scientific articles are licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 2.5 License. You are free to copy, distribute, display, and use the work for educational purposes, under the following conditions: You must attribute the work in its original form and "CLC bio" has to be clearly labeled as author and provider of the work. You may not use this work for commercial purposes. You may not alter, transform, nor build upon this work.

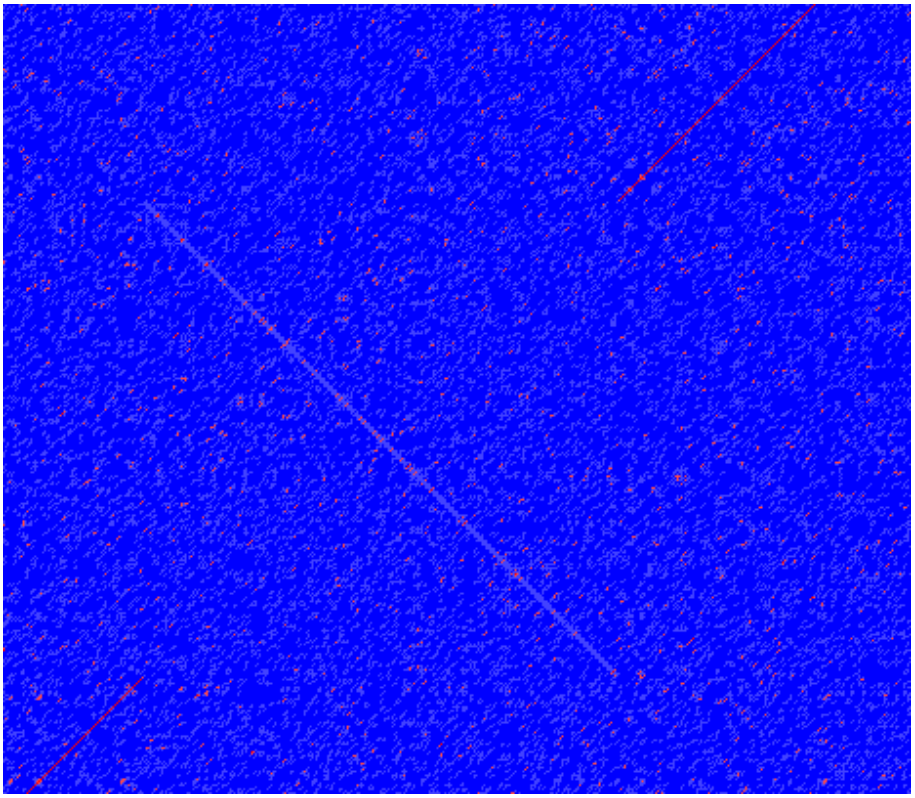


Figure 14.15: *The dot plot showing an inversion in a sequence. See also figure 14.12.*

SOME RIGHTS RESERVED

See http://creativecommons.org/licenses/by-nc-nd/2.5/ for more information on how to use the contents.

### 14.4.4 Bioinformatics explained: Scoring matrices

Biological sequences have evolved throughout time and evolution has shown that not all changes to a biological sequence is equally likely to happen. Certain amino acid substitutions (change of one amino acid to another) happen often, whereas other substitutions are very rare. For instance, tryptophan (W) which is a relatively rare amino acid, will only — on very rare occasions — mutate into a leucine (L).

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 14.1. 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].

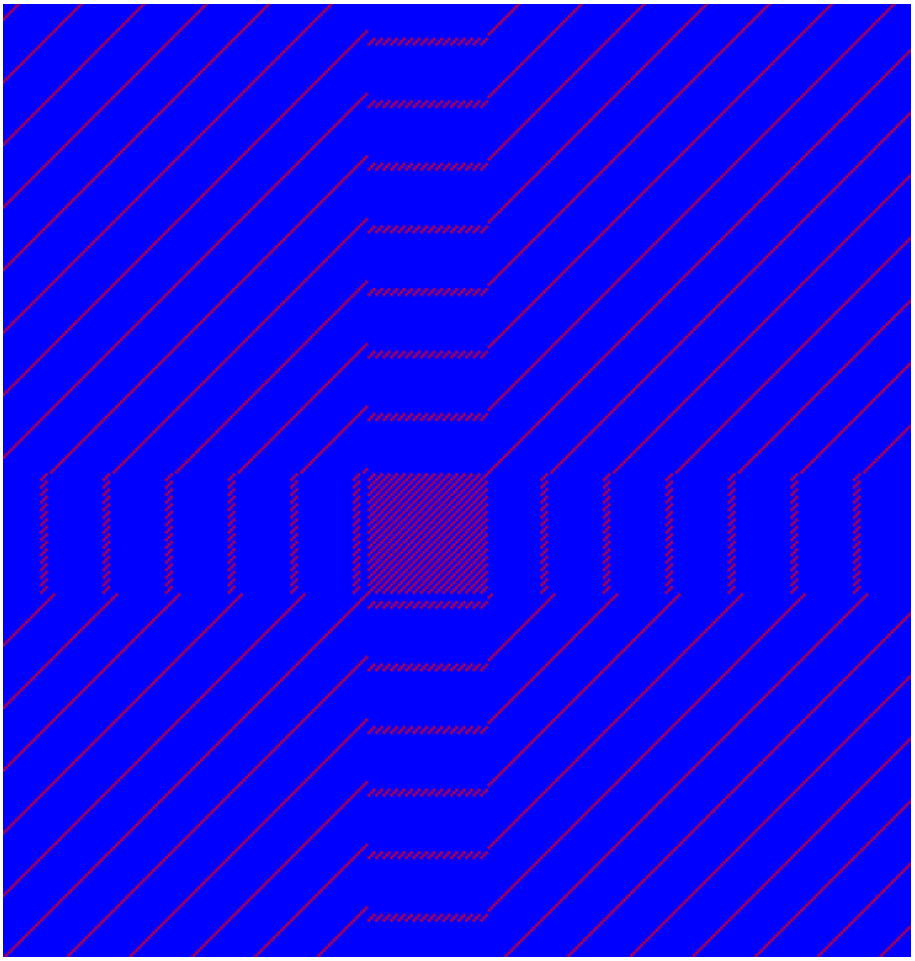


Figure 14.16: *The dot plot showing a low-complexity region in the sequence. The sequence is artificial and low complexity regions do not always show as a square.*

### 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 14.17).

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 | 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 14.1: **The BLOSUM62 matrix**. A tabular view of the BLOSUM62 matrix containing all possible substitution scores [Henikoff and Henikoff, 1992].

a scoring matrixï¿½called 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 14.17) 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.

#### Other useful resources

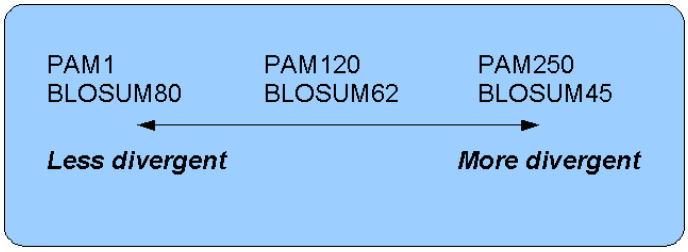



Figure 14.17: *Relationship between scoring matrices. The BLOSUM62 has become a* de facto *standard scoring matrix for a wide range of alignment programs. It is the default matrix in BLAST.*

BLOKS database
http://blocks.fhcrc.org/

NCBI help site
http://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs

**Creative Commons License**

All CLC bio's scientific articles are licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 2.5 License. You are free to copy, distribute, display, and use the work for educational purposes, under the following conditions: You must attribute the work in its original form and "CLC bio" has to be clearly labeled as author and provider of the work. You may not use this work for commercial purposes. You may not alter, transform, nor build upon this work.




See http://creativecommons.org/licenses/by-nc-nd/2.5/ for more information on how to use the contents.

## 14.5 Local complexity plot

In *CLC Genomics Workbench* it is possible to calculate local complexity for both DNA and protein sequences. The local complexity is a measure of the diversity in the composition of amino acids within a given range (window) of the sequence. The K2 algorithm is used for calculating local complexity [Wootton and Federhen, 1993]. To conduct a complexity calculation do the following:

> **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 14.18 shows an example of a local complexity plot.

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**. The values of the complexity plot approaches 1.0 as the distribution of amino acids become

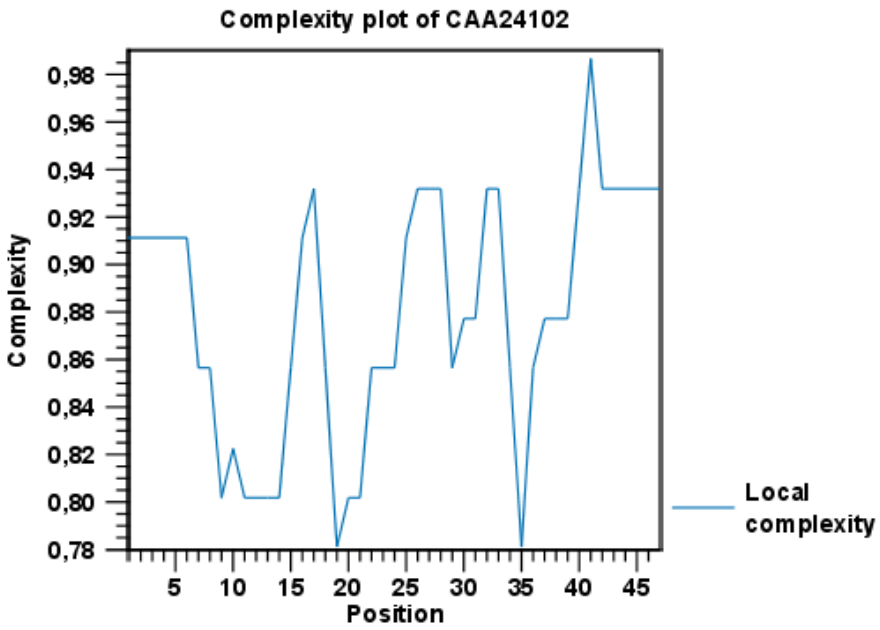



Figure 14.18: *An example of a local complexity plot.*

more complex.

See section C in the appendix for information about the graph view.

## 14.6 Sequence statistics

*CLC Genomics Workbench* can produce an output with many relevant statistics for protein sequences. Some of the statistics are also relevant to produce for DNA sequences. Therefore, this section deals with both types of statistics. The required steps for producing the statistics are the same.

To create a statistic for the sequence, do the following:

> **Toolbox** | **Classical Sequence Analysis** ( ) | **General Sequence Analysis** ( )| **Create Sequence Statistics** ( )

This opens a dialog where you can alter your choice of sequences. If you had already selected sequences in the Navigation Area, these will be shown in the **Selected Elements** window. However you can remove these, or add others, by using the arrows to move sequences in or out of the **Selected Elements** window. You can also add sequence lists.

**Note!** You cannot create statistics for DNA and protein sequences at the same time; they must be run separately.

When the sequences are selected, click **Next**.

This opens the dialog displayed in figure 14.19.

The dialog offers to adjust the following parameters:

- **Individual statistics layout.** If more sequences were selected in **Step 1**, this function generates separate statistics for each sequence.

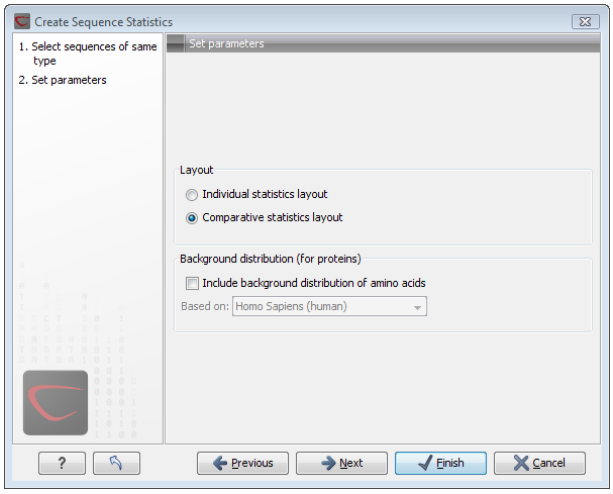



Figure 14.19: *Setting parameters for the sequence statistics.*

- **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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**. An example of protein sequence statistics is shown in figure 14.20.

**1 Protein statistics**

**1.1 Sequence information**

| | |
|---|---|
| Sequence type | Protein |
| Length | 147 |
| Organism | Mus musculus |
| Name | CAA32220 |
| Description | haemoglobin beta-h0 chain [Mus musculus]. |
| Modification Date | 18-APR-2005 |
| Weight | 16,412 kDa |

**1.2 Half-life**

| N-terminal aa | Half-life mammals | Half-life yeast | Half-life E.Coli |
|---|---|---|---|

Figure 14.20: *Example of protein sequence statistics.*

Nucleotide sequence statistics are generated using the same dialog as used for protein sequence statistics. However, the output of Nucleotide sequence statistics is less extensive than that of the protein sequence statistics.

**Note!** The headings of the tables change depending on whether you calculate 'individual' or 'comparative' sequence statistics.

The output of comparative protein sequence statistics include:

- 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
- Half-life
- Extinction coefficient
- Counts of Atoms
- Frequency of Atoms
- Count of hydrophobic and hydrophilic residues
- Frequencies of hydrophobic and hydrophilic residues
- Count of charged residues
- Frequencies of charged residues
- Amino acid distribution
- Histogram of amino acid distribution
- Annotation table
- Counts of di-peptides
- Frequency of di-peptides

The output of nucleotide sequence statistics include:

- General statistics:
  - 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 for linear sequences and sequence length for circular molecules. The `units` are monophosphates. Both the weight for single- and double stranded molecules are includes. The atomic composition is defined the same way.

- Atomic composition
- Nucleotide distribution table
- Nucleotide distribution histogram
- Annotation table
- Counts of di-nucleotides
- Frequency of di-nucleotides

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 14.6.1.

### 14.6.1 Bioinformatics explained: Protein statistics

Every protein holds specific and individual features which are unique to that particular protein. Features such as isoelectric point or amino acid composition can reveal important information of a novel protein. Many of the features described below are calculated in a simple way.

#### Molecular weight

The molecular weight is the mass of a protein or molecule. The molecular weight is simply calculated as the sum of the atomic mass of all the atoms in the molecule.

The weight of a protein is usually represented in Daltons (Da).

A calculation of the molecular weight of a protein does not usually include additional posttranslational 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.

| 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 14.2: **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*.

### 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].

### 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 14.2). 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.

### Creative Commons License

All CLC bio's scientific articles are licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 2.5 License. You are free to copy, distribute, display, and use the work for educational purposes, under the following conditions: You must attribute the work in its original form and "CLC bio" has to be clearly labeled as author and provider of the work. You may not use this work for commercial purposes. You may not alter, transform, nor build upon this work.




See http://creativecommons.org/licenses/by-nc-nd/2.5/ for more information on how to use the contents.

## 14.7 Join sequences

*CLC Genomics Workbench* can join several nucleotide or protein sequences into one sequence. This feature can for example be used to construct "supergenes" for phylogenetic inference by joining several disjoint genes into one. Note, that when sequences are joined, all their annotations are carried over to the new spliced sequence.

Two (or more) sequences can be joined by:

**Toolbox** | **General Sequence Analyses** | **Join sequences** ( )

This opens the dialog shown in figure 14.21.

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 14.22.

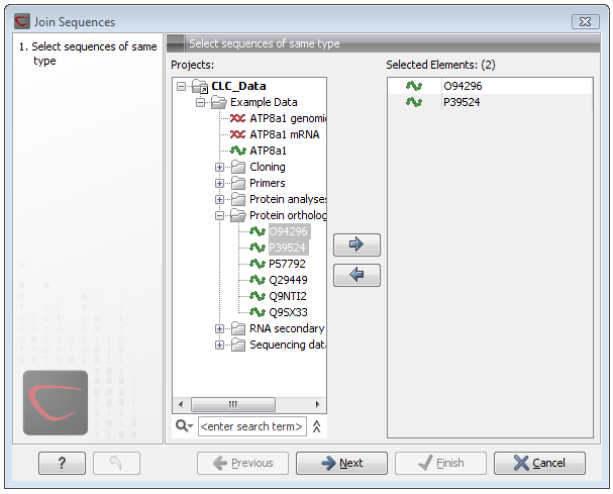



Figure 14.21: *Selecting two sequences to be joined.*

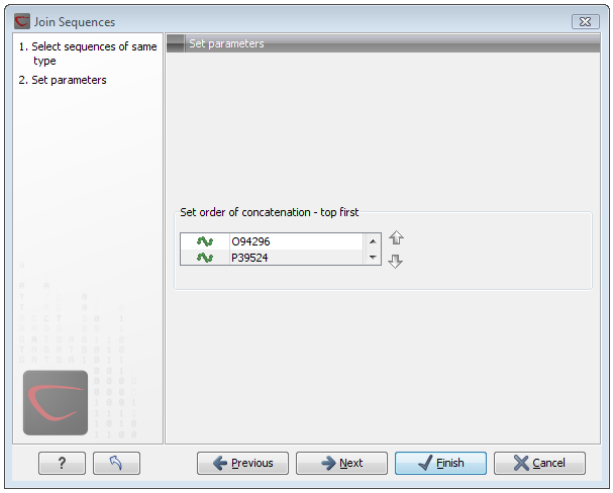



Figure 14.22: *Setting the order in which sequences are joined.*

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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

The result is shown in figure 14.23.

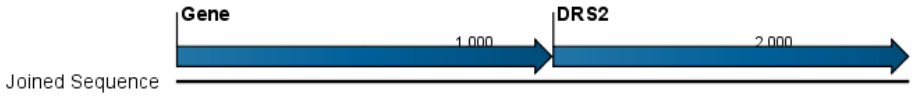



Figure 14.23: *The result of joining sequences is a new sequence containing the annotations of the joined sequences (they each had a HBB annotation).*

## 14.8 Pattern discovery

With *CLC Genomics Workbench* you can perform pattern discovery on both DNA and protein sequences. Advanced hidden Markov models can help to identify unknown sequence patterns across single or even multiple sequences.

In order to search for unknown patterns:

**Toolbox | Classical Sequence Analysis ( ) | General Sequence Analysis ( )| Pattern Discovery ( )**

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 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 14.24).

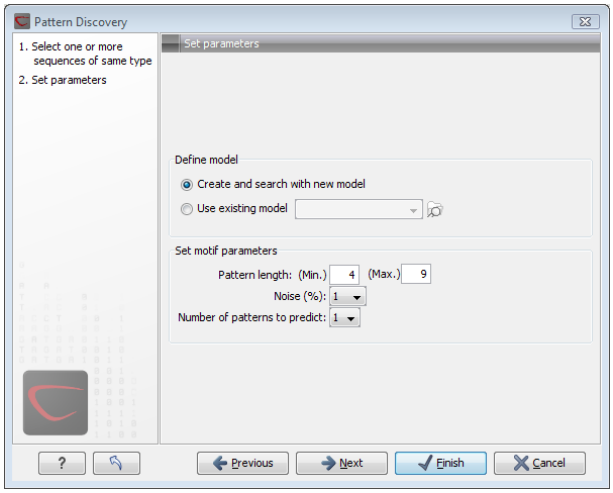



Figure 14.24: *Setting parameters for the pattern discovery. See text for details.*

In order to search unknown sequences with an already existing model:

Select to use an already existing model which is seen in figure 14.24. Models are represented with the following icon in the **Navigation Area** ( ).

### 14.8.1 Pattern discovery search parameters

Various parameters can be set prior to the pattern discovery. The parameters are listed below and a screenshot of the parameter settings can be seen in figure 14.24.

- **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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**. This will open a view showing the patterns found as annotations on the original sequence (see figure 14.25). If you have selected several sequences, a corresponding number of views will be opened.




Figure 14.25: *Sequence view displaying two discovered patterns.*

### 14.8.2 Pattern search output

If the analysis is performed on several sequences at a time the method will search for patterns in the sequences and open a new view for each of the sequences, in which a pattern was discovered. Each novel pattern will be represented as an annotation of the type **Region**. More information on each found pattern is available through the tool-tip, including detailed information on the position of the pattern and quality scores.

It is also possible to get a tabular view of all found patterns in one combined table. Then each found pattern will be represented with various information on obtained scores, quality of the pattern and position in the sequence.

A table view of emission values of the actual used HMM model is presented in a table view. This model can be saved and used to search for a similar pattern in new or unknown sequences.

## 14.9 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 19.3.1) . 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.

### 14.9.1 Dynamic motifs

In the **Side Panel** of sequence views, there is a group called **Motifs** (see figure 14.26).

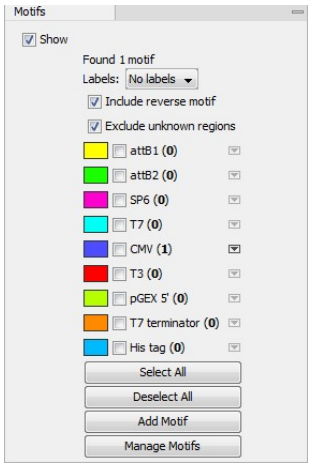



Figure 14.26: *Dynamic motifs in the Side Panel.*

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 14.27.

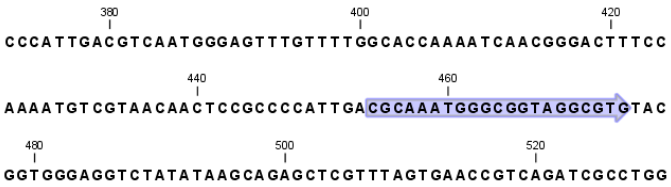



Figure 14.27: *Showing dynamic motifs on the sequence.*

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 14.28.

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:

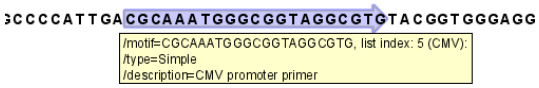



Figure 14.28: *Showing dynamic motifs on the sequence.*

- **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 14.26 is a pre-defined list that is included with the *CLC Genomics Workbench*. You can define your own set of motifs to use instead. In order to do this, you can either click on the **Add Motif** button in the side panel (see figure 14.26) and directly define and add motifs of choice as illustrated in figure 14.32. Alternatively, you can create and save a **Motif list** ( ) (see section 14.10). Subsequently, in the sequence view click the **Manage Motifs** button in the side panel which will bring up the dialog shown in figure 14.29.

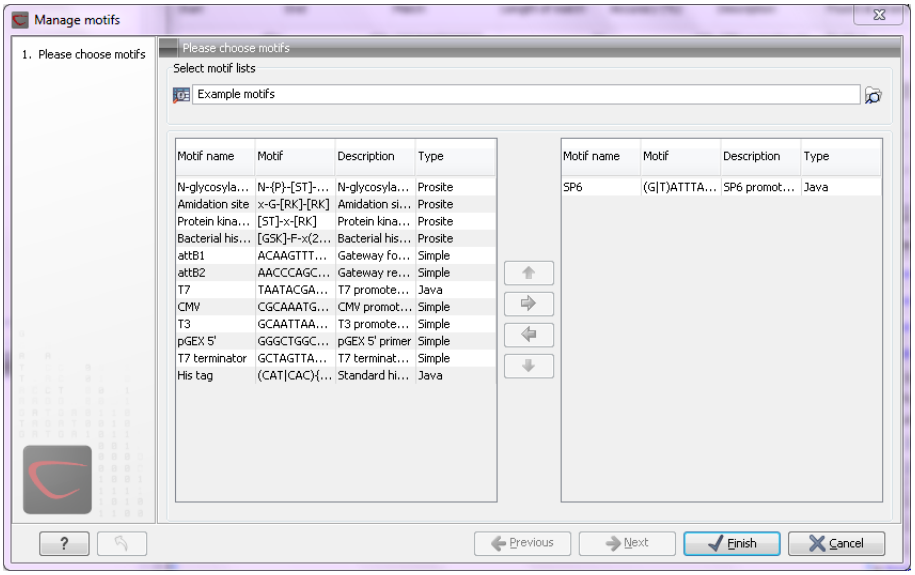



Figure 14.29: *Managing the motifs to be shown.*

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**.

### 14.9.2 Motif search from the Toolbox

The dynamic motifs described in section 14.9.1 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 14.30).

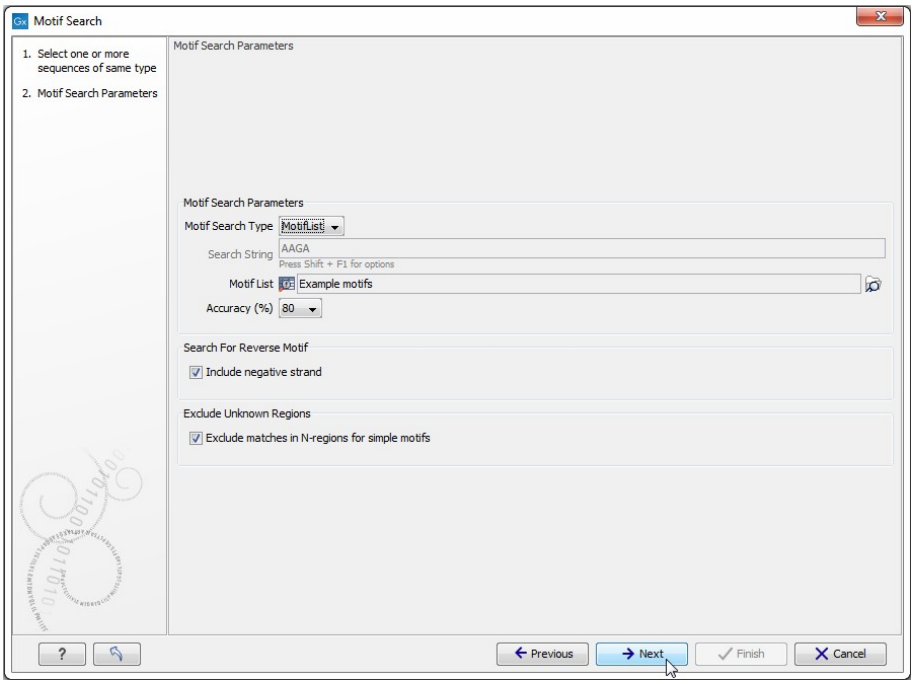



Figure 14.30: *Setting parameters for the motif search.*

The options for the motif search are:

- **Motif types.** Choose what kind of motif to be used:
  - Simple motif. Choosing this option means that you enter a simple motif, e.g. ATGATGNNATG.
  - Java regular expression. See section 14.9.3.
  - Prosite regular expression. For proteins, you can enter different protein patterns from the PROSITE database (protein patterns using regular expressions and describing specific amino acid sequences). The PROSITE database contains a great number of patterns and have been used to identify related proteins (see `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 14.10).

- **Motif.** If you choose to search with a simple motif, you should enter a literal string as your motif. Ambiguous amino acids and nucleotides are allowed. Example; ATGATGNNATG. If your motif type is Java regular expression, you should enter a regular expression according to the syntax rules described in section 14.9.3. Press **Shift + F1** key for options. For proteins, you can search with a Prosite regular expression and you should enter a protein pattern from the PROSITE database.

- **Accuracy.** If you search with a simple motif, you can adjust the accuracy of the motif to the match on the sequence. If you type in a simple motif and let the accuracy be 80%, the motif search algorithm runs through the input sequence and finds all subsequences of the same length as the simple motif such that the fraction of identity between the subsequence and the simple motif is at least 80%. A motif match is added to the sequence as an annotation with the exact fraction of identity between the subsequence and the simple motif. If you use a list of motifs, the accuracy applies only to the simple motifs in the list.

- **Search for reverse motif.** This enables searching on the negative strand on nucleotide sequences.

- **Exclude unknown regions.** Genome sequence often have large regions with unknown sequence. These regions are very often padded with N's. Ticking this checkbox will not display hits found in N-regions.Motif search handles ambiguous characters in the way that two residues are different if they do not have any residues in common. For example: For nucleotides, *N* matches any character and *R* matches *A,G*. For proteins, *X* matches any character and *Z* matches *E,Q*.

Click **Next** to adjust how to handle the results and then click **Finish**. There are 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 14.31.

- **Create table**. This will create an overview table of all the motifs found for all the input sequences.

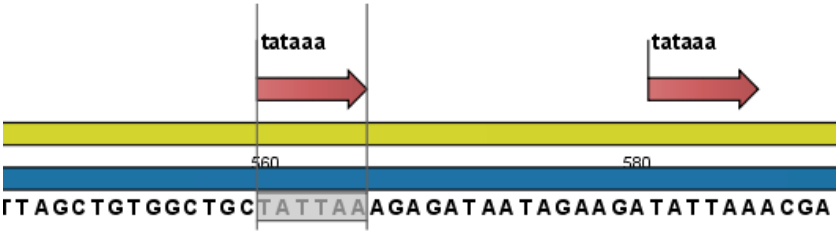



Figure 14.31: *Sequence view displaying the pattern found. The search string was 'tataaa'.*

### 14.9.3 Java regular expressions

A regular expressions is a string that describes or matches a set of strings, according to certain syntax rules. They are usually used to give a concise description of a set, without

having to list all elements. The simplest form of a regular expression is a literal string. The syntax used for the regular expressions is the Java regular expression syntax (see 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*.

## 14.10 Create motif list

*CLC Genomics Workbench* offers advanced and versatile options to create lists of sequence patterns or known motifs, represented either by a literal string or a regular expression.

A motif list can be created using one of two ways:

**Toolbox** | **Classical Sequence Analysis** ( ) | **General Sequence Analysis** ( )| **Create Motif List** ( )

**File** | **New** | **Motif List** ( )

**Add** ( ) button at the bottom of the view. This will open a dialog shown in figure 14.32.

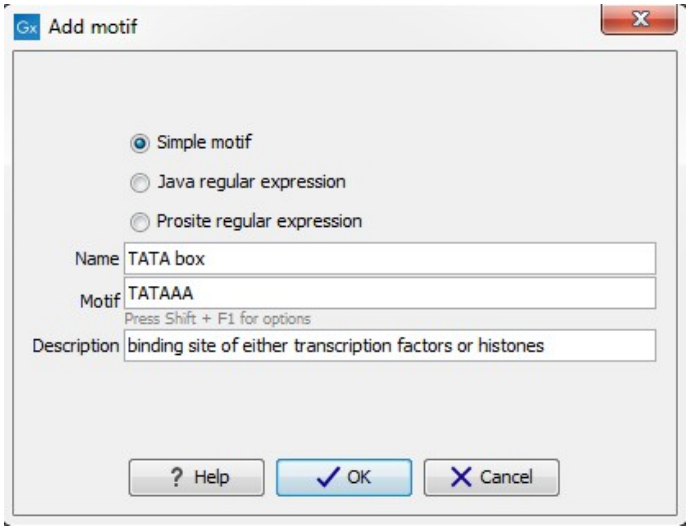



Figure 14.32: *Entering a new motif in the list.*

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 14.9.2 for more information about the syntax of motifs.
- **Description**. You can enter a description of the motif. In the result of a motif search, the description will appear in the result table and will be added as a note to the annotation on the sequence (visible in the **Annotation table** ( ) or by placing the mouse cursor on the annotation).
- **Type**. You can enter three different types of motifs: Simple motifs, java regular expressions or PROSITE regular expression. Read more in section 14.9.2.

The motif list can contain a mix of different types of motifs. This is practical because some motifs can be described with the simple syntax, whereas others need the more advanced regular expression syntax.

Instead of manually adding motifs, you can **Import From Fasta File** ( ). This will show a dialog where you can select a fasta file on your computer and use this to create motifs. This will

automatically take the name, description and sequence information from the fasta file, and put it into the motif list. The motif type will be "simple".

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 14.9).

# Chapter 15

# Nucleotide analyses

**Contents**


**15.1 Convert DNA to RNA . . . . . 302**
**15.2 Convert RNA to DNA . . . . . 303**
**15.3 Reverse complements of sequences . . . . . 304**
**15.4 Reverse sequence . . . . . 304**
**15.5 Translation of DNA or RNA to protein . . . . . 305**
15.5.1 Translate part of a nucleotide sequence . . . . . 307
**15.6 Find open reading frames . . . . . 307**
15.6.1 Open reading frame parameters . . . . . 308


*CLC Genomics Workbench* offers different kinds of sequence analyses, which only apply to DNA and RNA.

## 15.1 Convert DNA to RNA

*CLC Genomics Workbench* lets you convert a DNA sequence into RNA, substituting the T residues (Thymine) for U residues (Urasil):

**Toolbox** | **Classical Sequence Analysis** ( ) | **Nucleotide Analysis** ( )| **Convert DNA to RNA** ( )

This opens the dialog displayed in figure 15.1:

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** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

**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.

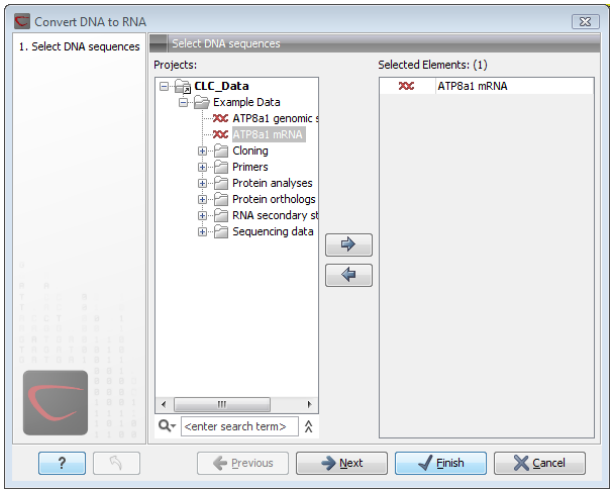



Figure 15.1: *Translating DNA to RNA.*

## 15.2 Convert RNA to DNA

*CLC Genomics Workbench* lets you convert an RNA sequence into DNA, substituting the U residues (Urasil) for T residues (Thymine):

**Toolbox** | **Classical Sequence Analysis** ( ) | **Nucleotide Analysis** ( )| **Convert RNA to DNA** ( )

This opens the dialog displayed in figure 15.2:

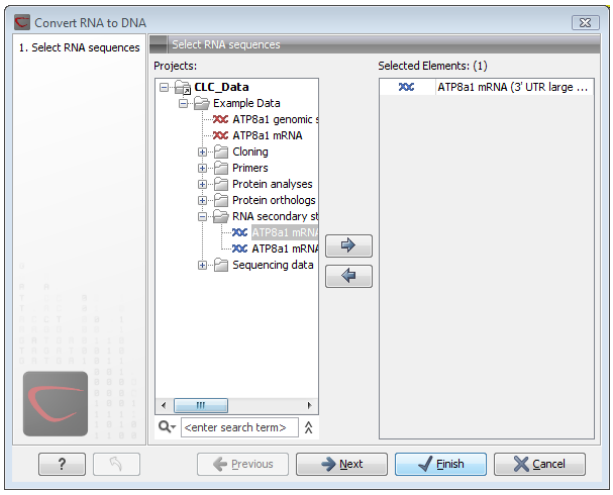



Figure 15.2: *Translating RNA to DNA.*

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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

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.

## 15.3 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 ( )**

This opens the dialog displayed in figure 15.3:

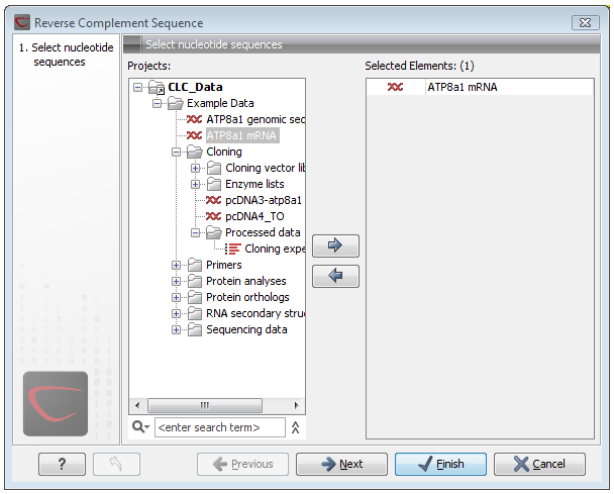



Figure 15.3: *Creating a reverse complement sequence.*

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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

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.

## 15.4 Reverse sequence

*CLC Genomics Workbench* is able to create the reverse of a nucleotide sequence.

**Note!** This is not the same as a reverse complement. If you wish to create the reverse complement, please refer to section 15.3.

To run the tool, go to:

> **Toolbox** | **Classical Sequence Analysis** ( ) | **Nucleotide Analysis** ( )| **Reverse Sequence** ( )

This opens the dialog displayed in figure 15.4:

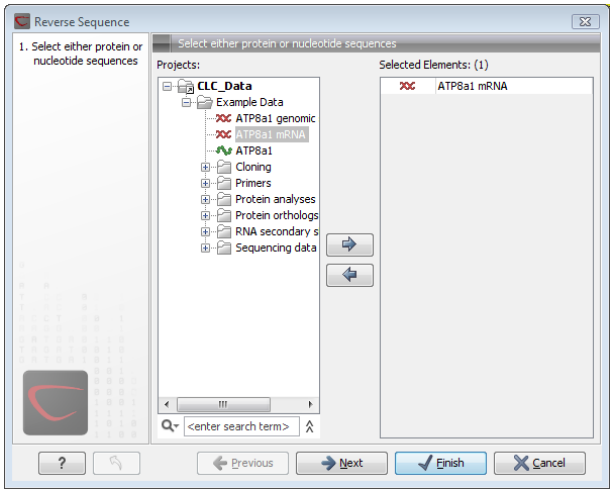



Figure 15.4: *Reversing a sequence.*

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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

**Note!** This is not the same as a reverse complement. If you wish to create the reverse complement, please refer to section 15.3.

## 15.5 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 15.5:

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 15.6:

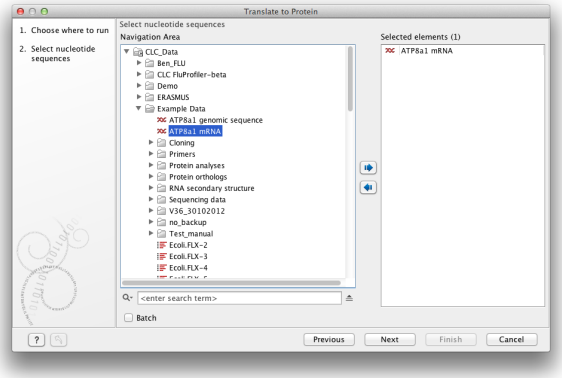



Figure 15.5: *Choosing sequences for translation.*

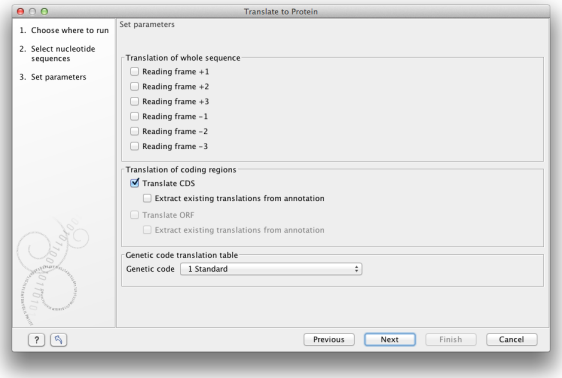



Figure 15.6: *Choosing translation of CDS's using standard translation table.*

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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**. 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.

### 15.5.1 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 and ORF" in the translation dialog (see figure 15.6).

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 (  )** | **choose a translation table** | **OK**

If the annotation contains information about the translation, this information will be used, and you do not have to specify a translation table.

The CDS and ORF annotations are colored yellow as default.

## 15.6 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 15.7:

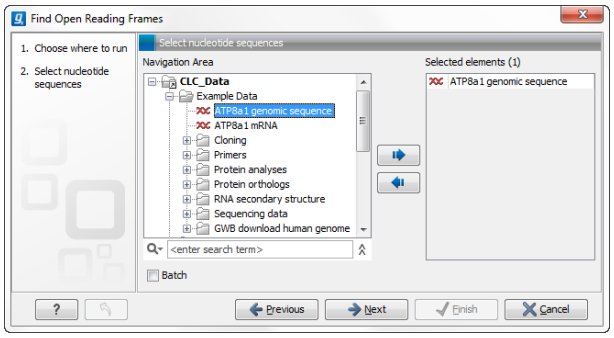



Figure 15.7: *Create Reading Frame dialog.*

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**.

### 15.6.1 Open reading frame parameters

This opens the dialog displayed in figure 15.8:

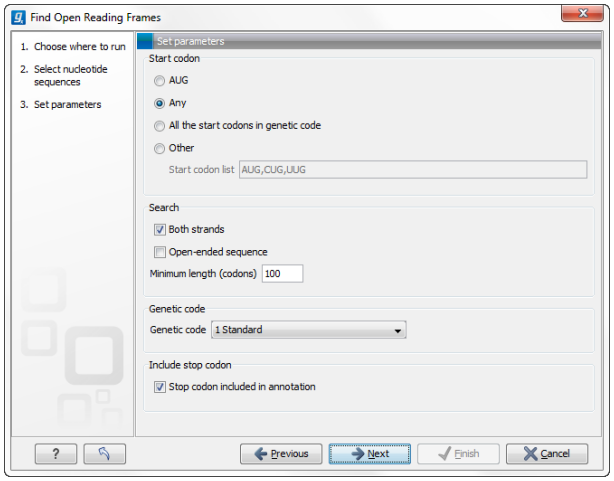



Figure 15.8: *Create Reading Frame dialog.*

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 15.9).

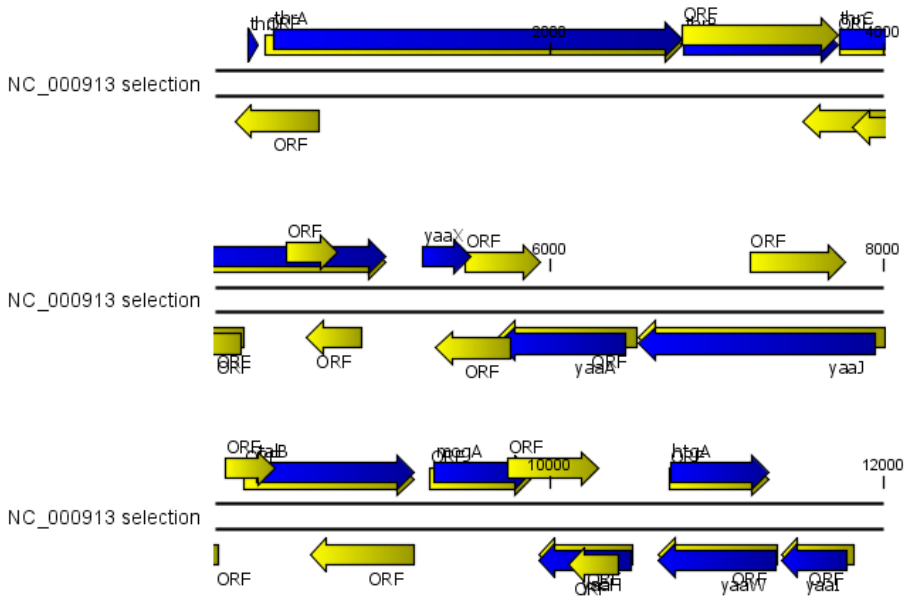



Figure 15.9: *The first 12,000 positions of the* E. coli *sequence NC_000913 downloaded from GenBank. The blue (dark) annotations are the genes while the yellow (brighter) annotations are the ORFs with a length of at least 100 amino acids. On the positive strand around position 11,000, a gene starts 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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

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 16

# Protein analyses

## Contents


**16.1 Signal peptide prediction . . . . . 311**
16.1.1 Signal peptide prediction parameter settings . . . . . 311
16.1.2 Signal peptide prediction output . . . . . 312
16.1.3 Bioinformatics explained: Prediction of signal peptides . . . . . 312
**16.2 Protein charge . . . . . 317**
16.2.1 Modifying the layout . . . . . 318
**16.3 Transmembrane helix prediction . . . . . 318**
**16.4 Antigenicity . . . . . 319**
16.4.1 Plot of antigenicity . . . . . 320
16.4.2 Antigenicity graphs along sequence . . . . . 321
**16.5 Hydrophobicity . . . . . 321**
16.5.1 Hydrophobicity plot . . . . . 321
16.5.2 Hydrophobicity graphs along sequence . . . . . 322
16.5.3 Bioinformatics explained: Protein hydrophobicity . . . . . 323
**16.6 Pfam domain search . . . . . 326**
16.6.1 Download of Pfam database . . . . . 326
16.6.2 Running Pfam Domain Search . . . . . 327
**16.7 Secondary structure prediction . . . . . 328**
**16.8 Protein report . . . . . 329**
16.8.1 Protein report output . . . . . 331
**16.9 Reverse translation from protein into DNA . . . . . 332**
16.9.1 Reverse translation parameters . . . . . 332
16.9.2 Bioinformatics explained: Reverse translation . . . . . 333
**16.10 Proteolytic cleavage detection . . . . . 335**
16.10.1 Proteolytic cleavage parameters . . . . . 336
16.10.2 Bioinformatics explained: Proteolytic cleavage . . . . . 338


*CLC Genomics Workbench* offers a number of analyses of proteins as described in this chapter.

## 16.1 Signal peptide prediction

Signal peptides target proteins to the extracellular environment either through direct plasmamembrane translocation in prokaryotes or is routed through the Endoplasmatic Reticulum in eukaryotic cells. The signal peptide is removed from the resulting mature protein during translocation across the membrane. For prediction of signal peptides, we query SignalP [Nielsen et al., 1997, Bendtsen et al., 2004b] located at `http://www.cbs.dtu.dk/services/SignalP/`. Thus an active internet connection is required to run the signal peptide prediction. Additional information on SignalP and Center for Biological Sequence analysis (CBS) can be found at `http://www.cbs.dtu.dk` and in the original research papers [Nielsen et al., 1997, Bendtsen et al., 2004b].

In order to predict potential signal peptides of proteins, the D-score from the SignalP output is used for discrimination of signal peptide versus non-signal peptide (see section 16.1.3). This score has been shown to be the most accurate [Klee and Ellis, 2005] in an evaluation study of signal peptide predictors.

In order to use SignalP, *you need to download the SignalP plugin* using the plugin manager, see section 1.7.1.

When the plugin is downloaded and installed, you can use it to predict signal peptides:

> **Toolbox** | **Classical Sequence Analysis (  )** | **Protein Analysis (  )**| **Signal Peptide 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.

Click **Next** to set parameters for the SignalP analysis.

### 16.1.1 Signal peptide prediction parameter settings

You should select which organism group the input sequences belong to. the default is eukaryote (see figure 16.1).

- Eukaryote (default)
- Gram-negative bacteria
- Gram-positive bacteria

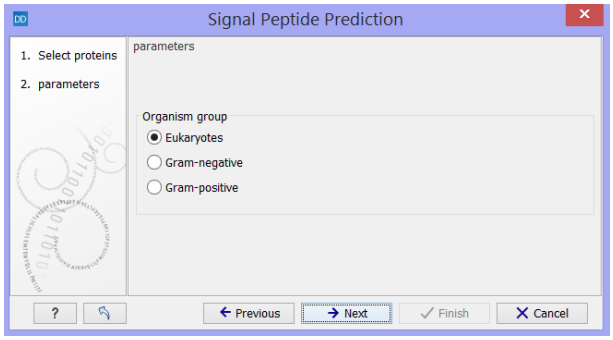



Figure 16.1: *Setting the parameters for signal peptide prediction.*

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 if a signal peptide is found. If no signal peptide is found in the sequence, a dialog box will be shown.

The predictions obtained can either be shown as annotations on the sequence, listed in a table or be shown as the detailed and full text output from the SignalP method. This can be used to interpret borderline predictions:

- Add annotations to sequence
- Create table
- Text

Click **Next** to adjust how to handle the results, then click **Finish**.

### 16.1.2 Signal peptide prediction output

After running the prediction as described above, the protein sequence will show predicted signal peptide as annotations on the original sequence (see figure 16.2). Make sure the Side Panel settings of the sequence is so that 'Show annotations' is checked in the 'Annotation layout' palette, and that the annotation type 'Signal peptide' is checked in the 'Annotation types' palette.

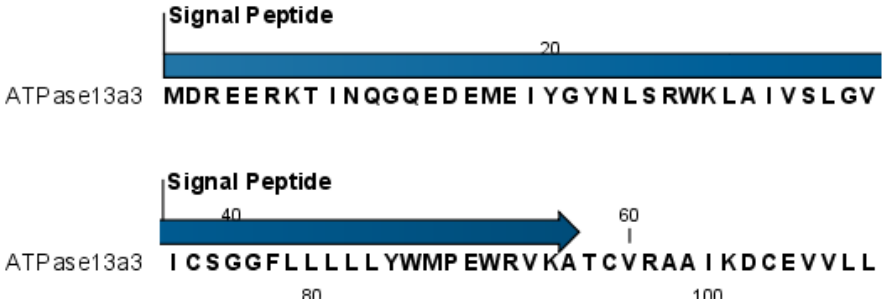



Figure 16.2: *N-terminal signal peptide shown as annotation on the sequence.*

Additional notes can be added through the **Edit annotation** ( ) right-click mouse menu. See section 10.3.2.

Undesired annotations can be removed through the **Delete Annotation ( )** right-click mouse menu. See section 10.3.4.

### 16.1.3 Bioinformatics explained: Prediction of signal peptides

#### Why the interest in signal peptides?

The importance of signal peptides was shown in 1999 when Günter Blobel received the Nobel Prize in physiology or medicine for his discovery that "proteins have intrinsic signals that govern their transport and localization in the cell" [Blobel, 2000]. He pointed out the importance of defined peptide motifs for targeting proteins to their site of function.

Performing a query to PubMed[1] reveals that thousands of papers have been published, regarding signal peptides, secretion and subcellular localization, including knowledge of using signal

[1] http://www.ncbi.nlm.nih.gov/entrez/

peptides as vehicles for chimeric proteins for biomedical and pharmaceutical industry. Many papers describe statistical or machine learning methods for prediction of signal peptides and prediction of subcellular localization in general. After the first published method for signal peptide prediction [von Heijne, 1986], more and more methods have surfaced, although not all methods have been made available publicly.

**Different types of signal peptides**

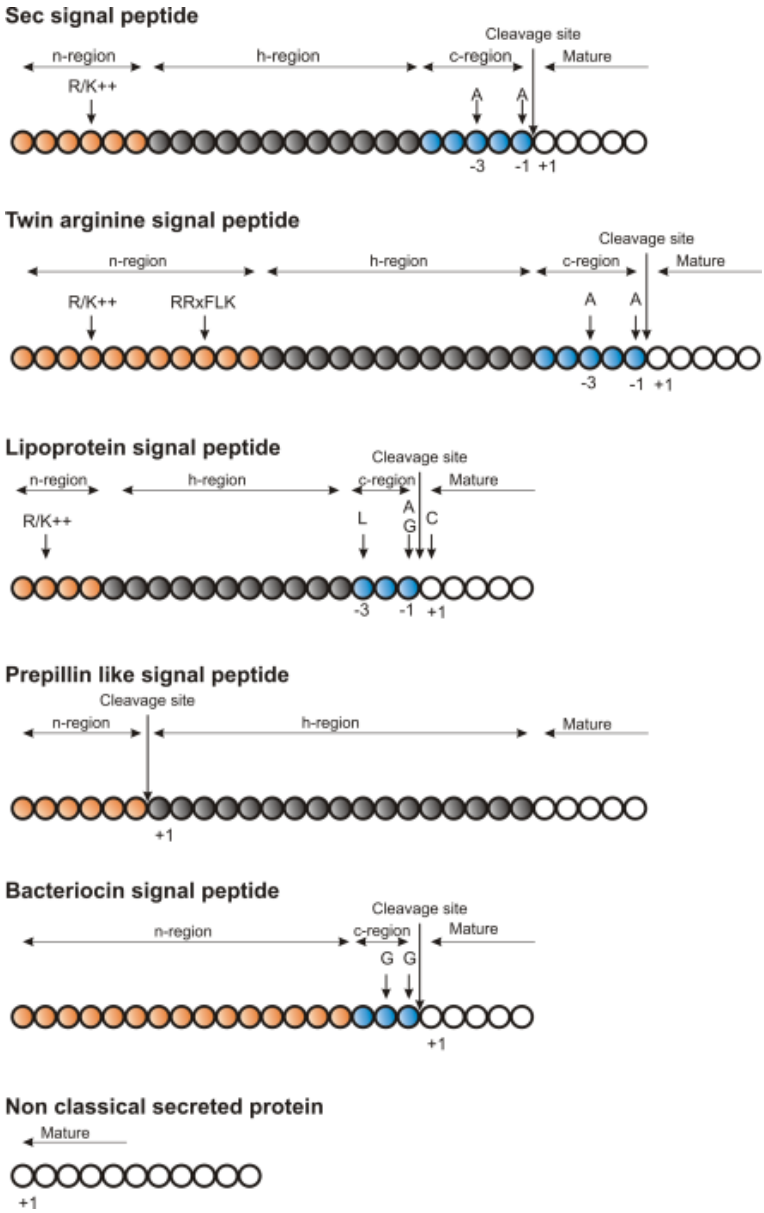



Figure 16.3: *Schematic representation of various signal peptides. Red color indicates n-region, gray color indicates h-region, cyan indicates c-region. All white circles are part of the mature protein.* $+1$ *indicates the first position of the mature protein. The length of the signal peptides is not drawn to scale.*

Soon after Günter Blobel's initial discovery of signal peptides, more targeting signals were found. Most cell types and organisms employ several ways of targeting proteins to the extracellular environment or subcellular locations. Most of the proteins targeted for the extracellular space or

subcellular locations carry specific sequence motifs (signal peptides) characterizing the type of secretion/targeting it undergoes.

Several new different signal peptides or targeting signals have been found during the later years, and papers often describe a small amino acid motif required for secretion of that particular protein. In most of the latter cases, the identified sequence motif is only found in this particular protein and as such cannot be described as a new group of signal peptides.

Describing the various types of signal peptides is beyond the scope of this text but several review papers on this topic can be found on PubMed. Targeting motifs can either be removed from, or retained in the mature protein after the protein has reached the correct and final destination. Some of the best characterized signal peptides are depicted in figure 16.3.

Numerous methods for prediction of protein targeting and signal peptides have been developed; some of them are mentioned and cited in the introduction of the SignalP research paper [Bendtsen et al., 2004b]. However, no prediction method will be able to cover all the different types of signal peptides. Most methods predicts classical signal peptides targeting to the general secretory pathway in bacteria or classical secretory pathway in eukaryotes. Furthermore, a few methods for prediction of non-classically secreted proteins have emerged [Bendtsen et al., 2004a, Bendtsen et al., 2005].

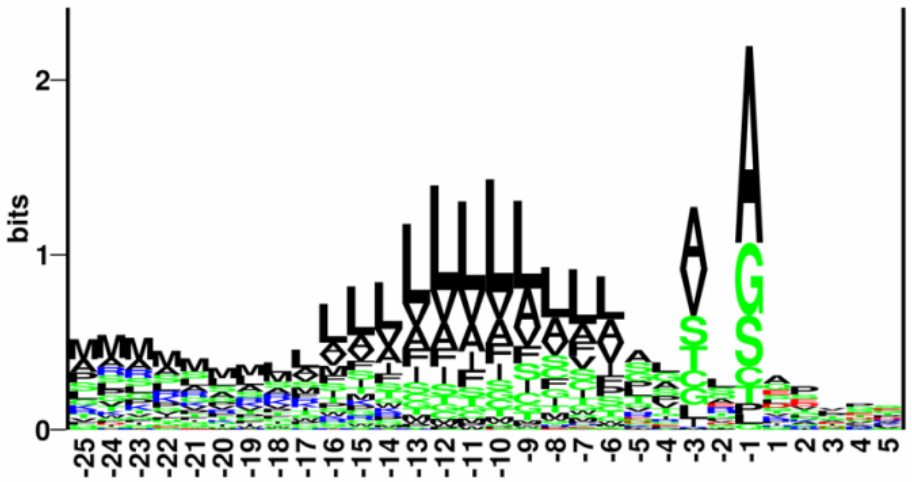



Figure 16.4: *Sequence logo of eukaryotic signal peptides, showing conservation of amino acids in bits [Schneider and Stephens, 1990]. Polar and hydrophobic residues are shown in green and black, respectively, while blue indicates positively charged residues and red negatively charged residues. The logo is based on an ungapped sequence alignment fixed at the -1 position of the signal peptides.*

### Prediction of signal peptides and subcellular localization

In the search for accurate prediction of signal peptides, many approaches have been investigated. Almost 20 years ago, the first method for prediction of classical signal peptides was published [von Heijne, 1986]. Nowadays, more sophisticated machine learning methods, such as neural networks, support vector machines, and hidden Markov models have arrived along with the increasing computational power and they all perform superior to the old weight matrix based methods [Menne et al., 2000]. Also, many other "classical" statistical approaches have been carried out, often in conjunction with machine learning methods. In the following sections, a wide

range of different signal peptide and subcellular prediction methods will be described.

Most signal peptide prediction methods require the presence of the correct N-terminal end of the preprotein for correct classification. As large scale genome sequencing projects sometimes assign the 5'-end of genes incorrectly, many proteins are annotated without the correct N-terminal [Reinhardt and Hubbard, 1998] leading to incorrect prediction of subcellular localization. These erroneous predictions can be ascribed directly to poor gene finding. Other methods for prediction of subcellular localization use information within the mature protein and therefore they are more robust to N-terminal truncation and gene finding errors.

### The SignalP method

One of the most cited and best methods for prediction of classical signal peptides is the SignalP method [Nielsen et al., 1997, Bendtsen et al., 2004b]. In contrast to other methods, SignalP also predicts the actual cleavage site; thus the peptide which is cleaved off during translocation over the membrane. Recently, an independent research paper has rated SignalP version 3.0 to be the best standalone tool for signal peptide prediction. It was shown that the D-score which is reported by the SignalP method is the best measure for discriminating secretory from non-secretory proteins [Klee and Ellis, 2005].

SignalP is located at `http://www.cbs.dtu.dk/services/SignalP/`

### What do the SignalP scores mean?

Many bioinformatics approaches or prediction tools do not give a yes/no answer. Often the user is facing an interpretation of the output, which can be either numerical or graphical. Why is that? In clear-cut examples there are no doubt; yes: this is a signal peptide! But, in borderline cases it is often convenient to have more information than just a yes/no answer. Here a graphical output can aid to interpret the correct answer. An example is shown in figure 16.5.

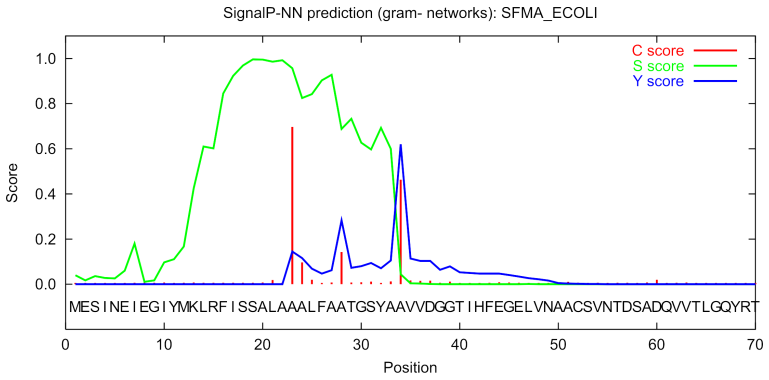



Figure 16.5: *Graphical output from the SignalP method of Swiss-Prot entry `SFMA_ECOLI`. Initially this seemed like a borderline prediction, but closer inspection of the sequence revealed an internal methionine at position 12, which could indicate a erroneously annotated start of the protein. Later this protein was re-annotated by Swiss-Prot to start at the M in position 12. See the text for description of the scores.*

The graphical output from SignalP (neural network) comprises three different scores, *C*, *S* and *Y*. Two additional scores are reported in the SignalP3-NN output, namely the *S-mean* and the *D-score*, but these are only reported as numerical values.

For each organism class in SignalP; Eukaryote, Gram-negative and Gram-positive, two different neural networks are used, one for predicting the actual signal peptide and one for predicting the position of the signal peptidase I (SPase I) cleavage site. The *S-score* for the signal peptide prediction is reported for every single amino acid position in the submitted sequence, with high scores indicating that the corresponding amino acid is part of a signal peptide, and low scores indicating that the amino acid is part of a mature protein.

The *C-score* is the "cleavage site" score. For each position in the submitted sequence, a C-score is reported, which should only be significantly high at the cleavage site. Confusion is often seen with the position numbering of the cleavage site. When a cleavage site position is referred to by a single number, the number indicates the first residue in the mature protein. This means that a reported cleavage site between amino acid 26-27 corresponds to the mature protein starting at (and include) position 27.

*Y-max* is a derivative of the C-score combined with the S-score resulting in a better cleavage site prediction than the raw C-score alone. This is due to the fact that multiple high-peaking C-scores can be found in one sequence, where only one is the true cleavage site. The cleavage site is assigned from the Y-score where the slope of the S-score is steep and a significant C-score is found.

The *S-mean* is the average of the S-score, ranging from the N-terminal amino acid to the amino acid assigned with the highest Y-max score, thus the S-mean score is calculated for the length of the predicted signal peptide. The S-mean score was in SignalP version 2.0 used as the criteria for discrimination of secretory and non-secretory proteins.

The *D-score* is introduced in SignalP version 3.0 and is a simple average of the S-mean and Y-max score. The score shows superior discrimination performance of secretory and non-secretory proteins to that of the S-mean score which was used in SignalP version 1 and 2.

For non-secretory proteins all the scores represented in the SignalP3-NN output should ideally be very low.

The hidden Markov model calculates the probability of whether the submitted sequence contains a signal peptide or not. The eukaryotic HMM model also reports the probability of a signal anchor, previously named uncleaved signal peptides. Furthermore, the cleavage site is assigned by a probability score together with scores for the n-region, h-region, and c-region of the signal peptide, if it is found.

**Other useful resources**

http://www.cbs.dtu.dk/services/SignalP

Pubmed entries for some of the original papers.
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=pubmed&cmd=Retrieve&dopt=AbstractPlus&list_uids=9051728&query_hl=1&itool=pubmed_docsum

http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=15223320&dopt=Citation

**Creative Commons License**

All CLC bio's scientific articles are licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 2.5 License. You are free to copy, distribute, display, and use the work for educational

purposes, under the following conditions: You must attribute the work in its original form and "CLC bio" has to be clearly labeled as author and provider of the work. You may not use this work for commercial purposes. You may not alter, transform, nor build upon this work.

SOME RIGHTS RESERVED

See http://creativecommons.org/licenses/by-nc-nd/2.5/ for more information on how to use the contents.

## 16.2 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 16.6:

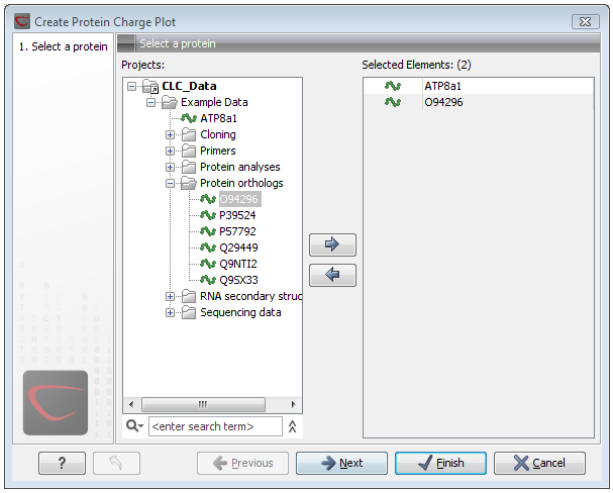



Figure 16.6: *Choosing protein sequences to calculate protein charge.*

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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

### 16.2.1 Modifying the layout

Figure 16.7 shows the electrical charges for three proteins. In the **Side Panel** to the right, you can modify the layout of the graph.

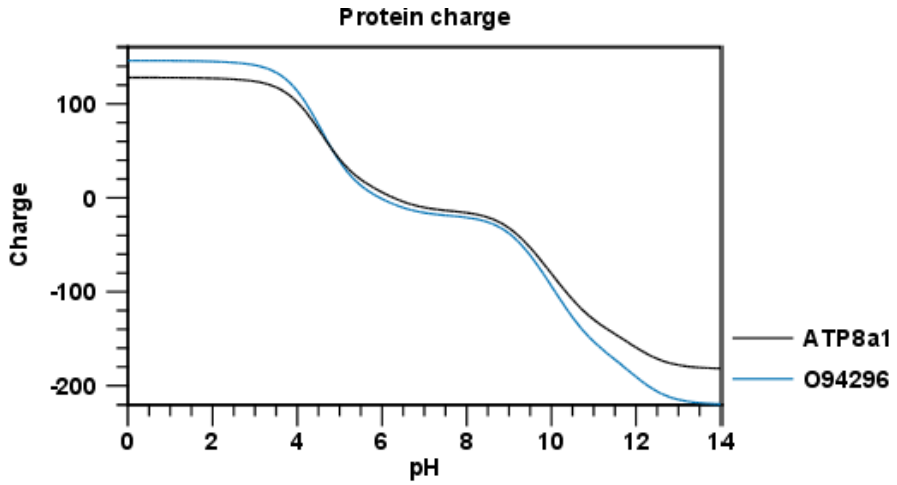



Figure 16.7: *View of the protein charge.*

See section C in the appendix for information about the graph view.

## 16.3 Transmembrane helix prediction

Many proteins are integral membrane proteins. Most membrane proteins have hydrophobic regions which span the hydrophobic core of the membrane bi-layer and hydrophilic regions located on the outside or the inside of the membrane. Many receptor proteins have several transmembrane helices spanning the cellular membrane.

For prediction of transmembrane helices, *CLC Genomics Workbench* uses TMHMM version 2.0 [Krogh et al., 2001] located at `http://www.cbs.dtu.dk/services/TMHMM/`, thus an active internet connection is required to run the transmembrane helix prediction. Additional information on THMHH and Center for Biological Sequence analysis (CBS) can be found at `http://www.cbs.dtu.dk` and in the original research paper [Krogh et al., 2001].

In order to use the transmembrane helix prediction, *you need to download the plugin* using the plugin manager (see section 1.7.1).

When the plugin is downloaded and installed, you can use it to predict transmembrane helices:

> **Toolbox** | **Classical Sequence Analysis** ( ) | **Protein Analysis** ( )| **Transmembrane Helix 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.

The predictions obtained can either be shown as annotations on the sequence, in a table or as the detailed and text output from the TMHMM method.

- Add annotations to sequence

- Create table
- Text

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

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 if a transmembrane helix is found. If a transmembrane helix is not found a dialog box will be presented.

After running the prediction as described above, the protein sequence will show predicted transmembrane helices as annotations on the original sequence (see figure 16.8). Moreover, annotations showing the topology will be shown. That is, which part the proteins is located on the inside or on the outside.

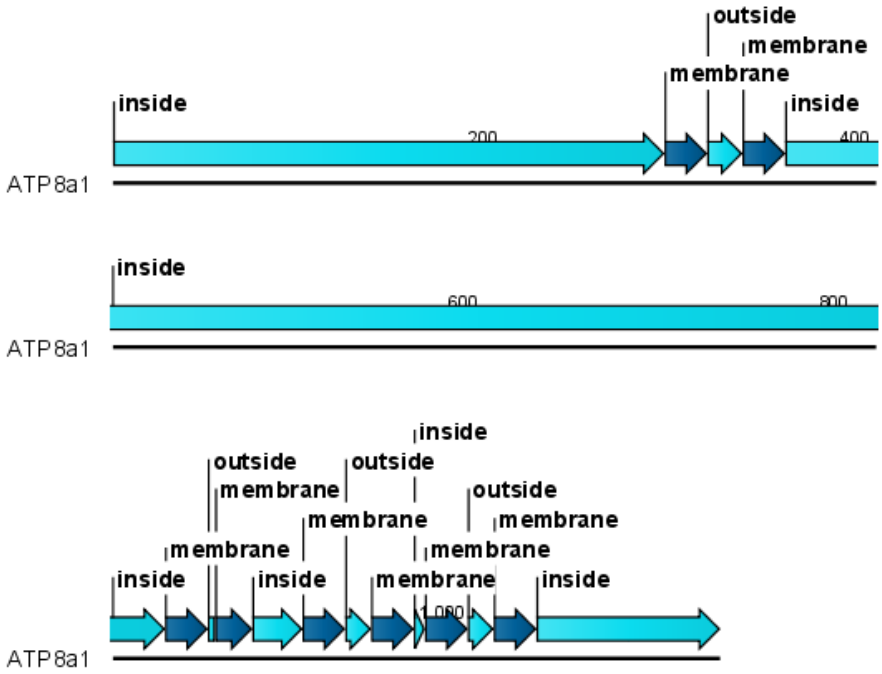



Figure 16.8: *Transmembrane segments shown as annotation on the sequence and the topology.*

Each annotation will carry a tooltip note saying that the corresponding annotation is predicted with TMHMM version 2.0. Additional notes can be added through the **Edit annotation** ( ) right-click mouse menu. See section 10.3.2.

Undesired annotations can be removed through the **Delete Annotation ( )** right-click mouse menu. See section 10.3.4.

## 16.4 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 method can not always be expected as the two methods are based on different training sets.

### 16.4.1 Plot of antigenicity

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 16.9.

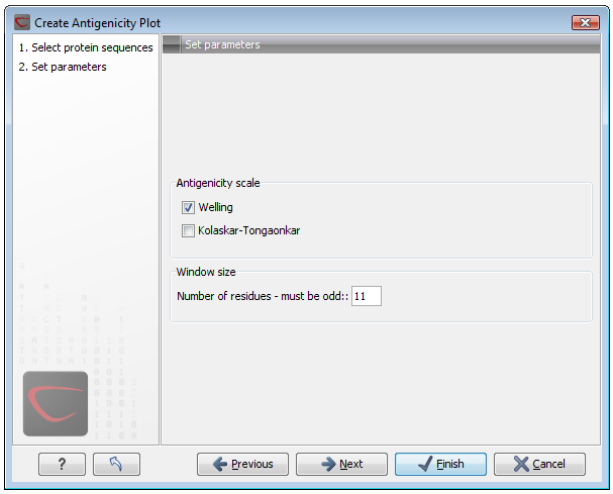



Figure 16.9: *Step two in the Antigenicity Plot allows you to choose different antigenicity scales and the window size.*

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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**. The result can be seen in figure 16.10.

See section C in the appendix for information about the graph view.

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

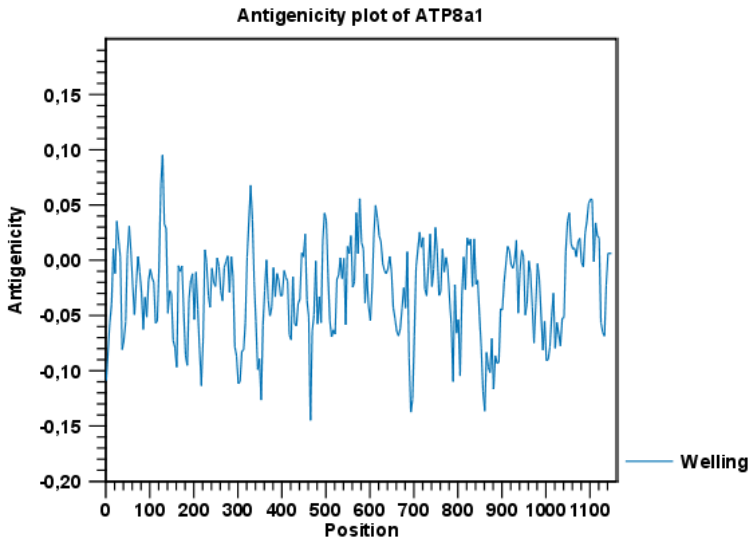



Figure 16.10: *The result of the antigenicity plot calculation and the associated Side Panel.*

can be set from 5 to 25 residues. The wider the window, the less fluctuations in the antigenicity scores.

### 16.4.2 Antigenicity graphs along sequence

Antigenicity graphs along the sequence can be displayed using the **Side Panel**. The functionality is similar to hydrophobicity (see section 16.5.2).

## 16.5 Hydrophobicity

*CLC Genomics Workbench* can calculate the hydrophobicity of protein sequences in different ways, using different algorithms. (See section 16.5.3). 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.

### 16.5.1 Hydrophobicity plot

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 16.11.

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 16.5.3 Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**. The result can be seen in figure 16.12.

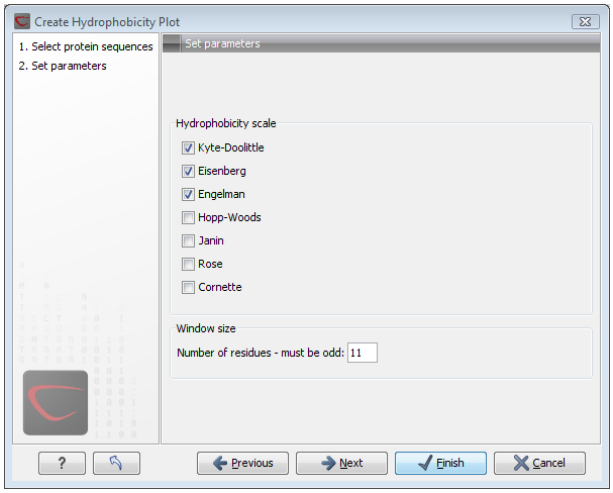



Figure 16.11: *Step two in the Hydrophobicity Plot allows you to choose hydrophobicity scale and the window size.*

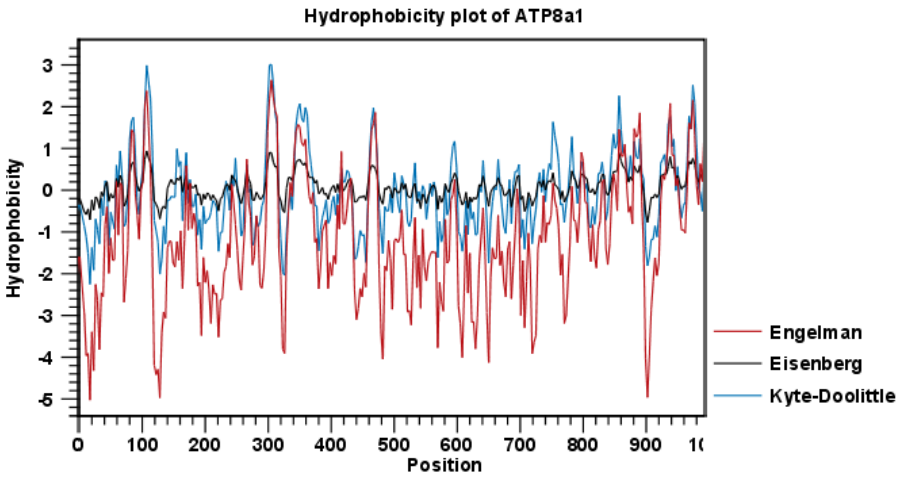



Figure 16.12: *The result of the hydrophobicity plot calculation and the associated Side Panel.*

See section C in the appendix for information about the graph view.

### 16.5.2 Hydrophobicity graphs along sequence

Hydrophobicity graphs along sequence can be displayed easily by activating the calculations from the **Side Panel** for a sequence.

**right-click protein sequence in Navigation Area | Show | Sequence | open Protein info in Side Panel**

or **double-click protein sequence in Navigation Area | Show | Sequence | open Protein info in Side Panel**

These actions result in the view displayed in figure 16.13.

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




Figure 16.13: *The different available scales in Protein info in* **CLC Genomics Workbench**.

hydrophobicity scores. (For more about the theory behind hydrophobicity, see 16.5.3 ).

In the following we will focus on the different ways that *CLC Genomics 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. (See figure 16.14).

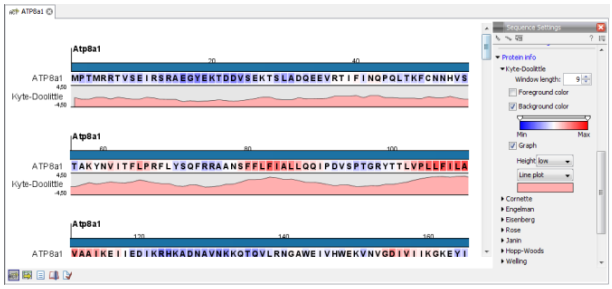


Figure 16.14: *The different ways of displaying the hydrophobicity scores, using the Kyte-Doolittle scale.*

**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 16.14. Notice that you can choose the height of the graphs underneath the sequence.

### 16.5.3 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.

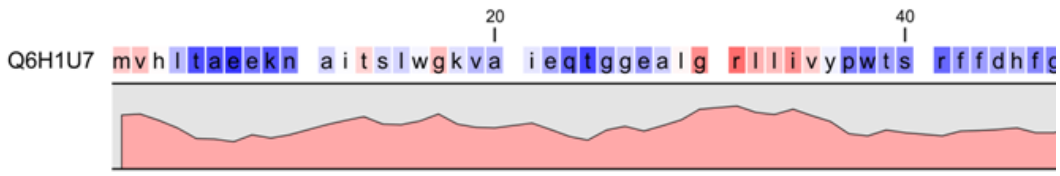



Figure 16.15: *Plot of hydrophobicity along the amino acid sequence. Hydrophobic regions on the sequence have higher numbers according to the graph below the sequence, furthermore hydrophobic regions are colored on the sequence. Red indicates regions with high hydrophobicity and blue indicates regions 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 16.15).

### 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

| 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 16.1: *Hydrophobicity scales. This table shows seven different hydrophobicity scales which are generally used for prediction of e.g. transmembrane regions and antigenicity.*

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.

### Other useful resources

AAindex: Amino acid index database
http://www.genome.ad.jp/dbget/aaindex.html

### Creative Commons License

All CLC bio's scientific articles are licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 2.5 License. You are free to copy, distribute, display, and use the work for educational purposes, under the following conditions: You must attribute the work in its original form and

"CLC bio" has to be clearly labeled as author and provider of the work. You may not use this work for commercial purposes. You may not alter, transform, nor build upon this work.




See http://creativecommons.org/licenses/by-nc-nd/2.5/ for more information on how to use the contents.

## 16.6 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 that cover 14831 protein domains and protein families as of March 2014. The database 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 e.g. 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 10.3.4.

### 16.6.1 Download of 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.

### 16.6.2 Running Pfam Domain Search

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 16.16).

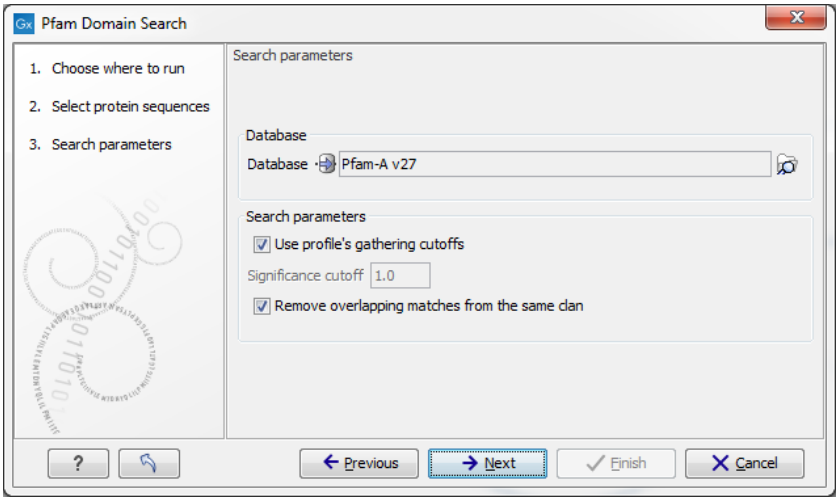



Figure 16.16: *Setting parameters for Pfam Domain Search.*

- **Database.** Choose which database to use when searching for Pfam domains. For information on how to download a Pfam database see section 16.6.1
- 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 16.17) and it can output a table with all the domains found.

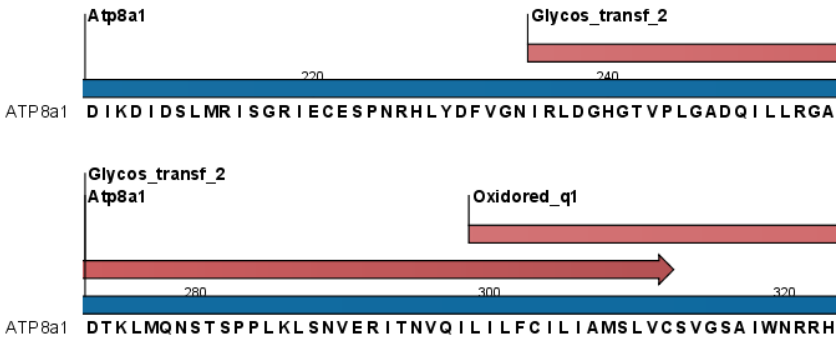



Figure 16.17: *Annotations (in red) that were added by the Pfam search tool.*

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

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.

## 16.7 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 16.18:

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.

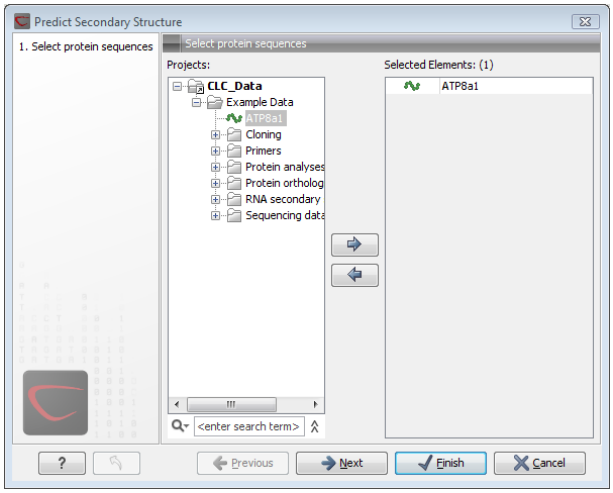



Figure 16.18: *Choosing one or more protein sequences for secondary structure prediction.*

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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

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 16.19).

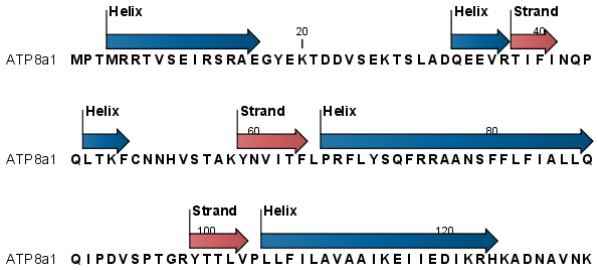



Figure 16.19: *Alpha-helices and beta-strands shown as annotations on the sequence.*

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 10.3.2.

Undesired alpha-helices or beta-sheets can be removed through the **Delete Annotation** ( ) right-click mouse menu. See section 10.3.4.

## 16.8 Protein report

*CLC Genomics Workbench* is able to produce protein reports, that allow you to easily generate different kinds of information regarding a protein.

Actually a protein report is a collection of some of the protein analyses which are described elsewhere in this manual.

To create a protein report do the following:

**Toolbox** | **Classical Sequence Analysis (**  **)** | **Protein Analysis (**  **)**| **Create Protein Report (**  **)**

This opens dialog **Step 1**, 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 dialog **Step 2** 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.** See section 14.6 for more about this topic.
- **Plot of charge as function of pH.** See section 16.2 for more about this topic.
- **Plot of hydrophobicity.** See section 16.5 for more about this topic.
- **Plot of local complexity.** See section 14.5 for more about this topic.
- **Dot plot against self.** See section 14.4 for more about this topic.
- **Secondary structure prediction.** See section 16.7 for more about this topic.
- **Pfam domain search.** See section 16.6 for more about this topic.
- **Local BLAST.** See section 12.1.3 for more about this topic.
- **NCBI BLAST.** See section 12.1.1 for more about this topic.

When you have selected the relevant analyses, click **Next**. **Step 3** to **Step 7** (if you select all the analyses in **Step 2**) are adjustments of parameters for the different analyses. The parameters are mentioned briefly in relation to the following steps, and you can turn to the relevant chapters or sections (mentioned above) to learn more about the significance of the parameters.

In **Step 3** you can adjust parameters 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.

In **Step 4** you can adjust parameters for hydrophobicity plots:

- **Window size.** Width of window on sequence (odd number).
- **Hydrophobicity scales.** Lets you choose between different scales.

In **Step 5** you can adjust a parameter for complexity plots:

- **Window size.** Width of window on sequence (must be odd).

In **Step 6** you can adjust parameters for dot plots:

- **Score model.** Different scoring matrices.
- **Window size.** Width of window on sequence.

In **Step 7** you can adjust parameters for Pfam domain search:

- **Database and search type.** Lets you choose different databases and specify the search for full domains or fragments. See section 16.6.2 for more info about this topic.
- **Significance cutoff.** Lets you set your E-value. See section 16.6.2 for more info about this topic.

In **Step 8** you can adjust parameters for BLAST search:

- **Program.** Lets you choose between different BLAST programs.
- **Database.** Lets you limit your search to a particular database.

### 16.8.1 Protein report output

An example of Protein report can be seen in figure 16.20.

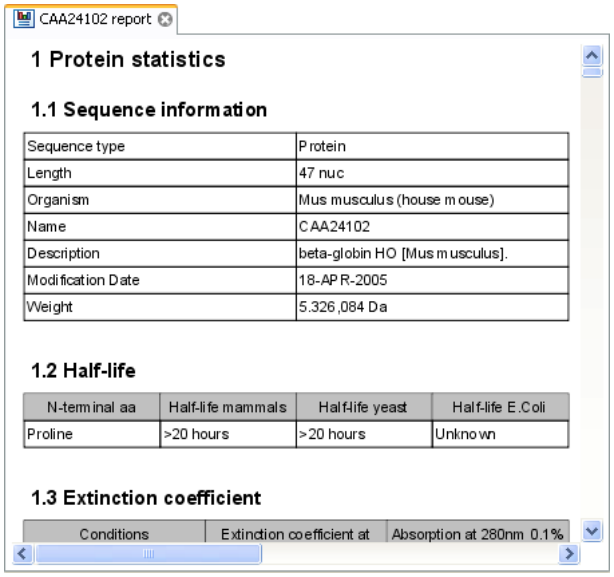



Figure 16.20: *A protein report. There is a Table of Contents in the Side Panel that makes it easy to browse the report.*

By double clicking a graph in the output, this graph is shown in a different view (*CLC Genomics Workbench* generates another tab). The report output and the new graph views can be saved by dragging the tab into the **Navigation Area**.

The content of the tables in the report can be copy/pasted out of the program and e.g. into Microsoft Excel. To do so:

**Select content of table** | Right-click the selection | Copy

You can also **Export** (  ) the report in Excel format.

## 16.9 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 16.9.2.

In order to make a reverse translation:

**Toolbox** | **Classical Sequence Analysis (** ) | **Protein Analysis (** )| **Reverse Translate (** )

This opens the dialog displayed in figure 16.21:

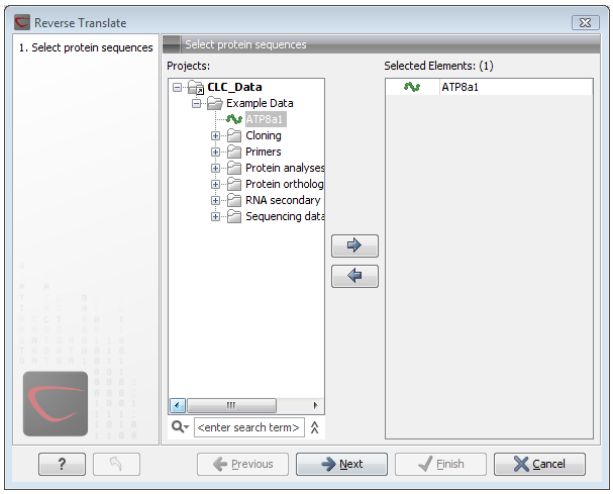



Figure 16.21: *Choosing a protein sequence for reverse translation.*

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.

Click **Next** to adjust the parameters for the translation.

### 16.9.1 Reverse translation parameters

Figure 16.22 shows the choices for making the translation.

- **Use random codon.** This will randomly back-translate an amino acid to a codon without using the translation tables. Every time you perform the analysis you will get a different result.
- **Use only the most frequent codon.** On the basis of the selected translation table, this parameter/option will assign the codon that occurs most often. When choosing this option, the results of performing several reverse translations will always be the same, contrary to the other two options.
- **Use codon based on frequency distribution.** This option is a mix of the other two options. The selected translation table is used to attach weights to each codon based on its

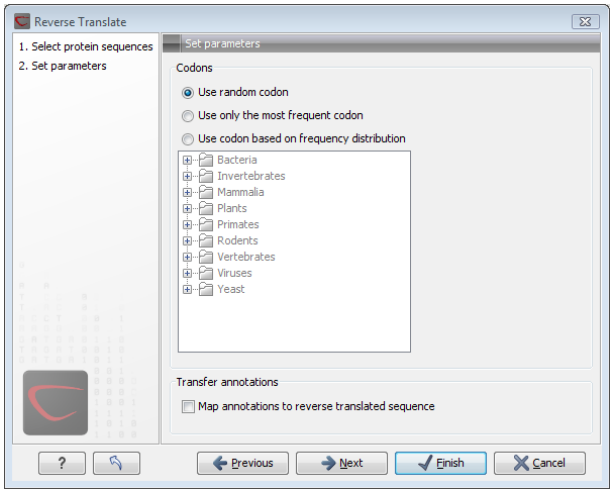



Figure 16.22: *Choosing parameters for the reverse translation.*

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 M.

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**. 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. 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 show the save dialog.

### 16.9.2 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

**Creative Commons License**

All CLC bio's scientific articles are licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 2.5 License. You are free to copy, distribute, display, and use the work for educational purposes, under the following conditions: You must attribute the work in its original form and "CLC bio" has to be clearly labeled as author and provider of the work. You may not use this work for commercial purposes. You may not alter, transform, nor build upon this work.




See http://creativecommons.org/licenses/by-nc-nd/2.5/ for more information on how to use the contents.

## 16.10 Proteolytic cleavage detection

*CLC Genomics Workbench* offers to analyze protein sequences with respect to cleavage by a selection of proteolytic enzymes. This section explains how to adjust the detection parameters and offers basic information on proteolytic cleavage in general.

### 16.10.1 Proteolytic cleavage parameters

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 and in textual format 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 16.23:

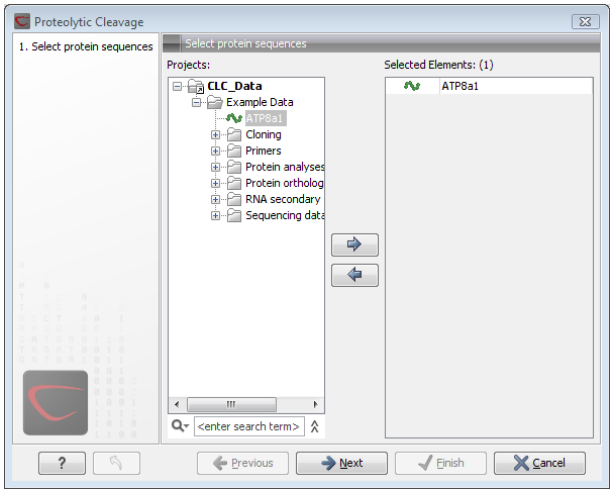



Figure 16.23: *Choosing sequence CAA32220 for proteolytic cleavage.*

*CLC Genomics Workbench* allows you to detect proteolytic cleavages for several sequences at a time. Correct the list of sequences by selecting a sequence and clicking the arrows pointing left and right. Then click **Next** to go to **Step 2**.

In **Step 2** you can select proteolytic cleavage enzymes. The list of available enzymes will be expanded continuously. Presently, the list contains the enzymes shown in figure 16.24. The full list of enzymes and their cleavage patterns can be seen in Appendix, section E.

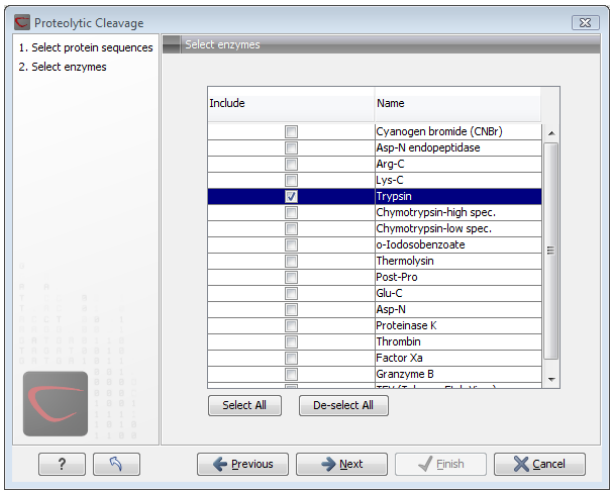



Figure 16.24: *Setting parameters for proteolytic cleavage detection.*

Select the enzymes you want to use for detection. When the relevant enzymes are chosen, click **Next**.

In **Step 3** you can set parameters for the detection. This limits the number of detected cleavages. Figure 16.25 shows an example of how parameters can be set.

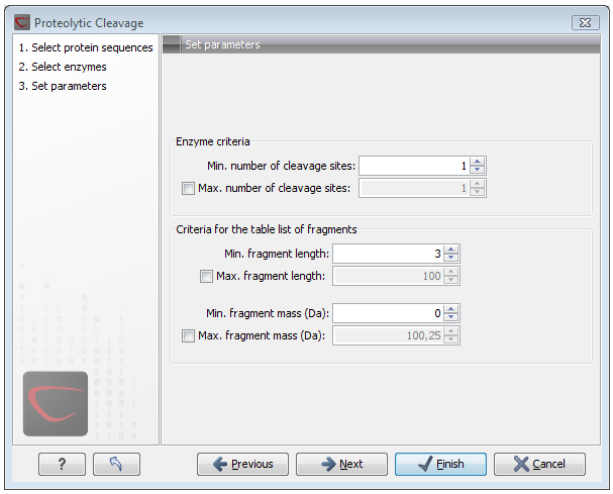



Figure 16.25: *Setting parameters for proteolytic cleavage detection.*

- **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.

**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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

The result of the detection is displayed in figure 16.26.

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.

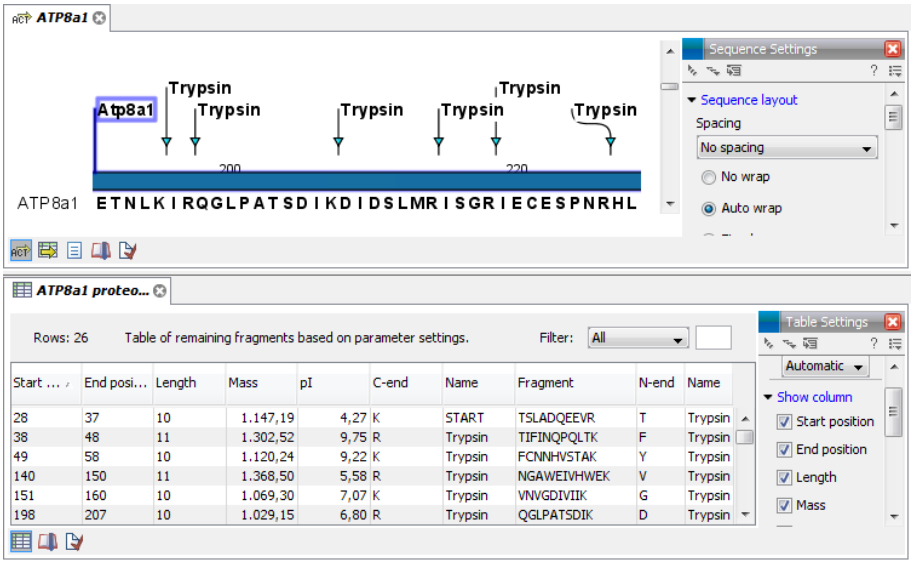



Figure 16.26: *The result of the proteolytic cleavage detection.*

### 16.10.2 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 16.27.

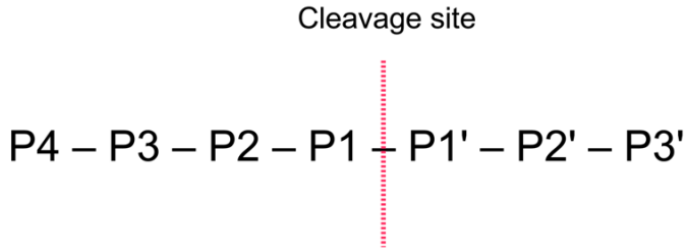



Figure 16.27: *Nomenclature of the peptide substrate. The substrate is cleaved between position P1-P1'.*

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 16.28).

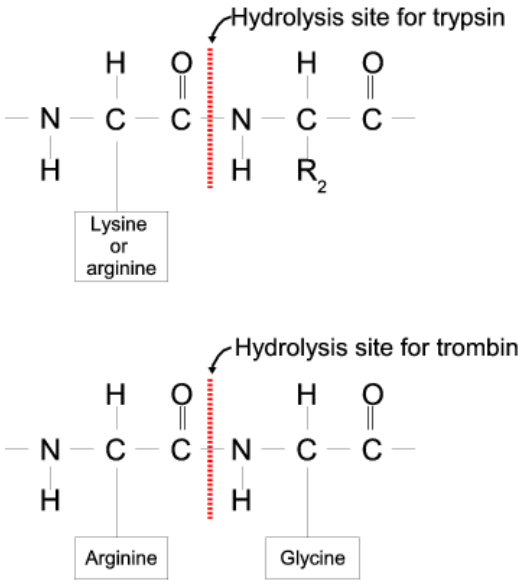



Figure 16.28: *Hydrolysis of the peptide bond between two amino acids. Trypsin cleaves unspecifically at lysine or arginine residues whereas trombin cleaves at arginines if asparate or glutamate is absent.*

Bioinformatics approaches are used to identify potential peptidase cleavage sites. Fragments can be found by scanning the amino acid sequence for patterns which match the corresponding cleavage site for the protease. When identifying cleaved fragments it is relatively important to know the calculated molecular weight and the isoelectric point.

**Other useful resources**

The Peptidase Database: http://merops.sanger.ac.uk/

### Creative Commons License

All CLC bio's scientific articles are licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 2.5 License. You are free to copy, distribute, display, and use the work for educational purposes, under the following conditions: You must attribute the work in its original form and "CLC bio" has to be clearly labeled as author and provider of the work. You may not use this work for commercial purposes. You may not alter, transform, nor build upon this work.




See http://creativecommons.org/licenses/by-nc-nd/2.5/ for more information on how to use the contents.

# Chapter 17

# Primers

## Contents


**17.1 Primer design - an introduction . . . 342**
17.1.1 General concept . . . 342
17.1.2 Scoring primers . . . 344
**17.2 Setting parameters for primers and probes . . . 344**
17.2.1 Primer Parameters . . . 345
**17.3 Graphical display of primer information . . . 347**
17.3.1 Compact information mode . . . 347
17.3.2 Detailed information mode . . . 347
**17.4 Output from primer design . . . 348**
17.4.1 Saving primers . . . 349
17.4.2 Saving PCR fragments . . . 349
17.4.3 Adding primer binding annotation . . . 349
**17.5 Standard PCR . . . 349**
17.5.1 User input . . . 350
17.5.2 Standard PCR output table . . . 352
**17.6 Nested PCR . . . 353**
17.6.1 Nested PCR output table . . . 355
**17.7 TaqMan . . . 355**
17.7.1 TaqMan output table . . . 357
**17.8 Sequencing primers . . . 357**
17.8.1 Sequencing primers output table . . . 358
**17.9 Alignment-based primer and probe design . . . 358**
17.9.1 Specific options for alignment-based primer and probe design . . . 358
17.9.2 Alignment based design of PCR primers . . . 359
17.9.3 Alignment-based TaqMan probe design . . . 361
**17.10 Analyze primer properties . . . 363**
**17.11 Find binding sites and create fragments . . . 364**
17.11.1 Binding parameters . . . 365
17.11.2 Results - binding sites and fragments . . . 365
**17.12 Order primers . . . 368**

*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.

## 17.1 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** ( )

In the primer view (see figure 17.1), the basic options for viewing the template sequence are the same as for the standard sequence view. See section 10.1 for an explanation of these options.

**Note!** This means that annotations such as e.g. known SNPs or exons, can be displayed on the template sequence to guide the choice of primer regions. Also, traces in sequencing reads can be shown along with the structure to guide e.g. the re-sequencing of poorly resolved regions.




Figure 17.1: *The initial view of the sequence used for primer design.*

### 17.1.1 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 17.2).

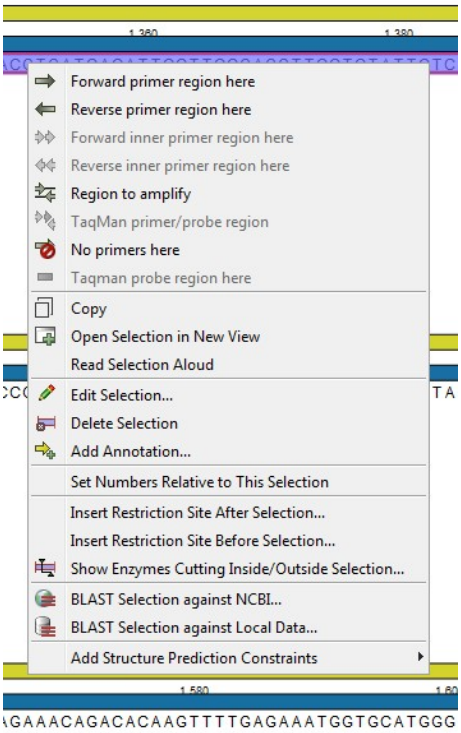



Figure 17.2: *Right-click menu allowing you to specify regions for the primer design*

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 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.

### 17.1.2 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.

## 17.2 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 17.3).

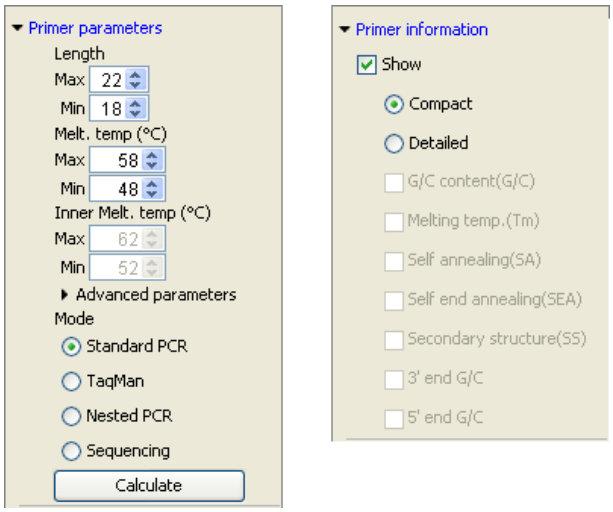



Figure 17.3: *The two groups of primer parameters (in the program, the Primer information group is listed below the other group).*

### 17.2.1 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 ($nM$)
    - **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

## 17.3 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 17.3).

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.

### 17.3.1 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 17.4).

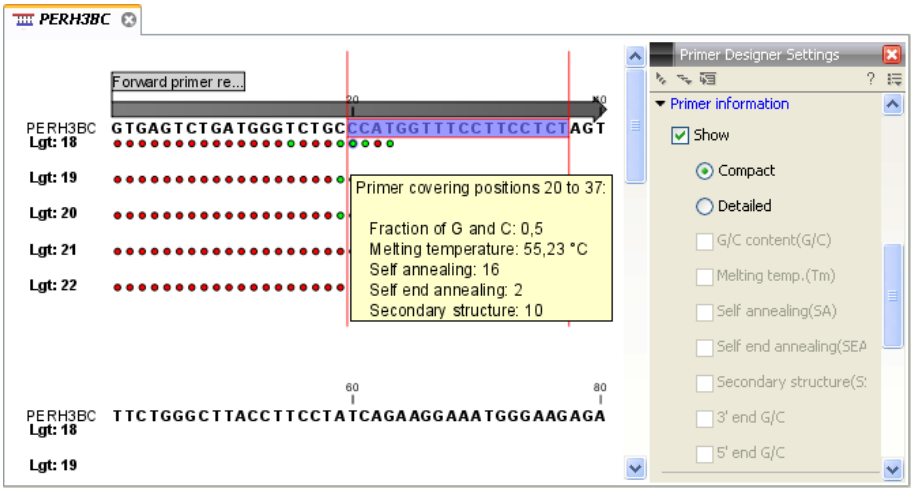



Figure 17.4: *Compact information mode*

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.

### 17.3.2 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 17.5).

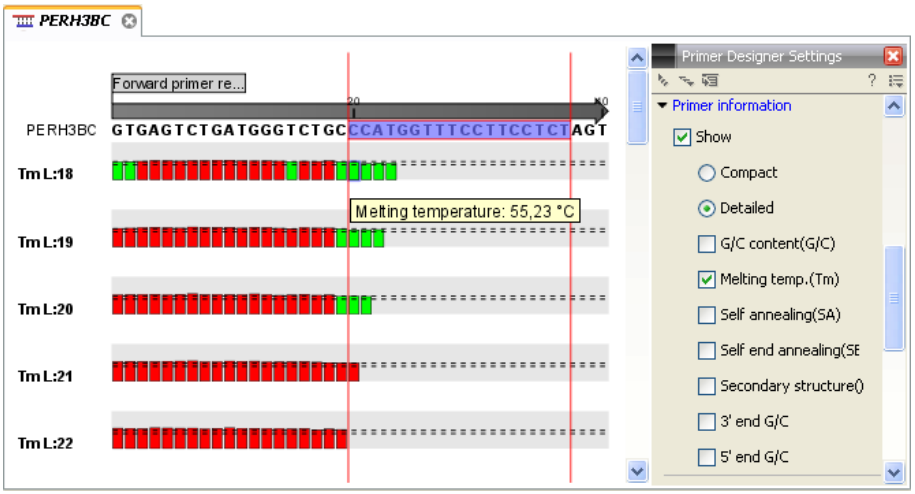



Figure 17.5: *Detailed information mode*

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
- 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.

## 17.4 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 17.6).

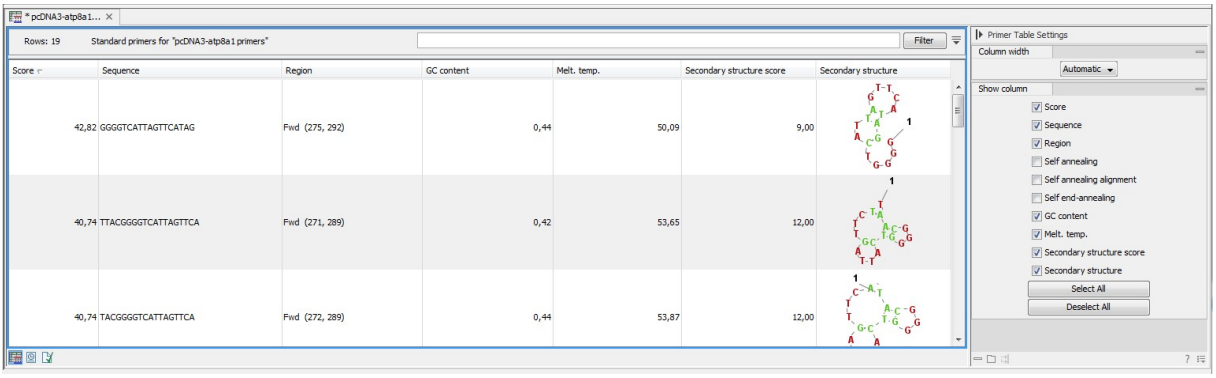



Figure 17.6: *Proposed primers*

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.

### 17.4.1 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, including BLAST. Furthermore, the primers can be edited using the standard sequence view to introduce e.g. mutations and restriction sites.

### 17.4.2 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.

### 17.4.3 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**.

## 17.5 Standard PCR

This mode is used to design primers for a PCR amplification of a single DNA fragment.

### 17.5.1 User input

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 17.3) and setting the desired parameter values in the Primer Parameters preference group, the **Calculate** button will activate the primer design algorithm.

#### 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 17.7).

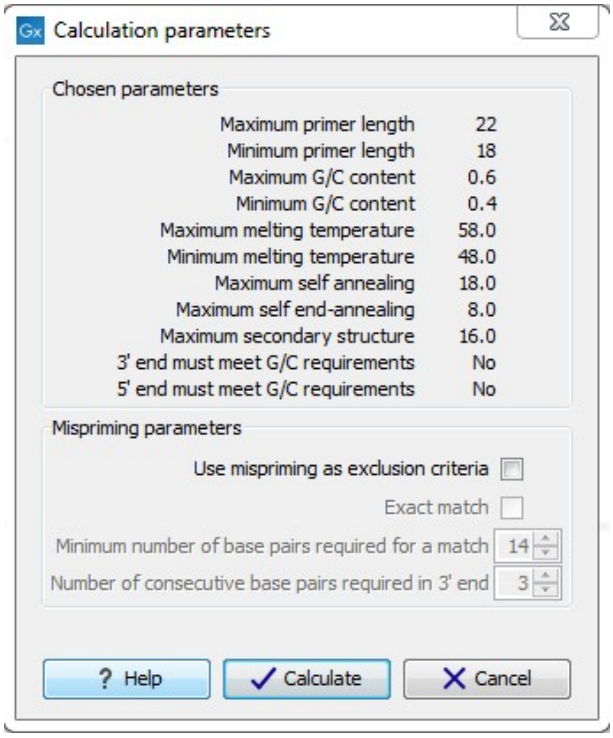



Figure 17.7: *Calculation dialog for PCR primers when only a single primer region has been defined.*

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.

### 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 17.8).

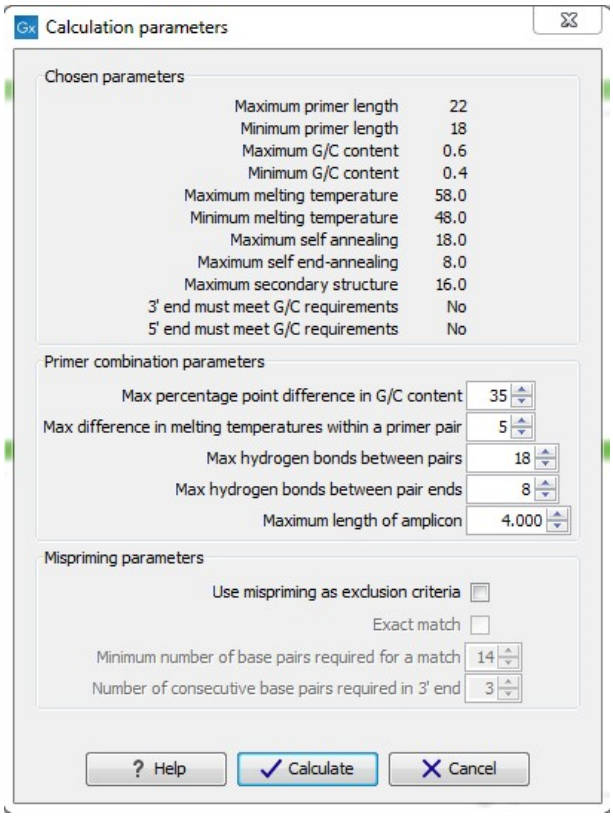



Figure 17.8: *Calculation dialog for PCR primers when two primer regions have been defined.*

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 17.5.1). 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.

### 17.5.2 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

## 17.6 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 17.3) 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 17.9).

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:

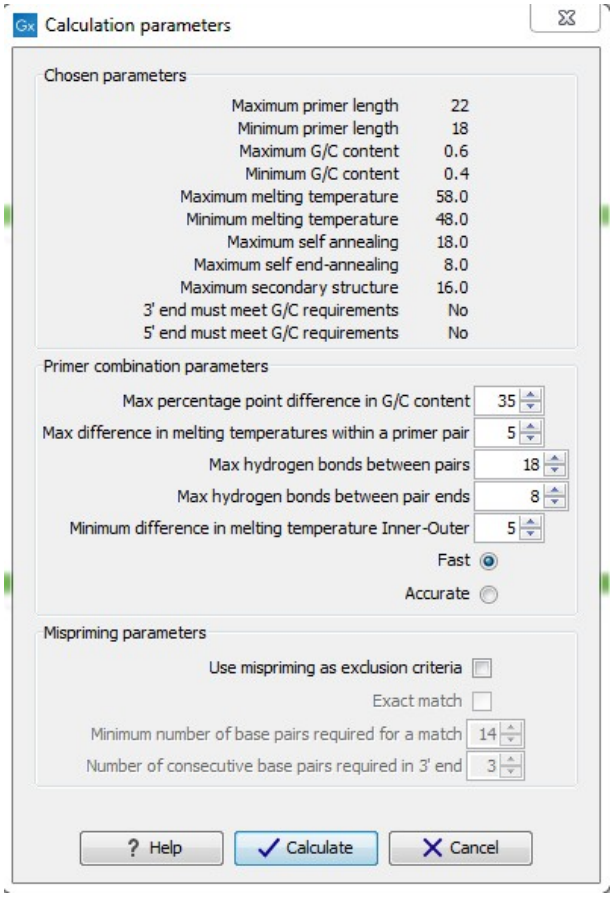



Figure 17.9: *Calculation dialog*

- 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.

### 17.6.1 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.

## 17.7 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 17.3) 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 17.10) which is similar to the *Nested PCR* dialog described above (see section 17.6).

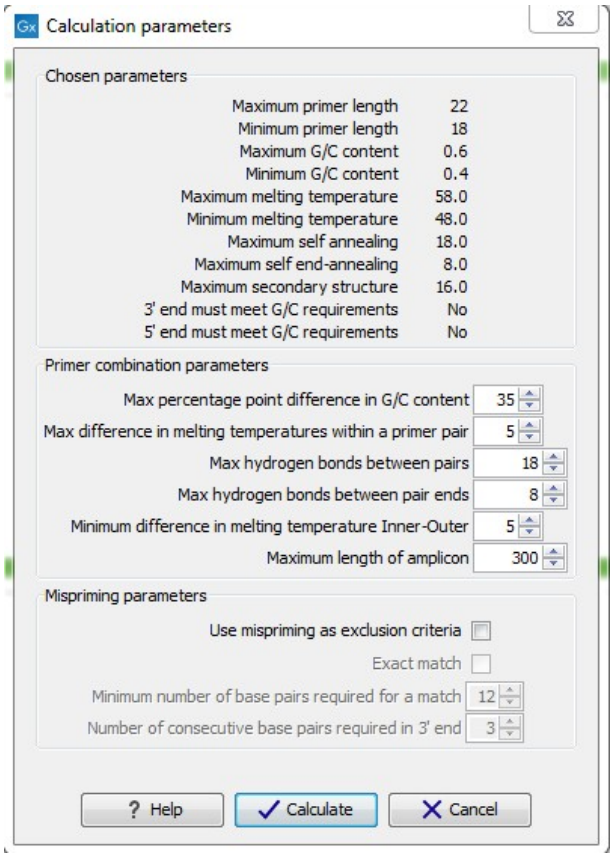



Figure 17.10: *Calculation dialog*

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.

### 17.7.1 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).

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.

## 17.8 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 17.3) 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 17.11).

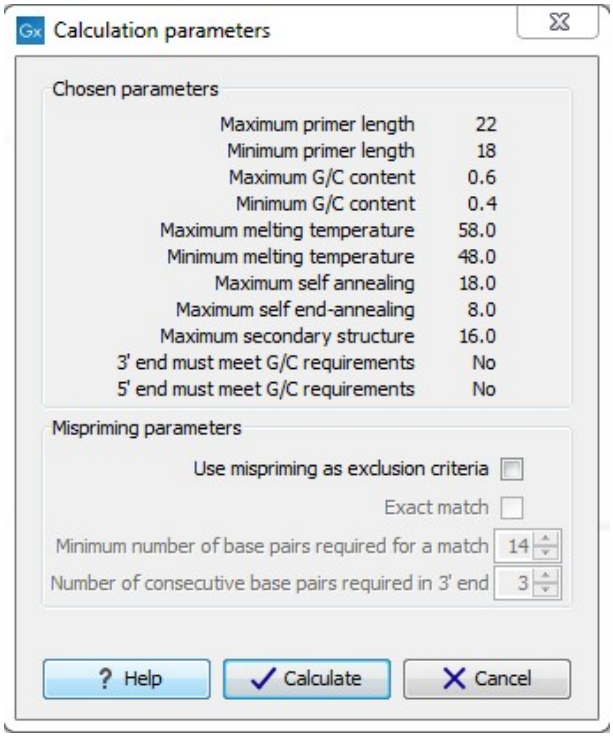



Figure 17.11: *Calculation dialog for sequencing primers*

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 the section 17.5 for a description.

### 17.8.1 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.

For each solution, the single primer information described under Standard PCR is available in the table.

## 17.9 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 in two ways:

**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 17.12), the basic options for viewing the template alignment are the same as for the standard view of alignments.

See section 20 for an explanation of these options.

**Note!** This means that annotations such as e.g. known SNPs or exons, can be displayed on the template sequence to guide the choice of primer regions. Since the definition of groups of sequences is essential to the primer design, the selection boxes of the standard view are shown as default in the alignment primer view.

### 17.9.1 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. This is elaborated below. The **Primer Parameters** group in the **Side Panel** has the same options for specifying primer requirements, but differs by the following (see figure 17.12):

- In the **Mode** submenu which specifies the reaction types the following options are found:
  - **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.




Figure 17.12: *The initial view of an alignment used for primer design.*

- The **Primer solution** submenu is used to 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 work flow when designing alignment based primers and probes is as follows:

- 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.

### 17.9.2 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 17.13.

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.

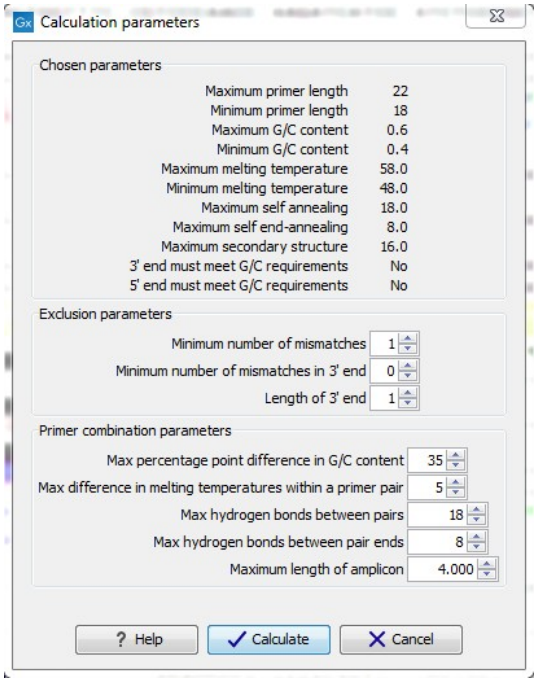



Figure 17.13: *Calculation dialog shown when designing alignment based PCR primers.*

### 17.9.3 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 17.14 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.

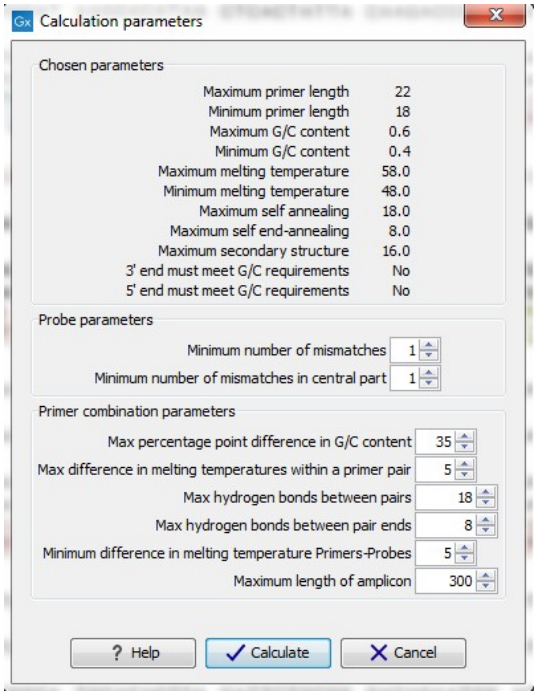



Figure 17.14: *Calculation dialog shown when designing alignment based TaqMan probes.*

## 17.10 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 17.15:

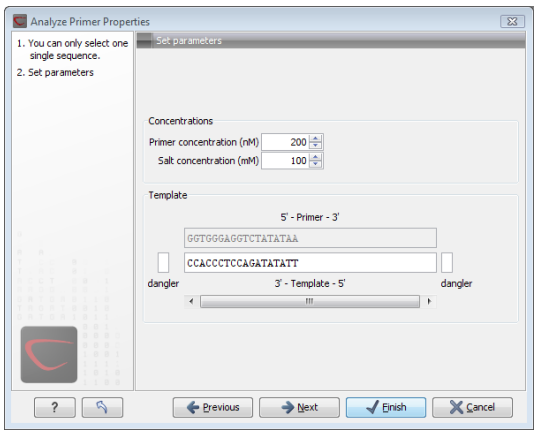



Figure 17.15: *The parameters for analyzing primer properties.*

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 ($nM$)
- **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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**. The result is shown in figure 17.16:

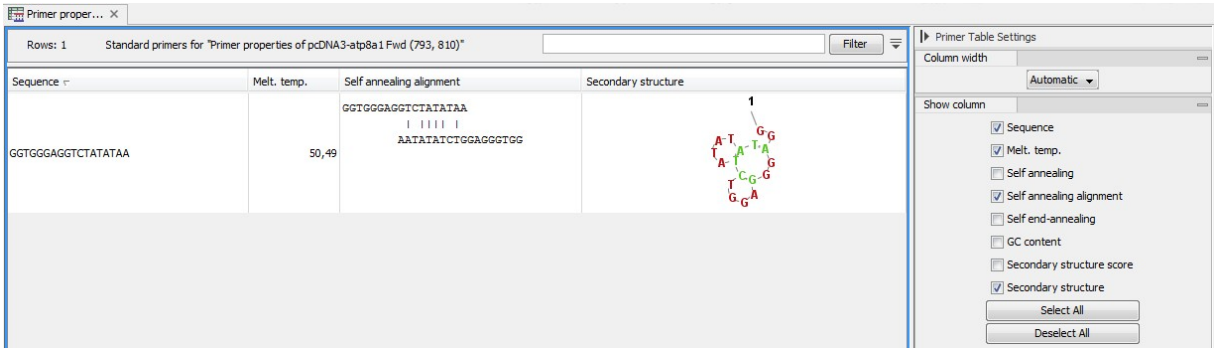



Figure 17.16: *Properties of a primer from the Example Data.*

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 17.5.2.

## 17.11 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 e.g. 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.

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.

### 17.11.1 Binding parameters

This opens the dialog displayed in figure 17.17:

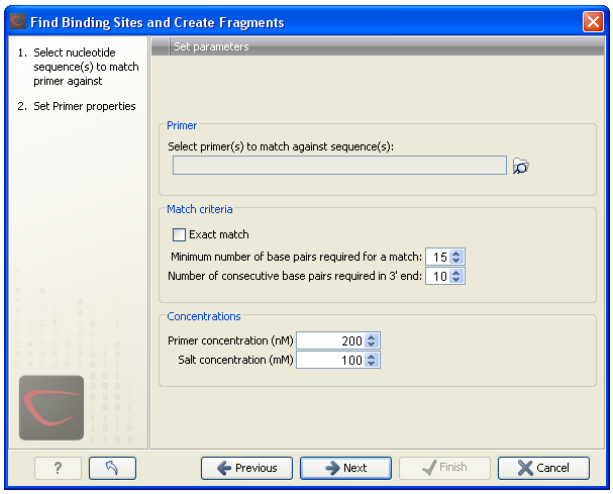



Figure 17.17: *Search parameters for finding primer binding sites.*

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 ($mM$)

### 17.11.2 Results - binding sites and fragments

Click **Next** to specify the output options as shown in figure 17.18:

The output options are:

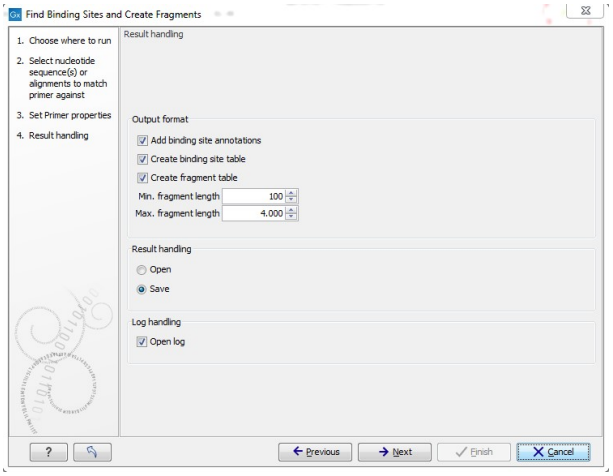



Figure 17.18: *Output options include reporting of binding sites and fragments.*

- **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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

An example of a **binding site annotation** is shown in figure 17.19.

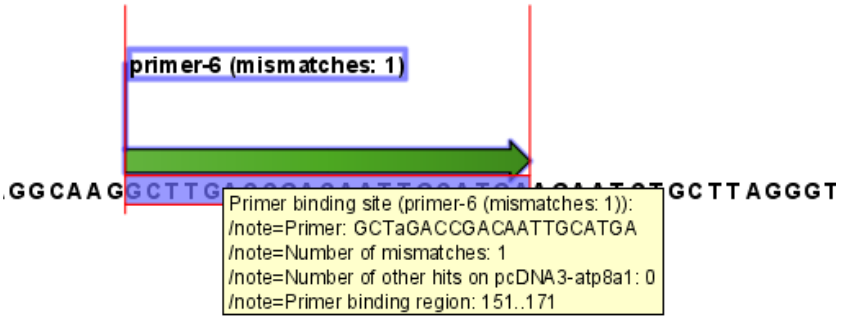



Figure 17.19: *Annotation showing a primer match.*

The annotation has the following information:

- **Sequence of the primer**. Positions with mismatches will be in lower-case (see the fourth position in figure 17.19 where the primer has an `a` and the template sequence has a `T`).
- **Number of mismatches**.
- **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 17.20.

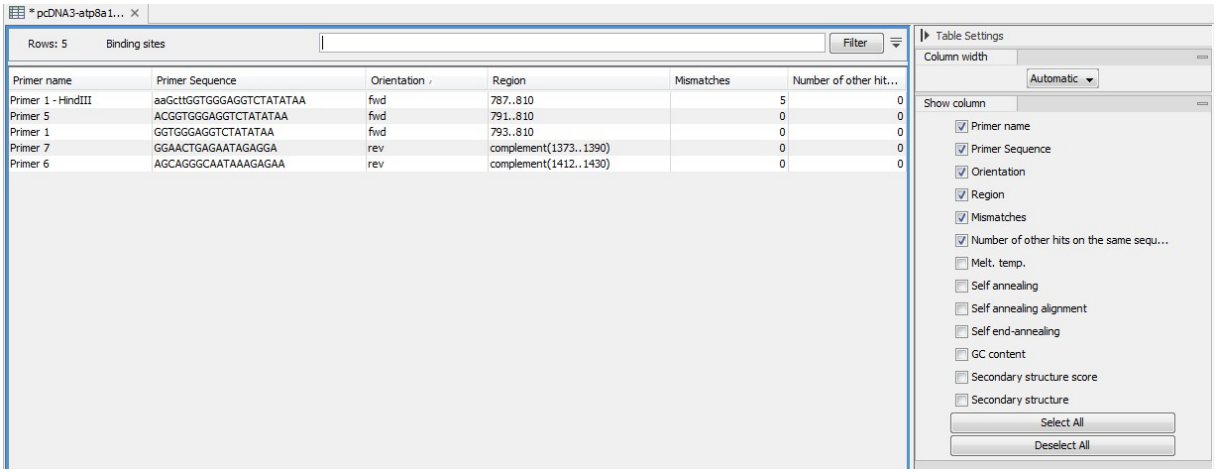



Figure 17.20: *A table showing all binding sites.*

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 17.5.2. You can use this table to browse the binding sites. If you make a split view of the table and the sequence (see section 2.1.6), 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 17.21.

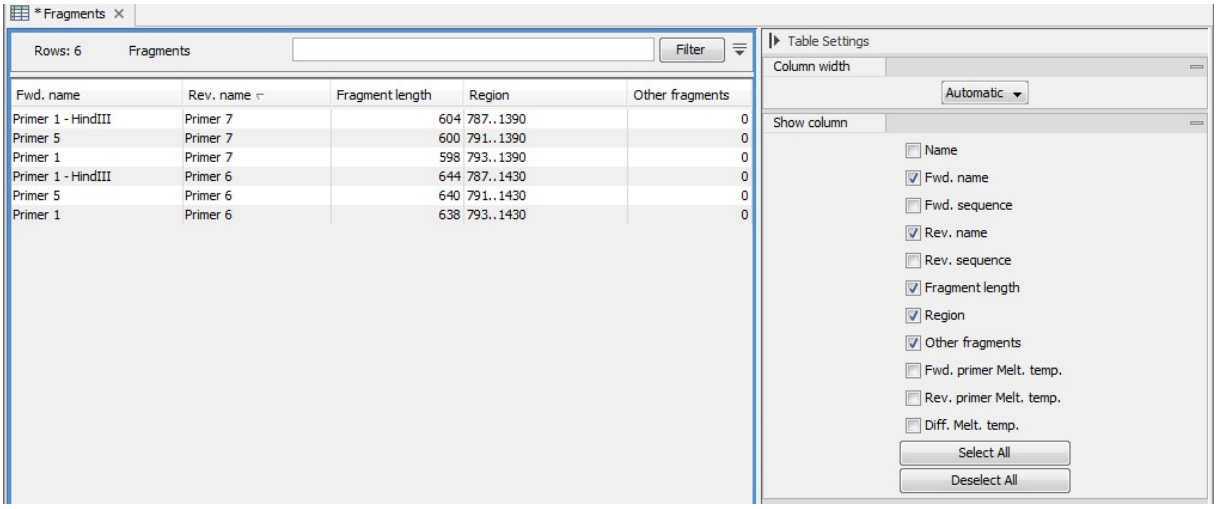



Figure 17.21: *A table showing all possible fragments of the specified size.*

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 the sequence (see section 2.1.6), 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 17.22.

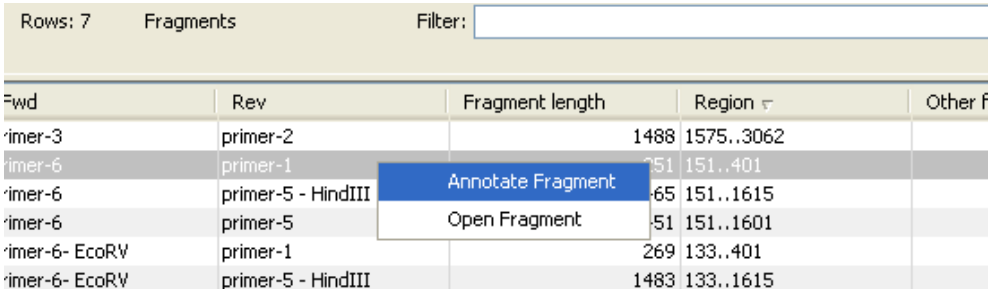



Figure 17.22: *Right-clicking a fragment allows you to annotate the region on the input sequence or open the fragment as a new sequence.*

This will put a *PCR fragment* annotations on the input sequence covering the region specified in the table. As you can see from figure 17.22, 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 19.1).

## 17.12 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 additional primers. Clicking **OK** opens a textual representation of the primers (see figure 17.23). 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. The created object can also be saved and exported as a text file.

See figure 17.23

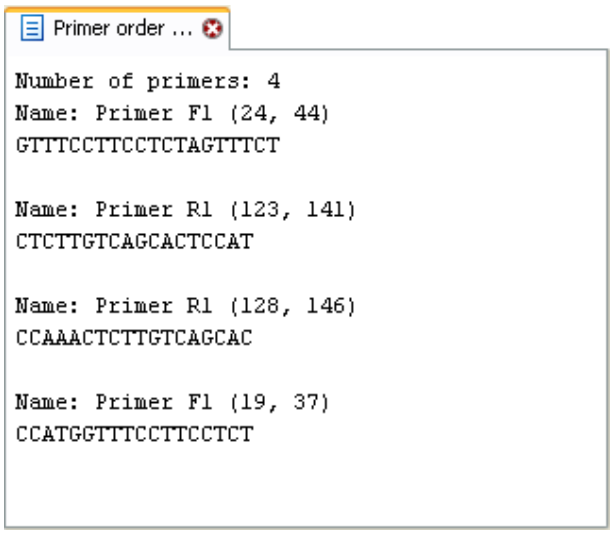



Figure 17.23: *A primer order for 4 primers.*

# Chapter 18

# Sequencing data analyses

## Contents


**18.1 Importing and viewing trace data . . . 371**
18.1.1 Scaling traces . . . 371
18.1.2 Trace settings in the Side Panel . . . 371
**18.2 Trim sequences . . . 372**
18.2.1 Trimming using the Trim tool . . . 373
18.2.2 Manual trimming . . . 375
**18.3 Assemble sequences . . . 375**
**18.4 Sort sequences by name . . . 377**
**18.5 Assemble sequences to reference . . . 380**
**18.6 Add sequences to an existing contig . . . 383**
**18.7 View and edit read mappings . . . 383**
18.7.1 View settings in the Side Panel . . . 385
18.7.2 Editing the read mapping . . . 388
18.7.3 Sorting reads . . . 388
18.7.4 Read conflicts . . . 389
18.7.5 Using the mapping . . . 389
18.7.6 Extract parts of a mapping . . . 389
18.7.7 Variance table . . . 391
**18.8 Reassemble contig . . . 393**
**18.9 Secondary peak calling . . . 394**


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.

## 18.1 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, including e.g. BLAST and open reading frame prediction.

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 18.1.

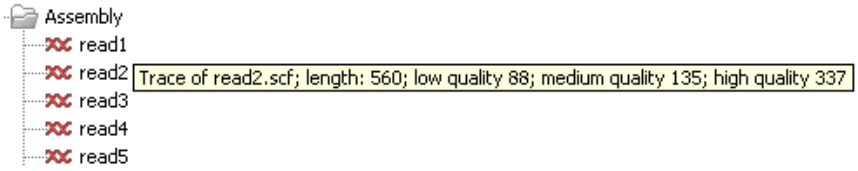



Figure 18.1: *A tooltip displaying information about the quality of the chromatogram.*

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 ( ).

### 18.1.1 Scaling traces

The traces can be scaled by dragging the trace vertically as shown in figure figure 18.2. 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 18.1.2.

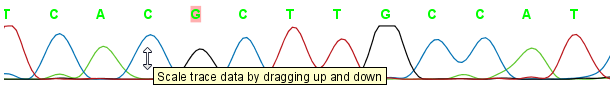



Figure 18.2: *Grab the traces to scale.*

### 18.1.2 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 18.3):

- **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 18.1.1.




Figure 18.3: *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 25.5.2.

## 18.2 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 18.4). 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 18.3). The trimming described here annotates, but does not remove data, allowing you to explore the output of different trimming schemes easily.

Trimming as a separate task can be done manually or using a tool designed specifically for this task.

To remove existing trimming information from a sequence, simply remove its trim annotation (see section 10.3.2).

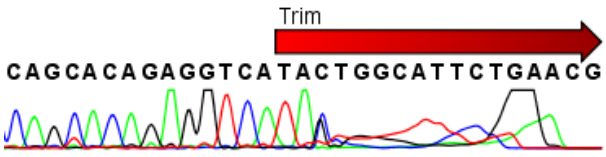



Figure 18.4: *Trimming creates annotations on the regions that will be ignored in the assembly process.*

**Note!** If you wish to remove regions that are trimmed, you should instead use the NGS trim tool (see section 23.1).

When exporting sequences in fasta format, there is an option to remove the parts of the sequence covered by trim annotations.

### 18.2.1 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 tool in the Workbench, go to the menu option:

> **Toolbox** | **Molecular Biology Tools (  )** | **Sequencing Data 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.

When the sequences are selected, click **Next**.

This opens the dialog displayed in figure 18.5.

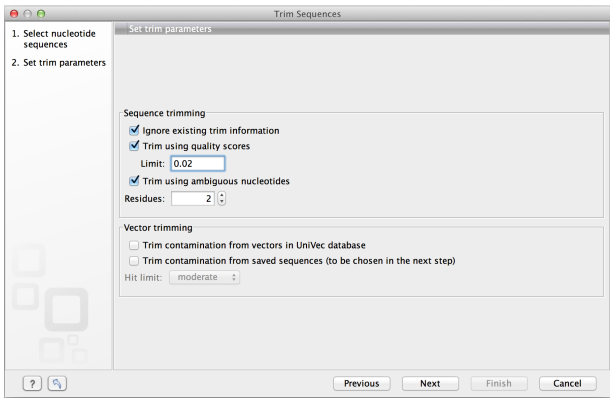



Figure 18.5: *Setting parameters for trimming.*

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://www.clcbio.com/files/usermanuals/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 I).

- **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.** 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.** 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.** 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.** 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.

    Note that selecting e.g. **Weak** will also include matches in the **Moderate** and **Strong** categories.

- **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**.

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**. 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.

### 18.2.2 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 tool) as regions to be ignored.

## 18.3 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 18.5). To perform the assembly:

> **Toolbox | Molecular Biology Tools ( ) | Sequencing Data 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.

**Note!** You can assemble a maximum of 2000 sequences at a time.

To assemble more sequences, please use the **De Novo Assembly** ( ) tool under **De Novo Sequencing ( )** in the **Toolbox** instead.

When the sequences are selected, click **Next**. This will show the dialog in figure 18.6

This dialog gives you the following options for assembly:

- **Minimum aligned read length.** The minimum number of nucleotides in a read which must be successfully aligned to the contig. If this criteria is not met by a read, the read is excluded from the assembly.

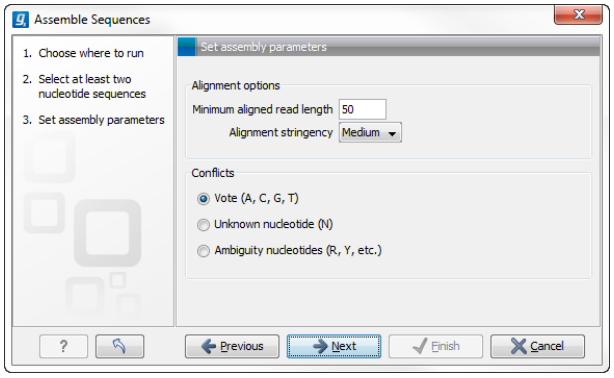



Figure 18.6: *Setting assembly parameters.*

- **Alignment stringency.** Specifies the stringency 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. Three stringency levels can be set:
  - **Low.**
  - **Medium.**
  - **High.**
- **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 I.

  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 18.7.4.
- **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 18.7.
- **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 18.7.7.

- **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.

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

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 18.7 on how to use the resulting contigs.

## 18.4 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** ( ) | **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 18.7.

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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**. 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 18.7.

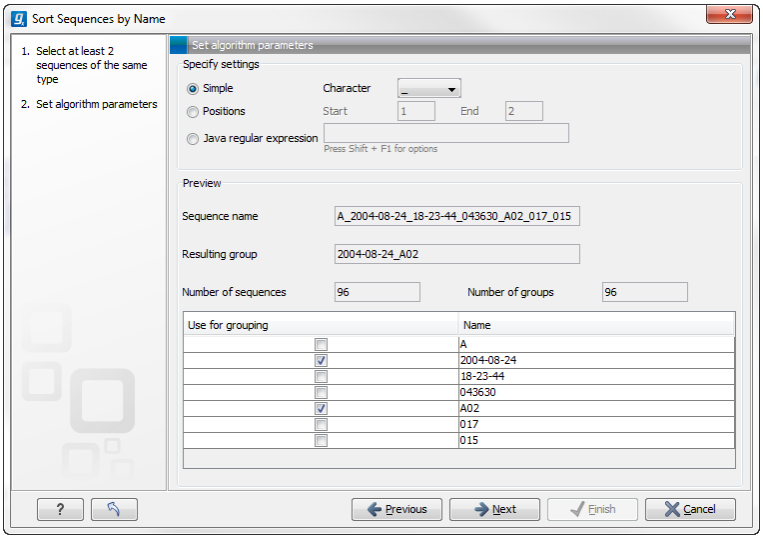



Figure 18.7: *Splitting up the name at every underscore (_) and using the date and analysis position for grouping.*

### Advanced splitting using regular expressions

You can see a more detail explanation of the regular expressions syntax in section 14.9.3. In this section you will see a practical example showing how to create a regular expression. Consider a list of files as shown below:

```
...
adk-29_adk1n-F
adk-29_adk2n-R
adk-3_adk1n-F
adk-3_adk2n-R
adk-66_adk1n-F
adk-66_adk2n-R
atp-29_atpA1n-F
atp-29_atpA2n-R
atp-3_atpA1n-F
atp-3_atpA2n-R
atp-66_atpA1n-F
atp-66_atpA2n-R
...
```

In this example, we wish to group the sequences into three groups based on the number after the "-" and before the "_" (i.e. 29, 3 and 66). The simple splitting as shown in figure 18.7 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 18.8. 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.

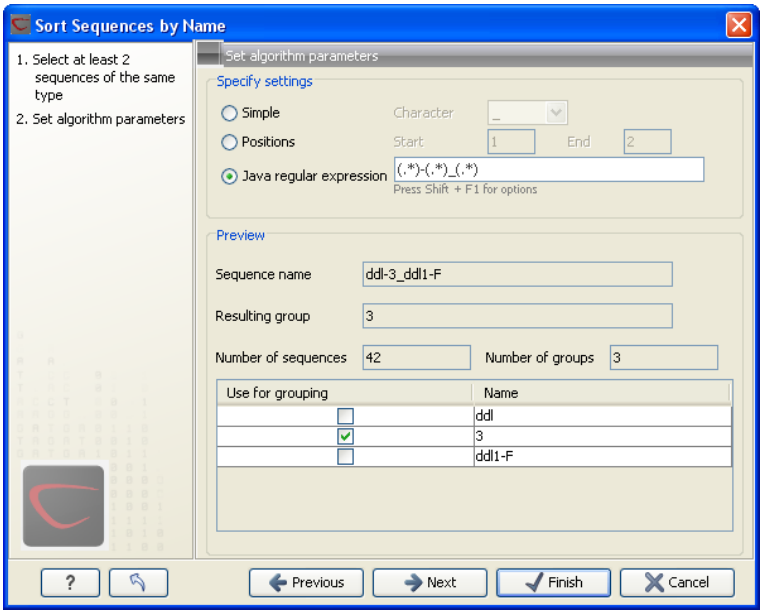



Figure 18.8: *Dividing the sequence into three groups based on the number in the middle of the name.*

## 18.5 Assemble sequences to reference

This section describes how to assemble a number of sequence reads into a contig using a reference sequence. A reference sequence can be particularly helpful when the objective is to characterize SNP variation in the data.

To start the assembly:

> **Toolbox** | **Molecular Biology Tools ( )** | **Sequencing Data 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.

**Note!** You can assemble a maximum of 2000 sequences at a time.

To assemble more sequences, please use the **Map Reads to Reference** ( ) under **NGS Core Tools ( )** in the **Toolbox**.

When the sequences are selected, click **Next**, and you will see the dialog shown in figure 18.9

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.

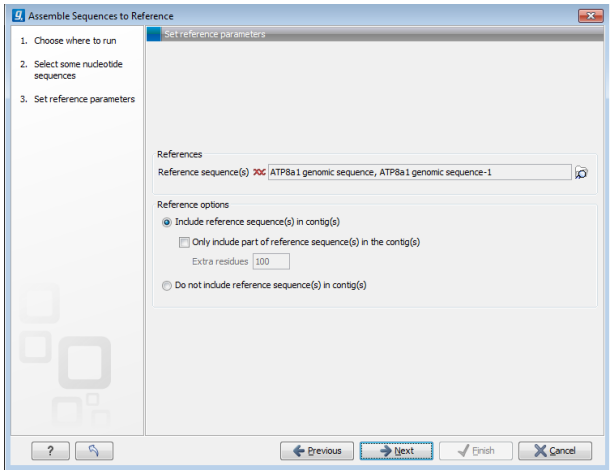



Figure 18.9: *Parameters for how the reference should be handled when assembling sequences to a reference sequence.*

  - **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 18.10

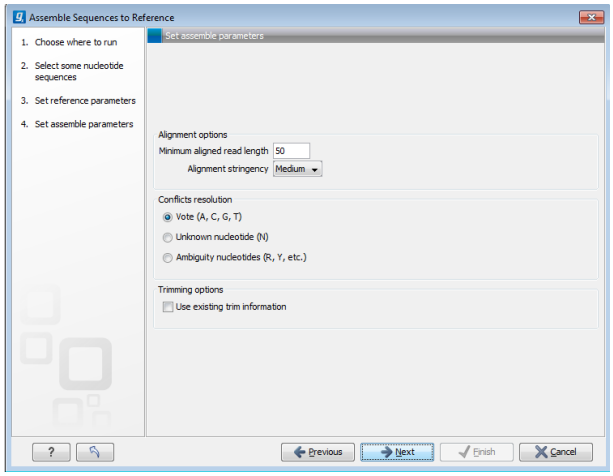



Figure 18.10: *Options for how the input sequences should be aligned and how nucleotide conflicts should be handled.*

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 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. Three stringency levels can be set:

  - **Low.**
  - **Medium.**
  - **High.**

  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 I.
  - **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 18.2 for more information about trimming).

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**. This will start the assembly process. See section 18.7 on how to use the resulting contigs.

## 18.6 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** ( ) | **Sequencing Data 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 18.2.1).

When the elements are selected, click **Next**, and you will see the dialog shown in figure 18.11

The options in this dialog are similar to the options that are available when assembling to a reference sequence (see section 18.5).

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**. This will start the assembly process. See section 18.7 on how to use the resulting contig.

Note that the new sequences will be added to the existing contig which will not be extended. If the new sequences extend beyond the existing contig, they will be cut off.

## 18.7 View and edit read mappings

The result of the mapping process is one or more read mappings where the sequence reads have been aligned (see figure 18.12). If multiple reference sequences were used, this information will

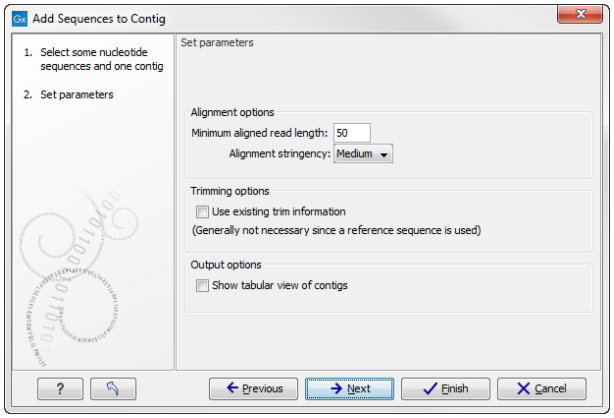



Figure 18.11: *Setting assembly parameters when assembling to an existing contig.*

be in a table where the actual visual mapping can be opened by double-clicking.

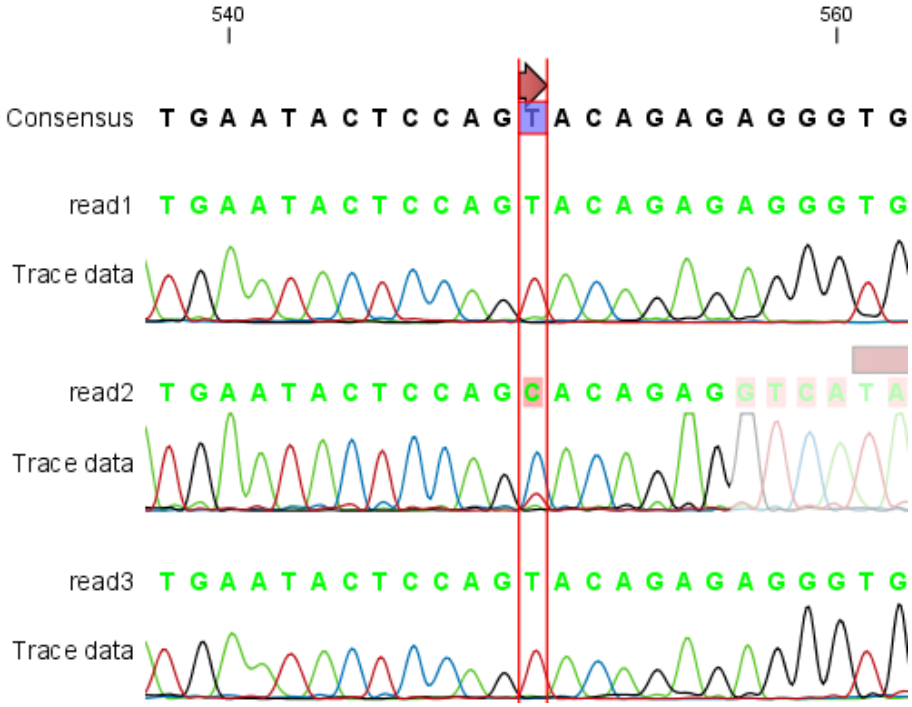



Figure 18.12: *The view of a read mapping. Notice that you can zoom to a very detailed level in read mappings.*

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 mapping. This may be due to trimming before or during the assembly or due to misalignment to the other reads.

You can easily adjust the trimmed area to include more of the read in the mapping: simply drag the edge of the faded area as shown in figure 18.13.



Figure 18.13: *Dragging the edge of the faded area.*

**Note!** This is only possible when you can see the residues on the reads. This means that you need to have zoomed in to 100% or more and chosen **Compactness** levels "Not compact", "Low" or "Packed". Otherwise the handles for dragging are not available (this is done in order to make the visual overview more simple).

If reads have been reversed, this is indicated by red. Otherwise, the residues are colored green. The colors can be changed in the **Side Panel** as described in section 25.5.2

If you find out that the reversed reads should have been the forward reads and vice versa, you can reverse complement the whole mapping (imagine flipping the whole mapping):

**right-click in the empty white area of the mapping | Reverse Complement**

### 18.7.1 View settings in the Side Panel

Apart from this the view resembles that of alignments (see section 20.2) but has some extra preferences in the **Side Panel**:

1

- **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 section 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 18.1.2 and section 10.1.
    - **Low.** Hides trace data, quality scores and puts the reads' annotations on the sequence.
    - **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 25.25.

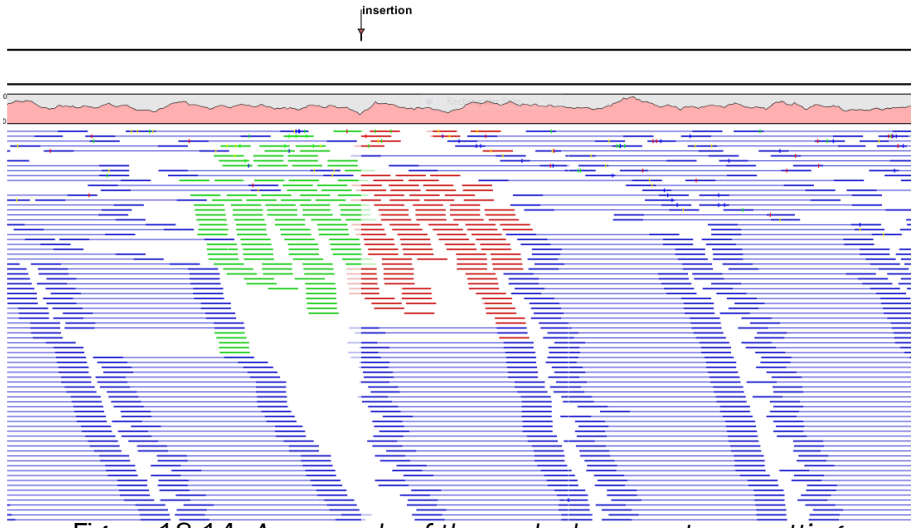



Figure 18.14: *An example of the packed compactness setting.*

- **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 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 horizontal lines in. 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). E.g. 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

[1]Note that for interpretation of mappings with large amounts of data, have a look at section 25.5

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.

- **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 6.6).
      - **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.** At the top of the **Side Panel**:
  - **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 mapping, which can be adjusted in the View settings. Those covered here were the key ones relevant for standard review of mapping results.

### 18.7.2 Editing the read mapping

When editing 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 7) 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 mapping view, you can use **Zoom in** (  ) to zoom to a greater level of detail than in other views (see figure 18.12). 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 mappings with more than 1000 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.

### 18.7.3 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.

### 18.7.4 Read conflicts

When the mapping is created, 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 have a different residue*.

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**.

### 18.7.5 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 25.8).

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 e.g. for Primer design ( ).

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**.

### 18.7.6 Extract parts of a mapping

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. You are now presented with the dialog shown in Figure 18.15.

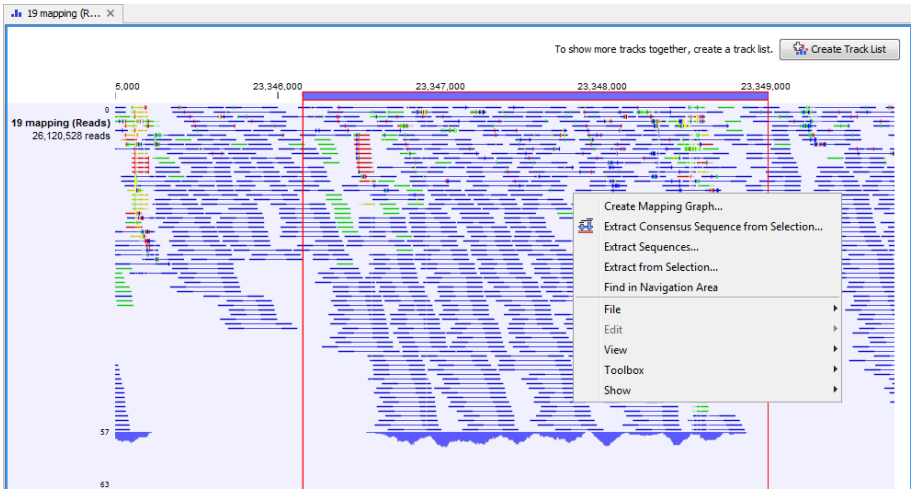



Figure 18.15: *Extracting parts of a mapping.*

When you choose the **Extract from Selection** option you are presented by the dialog shown in figure 18.16.

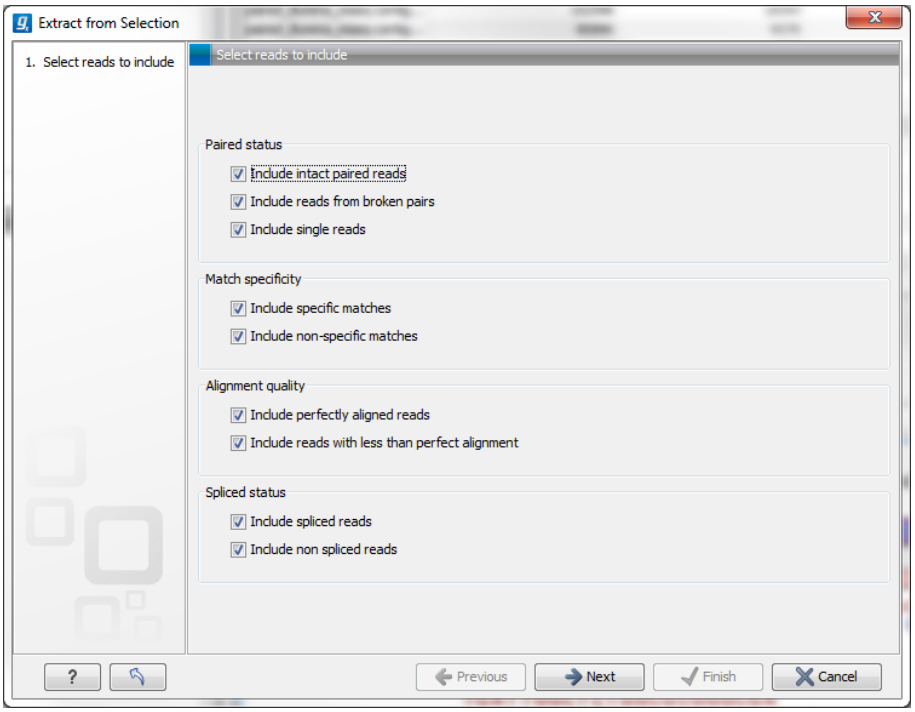



Figure 18.16: *Selecting the reads to include.*

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.

**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 14.2)

You will now have all paired reads from the original mapping in a list.

### 18.7.7 Variance table

In addition to the standard graphical display of a 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 18.17.

The table has the following columns:

- **Reference position.** The position of the conflict measured from the starting point of the reference sequence.

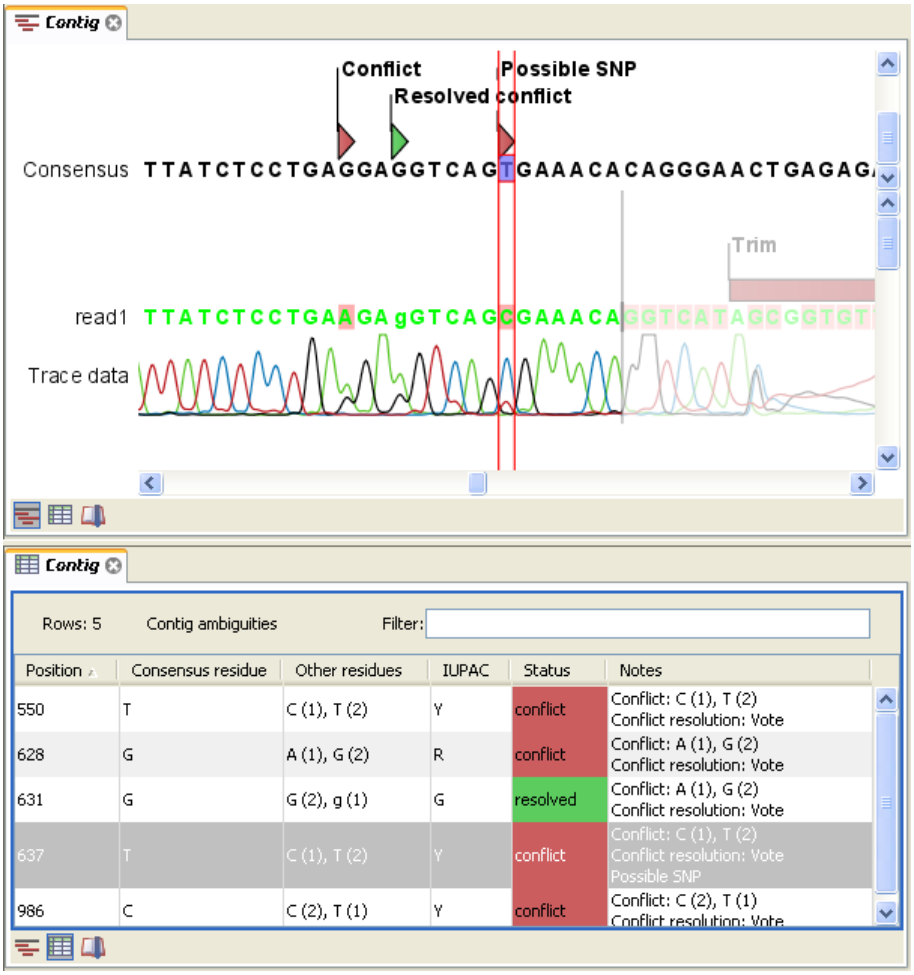



Figure 18.17: *The graphical view is displayed at the top. At the bottom the conflicts are shown in a table. At the conflict at position 637, the user has entered a comment in the table. This comment is now also reflected on the tooltip of the conflict annotation in the graphical view above.*

- **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 18.17, 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 I.)
- **Status.** The status can either be conflict or resolved:
  - **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 18.17). 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 mapping, apart from 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.

## 18.8 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 ( ) | Sequencing Data Analysis ( )| Reassemble Contig ( ) | select the contig from Navigation Area, move to 'Selected Elements' and click Next**

or **right-click in the empty white area of the contig | Reassemble contig ( )**

This opens a dialog as shown in figure 18.18

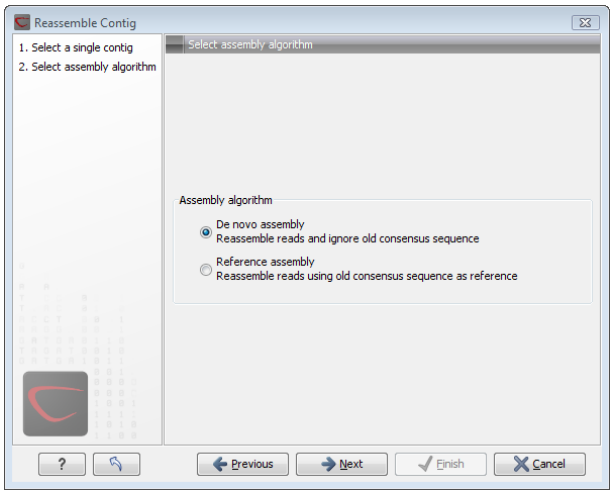



Figure 18.18: *Re-assembling a contig.*

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 18.3. The consensus sequence of the contig will be ignored.
- **Reference assembly**. This will use the consensus sequence of the contig as reference. When you click **Next**, you will follow the same steps as described in section 18.5.

When you click **Finish**, a new contig is created, so you do not lose the information in the old contig.

## 18.9 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 18.2).

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 ( ) | Sequencing Data 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 18.19.

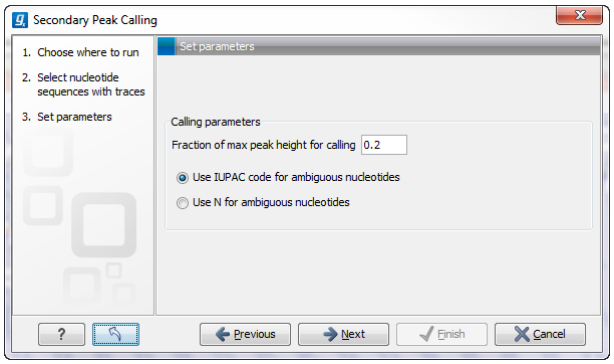



Figure 18.19: *Setting parameters secondary peak calling.*

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 I).

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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**. 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.

# Chapter 19

# Cloning and cutting

## Contents


**19.1 Molecular cloning . . . 397**
19.1.1 Introduction to the cloning editor . . . 398
19.1.2 The cloning workflow . . . 399
19.1.3 Manual cloning . . . 402
19.1.4 Insert restriction site . . . 407
**19.2 Gateway cloning . . . 407**
19.2.1 Add attB sites . . . 408
19.2.2 Create entry clones (BP) . . . 413
19.2.3 Create expression clones (LR) . . . 415
**19.3 Restriction site analysis . . . 416**
19.3.1 Dynamic restriction sites . . . 417
19.3.2 Restriction site analysis from the Toolbox . . . 424
**19.4 Gel electrophoresis . . . 430**
19.4.1 Separate fragments of sequences on gel . . . 430
19.4.2 Separate sequences on gel . . . 431
19.4.3 Gel view . . . 431
**19.5 Restriction enzyme lists . . . 432**
19.5.1 Create enzyme list . . . 433
19.5.2 View and modify enzyme list . . . 434


*CLC Genomics Workbench* offers graphically advanced *in silico* cloning and design of vectors for various purposes together with restriction enzyme analysis and functionalities for managing lists of restriction enzymes.

First, after a brief introduction, restriction cloning and general vector design is explained. Next, we describe how to do Gateway Cloning [1]. Finally, the general restriction site analyses are described.

[1] Gateway is a registered trademark of Invitrogen Corporation

## 19.1 Molecular cloning

Molecular cloning is a very important tool in the quest to understand gene function and regulation. Through molecular cloning it is possible to study individual genes in a controlled environment. Using molecular cloning it is possible to build complete libraries of fragments of DNA inserted into appropriate cloning vectors.

The *in silico* cloning process in *CLC Genomics Workbench* begins with the selection of sequences to be used:

**Toolbox** | **Molecular Biology Tools** ( ) | **Cloning and Restriction Sites** ( )| **Cloning** ( )

This will open a dialog where you can select the sequences containing the fragments you want to clone as well as sequences to be used as vector (figure 19.1).

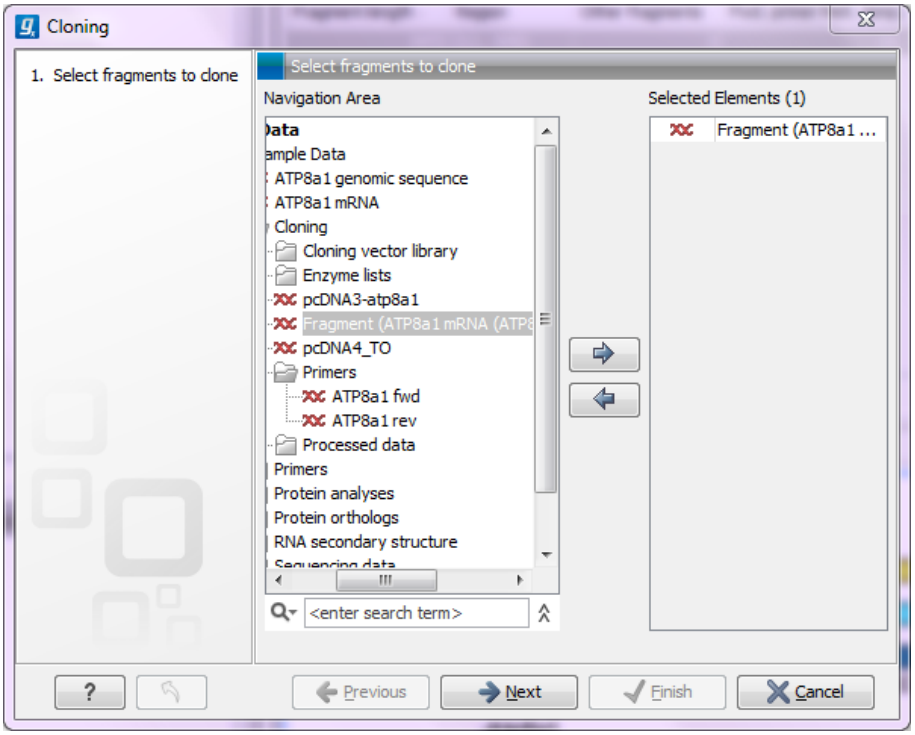



Figure 19.1: *Selecting one or more sequences containing the fragments you want to clone.*

The *CLC Genomics Workbench* will now create a sequence list of the selected fragments and vector sequences (if you have selected both fragments and vectors) and open it in the cloning editor as shown in figure 19.2.

When you save the cloning experiment, it is saved as a **Sequence list**. See section 10.6 for more information about sequence lists. If you need to open the list later for cloning work, simply switch to the **Cloning** ( ) editor at the bottom of the view.

If you later in the process need additional sequences, you can easily add more sequences to the view. Just:

**right-click anywhere on the empty white area** | **Add Sequences**

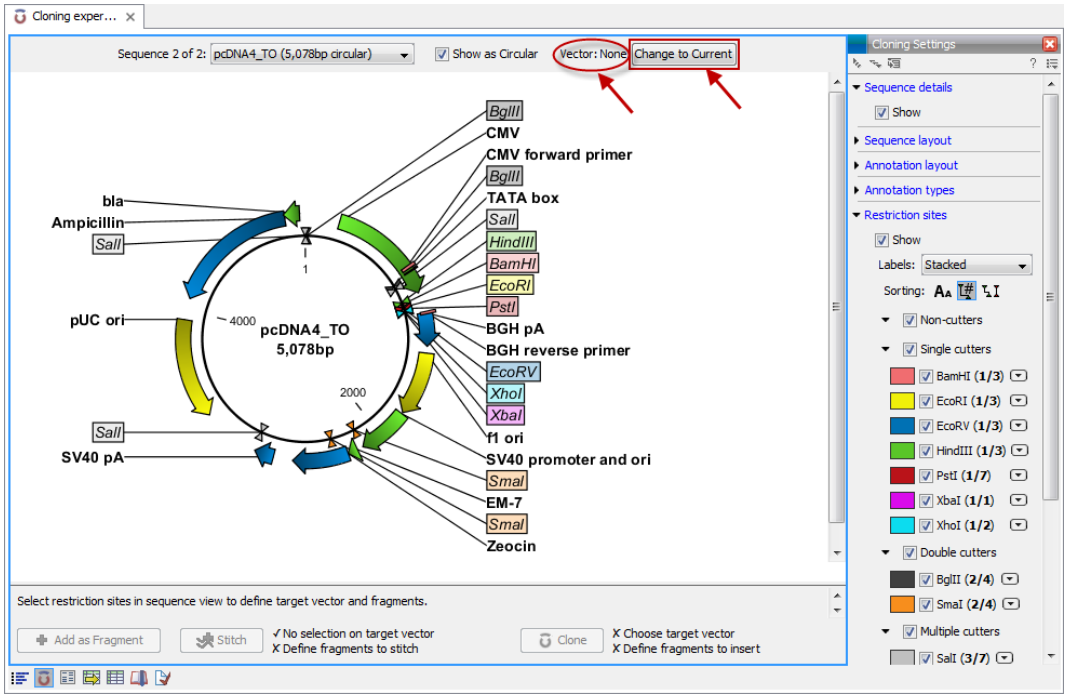



Figure 19.2: *Cloning editor.*

### 19.1.1 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. See section 10.1 for an explanation of these options. This means that e.g. known SNP's, exons and other 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. At the right-hand side, you can select whether or not to select a vector. When no vector has been selected a button **Change to Current** is enabled. This button can be used 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. This is explained in details below.
- At the bottom, there is a panel where the selection of fragments and target vector is performed (see elaboration below).

There are essentially three ways of performing cloning in the *CLC Genomics Workbench*. The *first* is the most straight-forward approach, which is based on a simple model of selecting restriction sites for cutting out one or more fragments and defining how to open the vector to insert the fragments. This is described as *the cloning workflow* below. The *second* approach is unguided and more flexible and allows you to manually cut, copy, insert and replace parts of the sequences. This approach is described under *manual cloning* below. *Finally*, the *CLC Genomics Workbench* also supports *Gateway cloning* (see section 19.2).

#### The cloning editor

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:

1. **The cloning mode** is utilized 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.
2. **The stitch mode** is utilized when the user deselects or 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.

### 19.1.2 The cloning workflow

The *cloning workflow* is designed to support restriction cloning workflows through the following steps:

1. Define one or more fragments
2. Define how the vector should be opened
3. Specify orientation and order of the fragment

#### Defining 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 19.3.1). 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.

When this is done, the panel below will update to reflect the selections (see figure 19.3).

In this example you can see that there are now two options listed in the panel below the view. This is because there are now two options for selecting the fragment that should be used for cloning. 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 the **Add Current Sequence as Fragment** ( ).

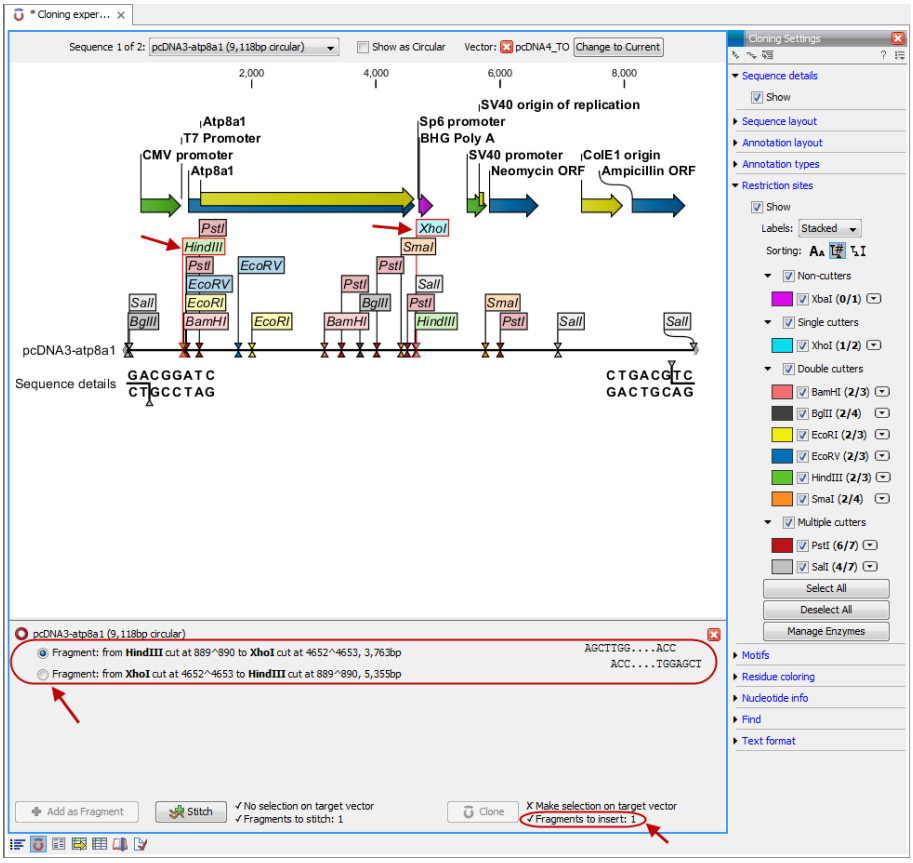



Figure 19.3: *HindIII and XhoI cut sites selected to cut out fragment.*

At any time, the selection of cut sites can be cleared by clicking the **Remove** ( ) icon to the right of the fragment selections. 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**.

### Defining target vector

When selecting among the sequences in the panel at the top, the vector sequence has "vector" appended to its name. If you wish to use one of the other sequences as vector, select this sequence in the list and click **Change to Current**.

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. 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) (figure 19.4).

Just as when cutting out the fragment, there is a lot of choices regarding which sequence should be used as the vector.

At any time, the selection of cut sites can be cleared by clicking the **Remove** ( ) icon to the right of the target vector selections. 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**.

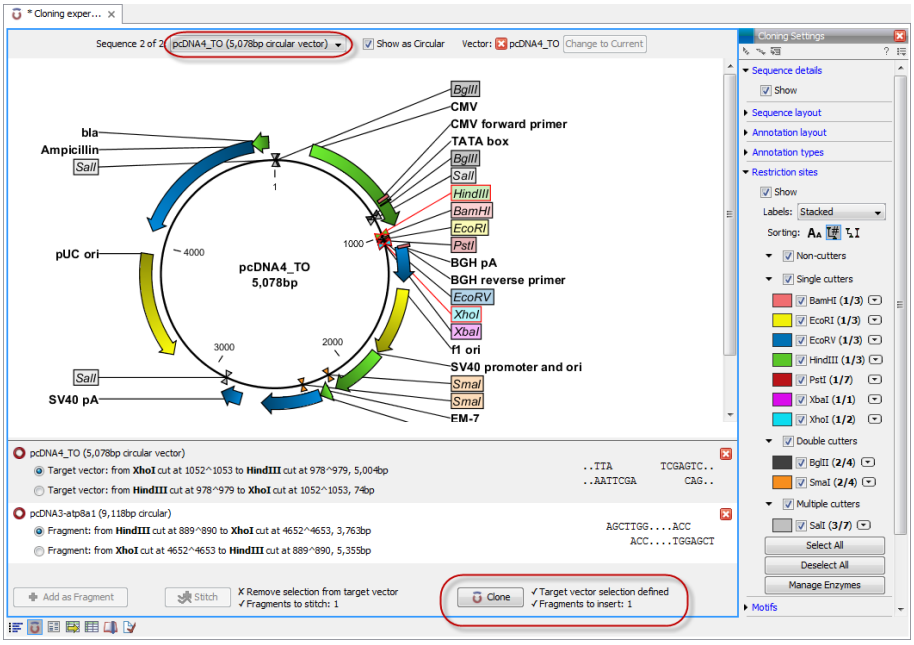



Figure 19.4: *HindIII and XhoI sites used to open the vector. Note that the "Cloning" button has now been enabled as both criteria ("Target vector selection defined" and "Fragments to insert:...") have been defined.*

When the right target vector is selected, you are ready to **Perform Cloning** ( ), see below.

### Perform cloning

Once selections have been made for both fragments and vector, click **Cloning** ( ). This will display a dialog to adapt overhangs and change orientation as shown in figure 19.5)

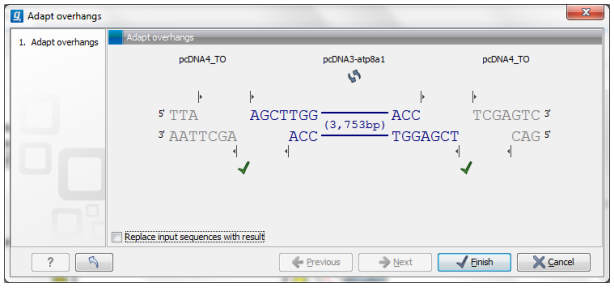



Figure 19.5: *Showing the insertion point of the vector.*

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 can blunt end or fill in the overhangs using the **drag handles** ( ). Click and drag with the mouse to adjust the overhangs.

Whenever you drag the handles, the status of the insertion point is indicated below:

- The overhangs match ( ).
- The overhangs do not match ( ). In this case, you will not be able to click **Finish**. Drag the handles to make 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 ( )/ ( ).

There is an option for the result of the cloning: **Replace input sequences with result**. Per default, the construct will be opened in a new view and can be saved separately. By selecting this option, the construct will also be added to the input sequence list and the original fragment and vector sequences will be deleted.

When you click **Finish** the final construct will be shown (see figure 19.6).

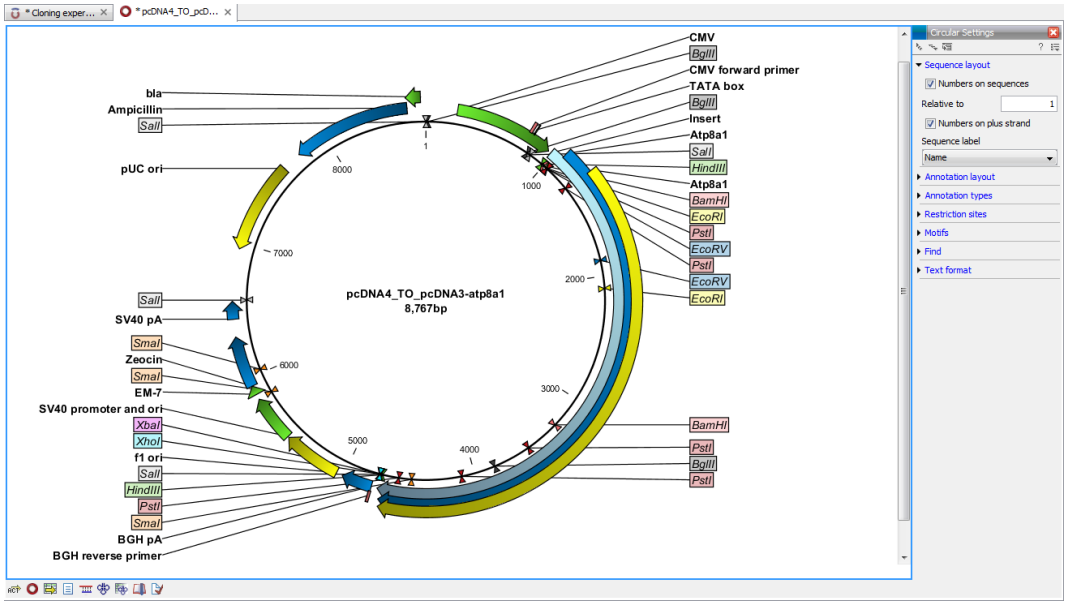



Figure 19.6: *The final construct.*

You can now **Save** ( ) this sequence for later use. 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.

### 19.1.3 Manual cloning

If you wish to use the manual way of cloning (as opposed to using the cloning workflow explained above in section 19.1.2), you can disregard the panel at the bottom. The manual cloning approach is based on a number of ways that you can manipulate the sequences. All manipulations of sequences are done manually, giving you full control over how the final construct is made. Manipulations are performed through right-click menus, which have three different appearances depending on where you click, as visualized in figure 19.7.

- **Right-click the sequence name (to the left) to manipulate the whole sequence**.
- **Right-click a selection to manipulate the selection**.

The two menus are described in the following:

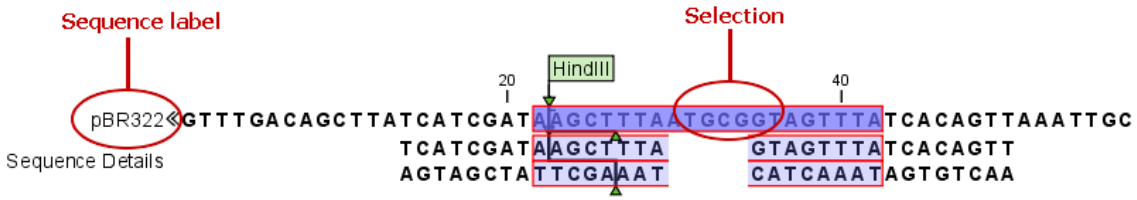



Figure 19.7: *The red circles mark the two places you can use for manipulating the sequences.*

### Manipulate the whole sequence

Right-clicking the sequence name at the left side of the view reveals several options on sorting, opening and editing the sequences in the view (see figure 19.8).

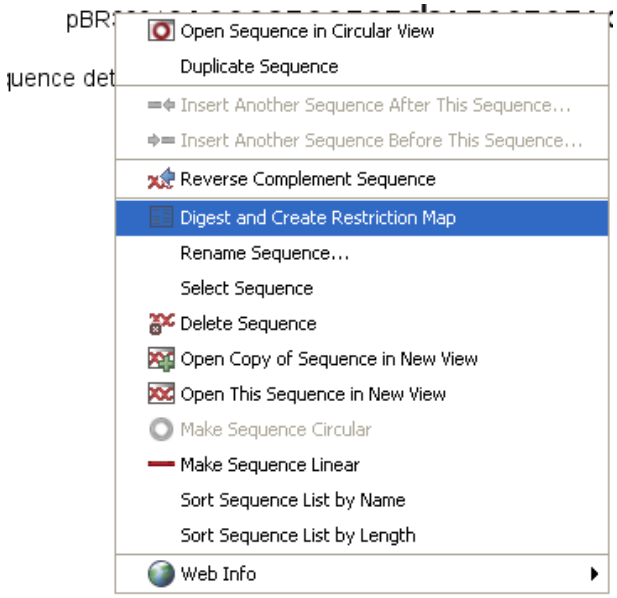



Figure 19.8: *Right click on the sequence in the cloning view.*

- **Open sequence in circular view ( )**
  Opens the sequence in a new circular view. If the sequence is not circular, you will be asked if you wish to make it circular or not. (This will not forge ends with matching overhangs together - use "Make Sequence Circular" ( ) instead.)

- **Duplicate sequence**
  Adds a duplicate of the selected sequence. The new sequence will be added to the list of sequences shown on the screen.

- **Insert sequence after this sequence ( )**
  Insert another sequence after this sequence. The sequence to be inserted can be selected from a list which contains the sequences present in the cloning editor. 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. Otherwise a warning is displayed.

- **Insert sequence before this sequence ( )**
  Insert another sequence before this sequence. The sequence to be inserted can be selected from a list which contains the sequences present in the cloning editor. 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. Otherwise a warning is displayed.

- **Reverse sequence**
  Reverse 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* (see the following item in the list).
- **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 Sequence with Selected Enzymes and Run on Gel ( )**
  See section 19.4.1
- **Rename sequence**
  Renames the sequence.
- **Select sequence**
  This will select the entire sequence.
- **Delete sequence ( )**
  This deletes the given sequence from the cloning editor.
- **Open sequence ( )**
  This will open the selected sequence in a normal sequence view.
- **Make sequence circular** ( )
  This will convert 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 ( )**
  This will convert a sequence from a circular to a linear form, removing the << and >> at the ends.

#### Manipulate parts of the sequence

Right-clicking a selection reveals several options on manipulating the selection (see figure 19.9).

- **Duplicate Selection.** If a selection on the sequence is duplicated, the selected region will be added as a new sequence to the cloning editor with a new sequence name representing the length of the fragment. When a sequence region between two restriction sites are double-clicked the entire region will automatically be selected. This makes it very easy to make a new sequence from a fragment created by cutting with two restriction sites (right-click the selection and choose **Duplicate selection**).
- **Replace Selection with sequence.** This will replace the selected region with a sequence. The sequence to be inserted can be selected from a list containing all sequences in the cloning editor.

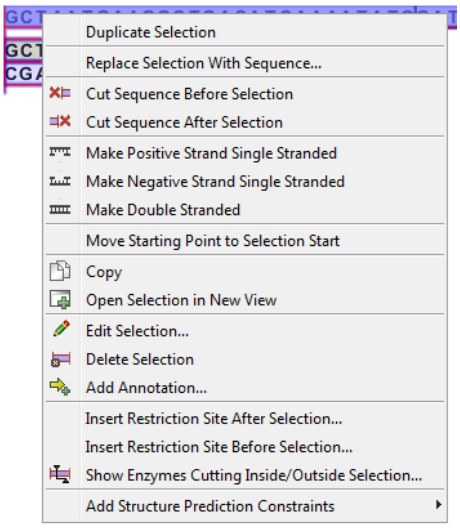



Figure 19.9: *Right click on a sequence selection in the cloning view.*

- **Cut Sequence Before Selection ( ).** This will cleave the sequence before the selection and will result in two smaller fragments.
- **Cut Sequence After Selection ( ).** This will cleave the sequence after the selection and will result in two smaller fragments.
- **Make Positive Strand Single Stranded ( ).** This will make the positive strand of the selected region single stranded.
- **Make Negative Strand Single Stranded ( ).** This will make 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 ( ).** This will copy the selected region to the clipboard, which will enable it for use in other programs.
- **Open Selection in New View ( ).** This will open the selected region in the normal sequence view.
- **Edit Selection ( ).** This will open a dialog box, in which is it possible to edit the selected residues.
- **Delete Selection ( ).** This will delete the selected region of the sequence.
- **Add Annotation ( ).** This will open the **Add annotation** dialog box.
- **Insert Restriction Sites After/Before Selection.** This will show a dialog where you can choose from a list restriction enzymes (see section 19.1.4).
- **Show Enzymes Cutting Inside/Outside Selection ( ).** This will add enzymes cutting this selection to the Side Panel.
- **Add Structure Prediction Constraints.** This is relevant for RNA secondary structure prediction (see section 22.1.4).

### Insert one sequence into another

Sequences can be inserted into each other in several ways as described in the lists above. When you chose to insert one sequence into another you will be presented with a dialog where all sequences in the view are present (see figure 19.10).

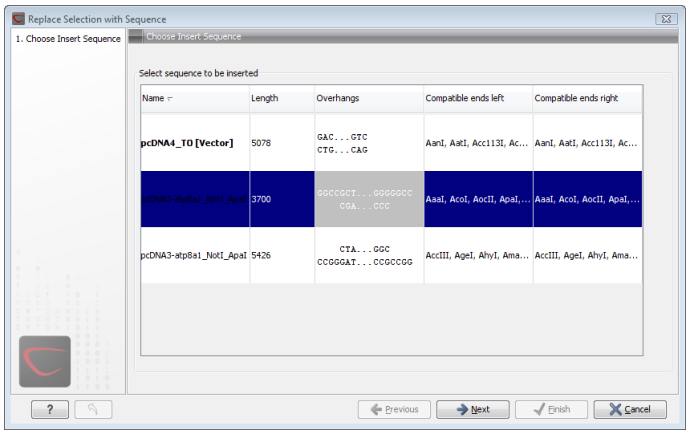



Figure 19.10: *Select a sequence for insertion.*

The sequence that you have chosen to insert into will be marked with **bold** and the text **[vector]** is appended to the sequence name. Note that this is completely unrelated to the vector concept in the cloning workflow described in section 19.1.2.

The list furthermore includes the length of the fragment, an indication of the overhangs, and a list of enzymes that are compatible with this overhang (for the left and right ends, respectively). If not all the enzymes can be shown, place your mouse cursor on the enzymes, and a full list will be shown in the tool tip.

Select the sequence you wish to insert and click **Next**.

This will show the dialog in figure 19.11).

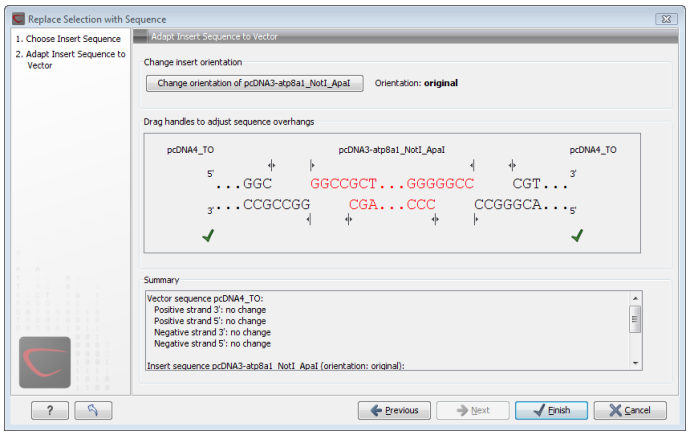



Figure 19.11: *Drag the handles to adjust overhangs.*

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** ( ).

Whenever you drag the handles, the status of the insertion point is indicated below:

- The overhangs match ( ).
- The overhangs do not match ( ). In this case, you will not be able to click **Finish**. Drag the handles to make the overhangs match.

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 ( ) when you click **Finish**.

When you click **Finish** and the sequence is inserted, it will be marked with a selection.

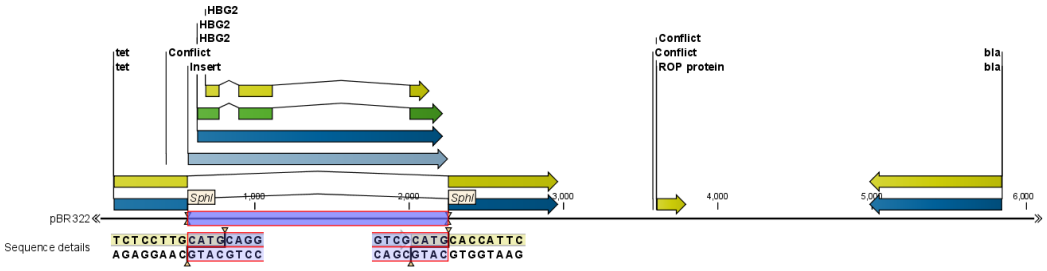



Figure 19.12: *One sequence is now inserted into the cloning vector. The sequence inserted is automatically selected.*

### 19.1.4 Insert restriction site

If you make a selection on the sequence, right-click, 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 19.13

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 etc., this can be typed into the text fields adjacent to the recognition sequence. .

Click **OK** will insert the sequence 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. Furthermore, an restriction site annotation is added.

## 19.2 Gateway cloning

*CLC Genomics Workbench* offers tools to perform *in silico* Gateway cloning[2], including Multi-site Gateway cloning.

The three tools for doing Gateway cloning in the *CLC Genomics Workbench* mimic the procedure followed in the lab:

[2]Gateway is a registered trademark of Invitrogen Corporation

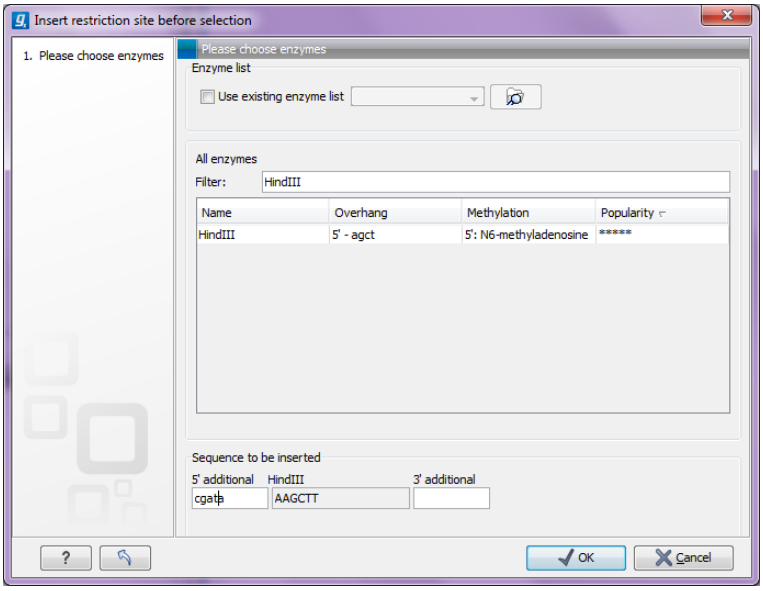



Figure 19.13: *Inserting the HindIII recognition sequence.*

- 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.invitrogen.com/site/us/en/home/Products-and-Services/Applications/Cloning/Gateway-Cloning.html`

To perform these analyses in the *CLC Genomics Workbench*, you need to import donor and expression vectors. These can be downloaded from Invitrogen's web site and directly imported into the Workbench: `http://tools.invitrogen.com/downloads/Gateway%20vectors.ma4`

### 19.2.1 Add attB sites

The first step in the Gateway cloning process is to amplify the target sequence with primers including so-called attB sites. In the *CLC Genomics Workbench*, you can add attB sites to a sequence fragment in this way:

**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 first. This can be done in two ways:

- If the fragment is covered by an annotation (if you want to use e.g. a CDS), simply right-click the annotation and **Open Annotation in New View**

- Otherwise you can simply make a selection on the sequence, right-click and **Open Selection in New View**

In both cases, the selected part of the sequence will be copied and opened as a new sequence which can be **Saved** ( ).

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 19.14.

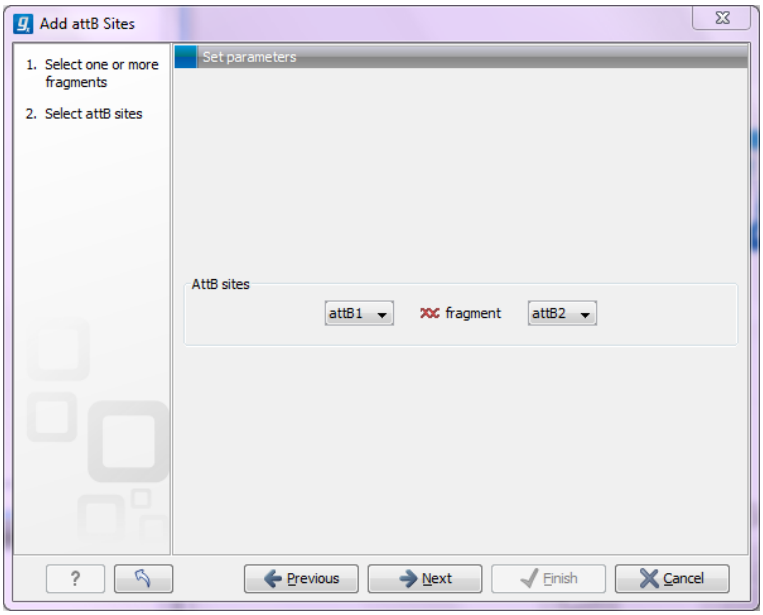



Figure 19.14: *Selecting which attB sites to add.*

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.

Click **Next** will give you options 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). See an example of this in figure 19.20 where a Shine-Dalgarno site has been added between the attB site and the gene of interest.

At the top of the dialog (see figure 19.15), you can specify primer additions such as a Shine-Dalgarno site, start codon etc. Click in the text field and press **Shift + F1 (Shift + Fn + F1 on Mac)** to show some of the most common additions (see figure 19.16).

Use the up and down arrow keys to select a tag and press **Enter**. This will insert the selected sequence as shown in figure 19.17.

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 (i.e. the sequence you selected as input). 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 (like the green Shine-Dalgarno site in figure 19.17).

This default list of primer additions can be modified, see section 19.2.1.

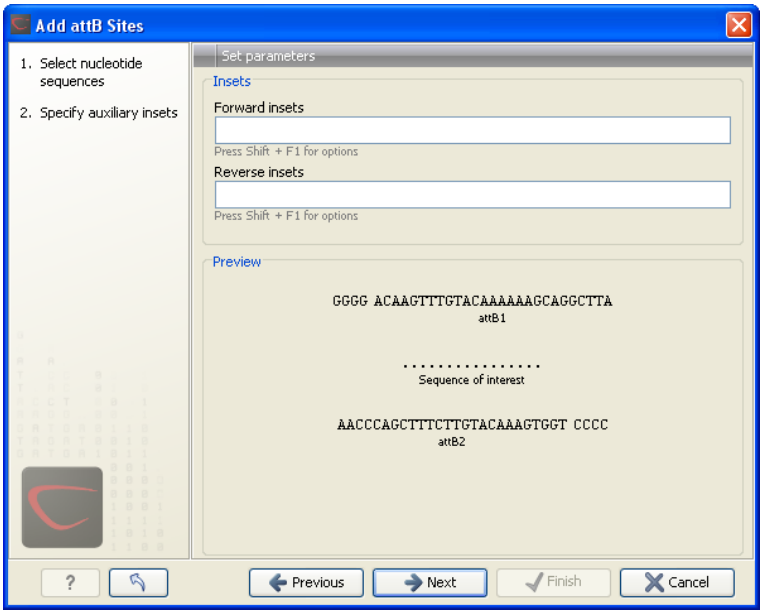



Figure 19.15: *Primer additions 5' of the template-specific part of the primer.*

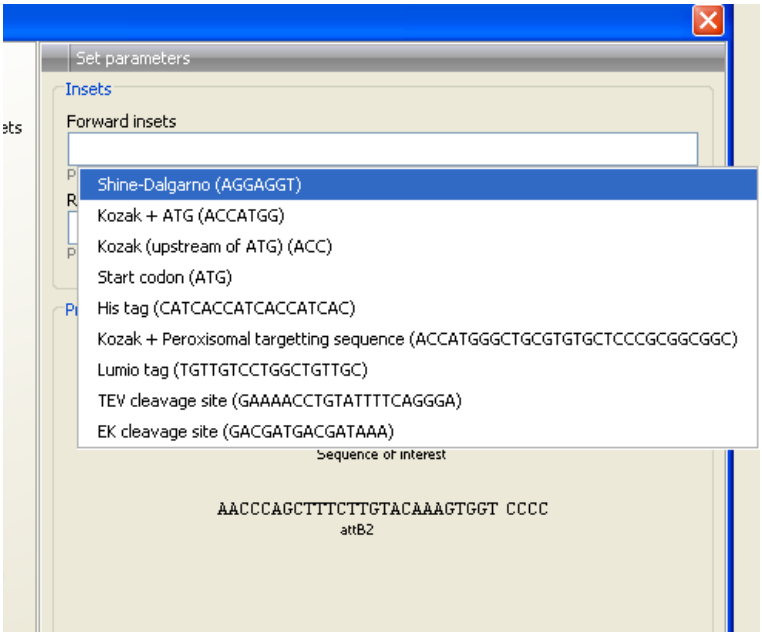



Figure 19.16: *Pressing Shift + F1 shows some of the common additions. This default list can be modified, see section 19.2.1.*

You can also manually type a sequence with the keyboard or paste in a sequence from the clipboard by pressing **Ctrl + v (⌘ + v on Mac)**.

Clicking **Next** allows you to specify the length of the template-specific part of the primers as shown in figure 19.18.

The *CLC Genomics 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 and you can also get a list of primers as you can see when clicking **Next** (see figure 19.19.

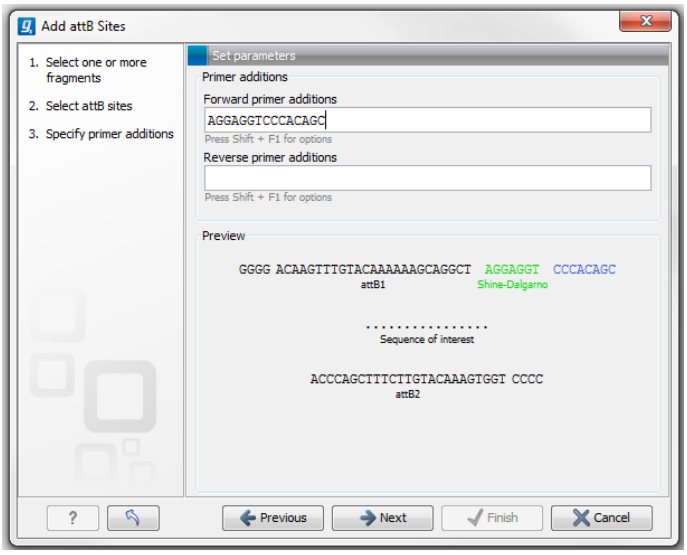



Figure 19.17: *A Shine-Dalgarno sequence has been inserted.*

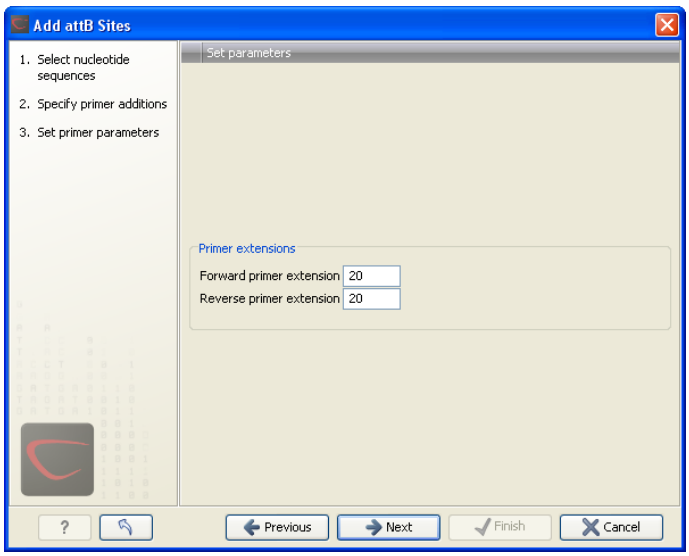



Figure 19.18: *Specifying the length of the template-specific part of the primers.*

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. Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

The attB sites, the primer additions and the primer regions are annotated in the final result as shown in figure 19.20.

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 work flow (see section 19.2.2). When you open the sequence again, you may need to switch on the relevant annotation types to show the sites and primer additions as illustrated in figure 19.20.

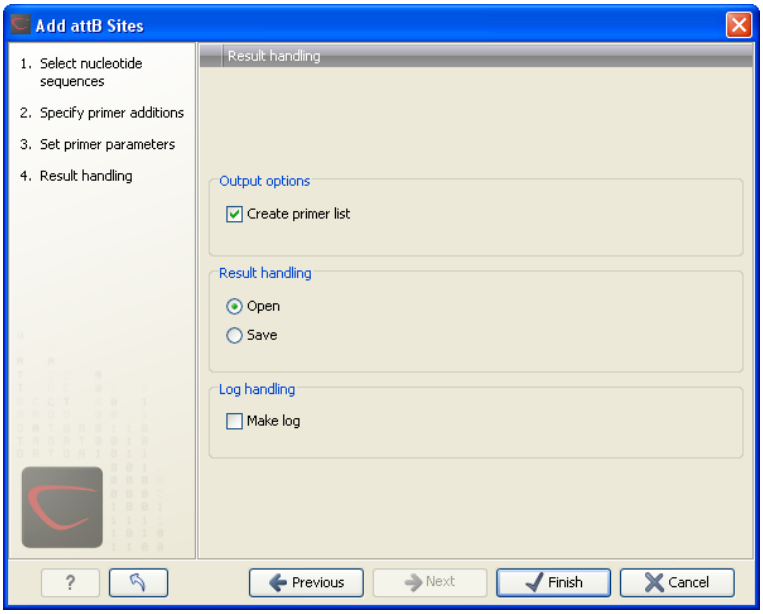



Figure 19.19: *Besides the main output which is a copy of the input sequence(s) now including attB sites and primer additions, you can get a list of primers as output.*

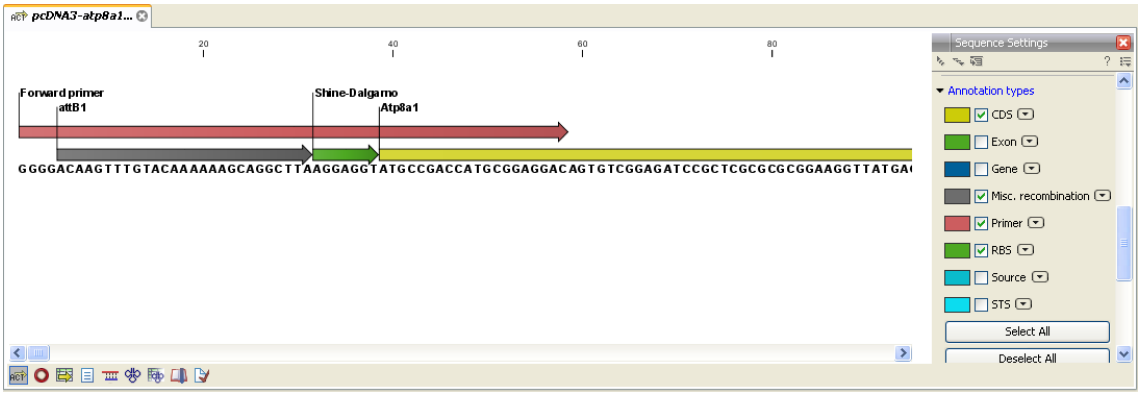



Figure 19.20: *the attB site plus the Shine-Dalgarno primer addition is annotated.*

### 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 19.15 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** | **Advanced**

In the advanced preferences dialog, scroll to the part called **Gateway cloning primer additions** (see figure 19.21).

Each element in the list has the following information:

**Name** The name of the sequence. 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. The sequence is always defined on the sense strand (although the reverse primer would be reverse complement).

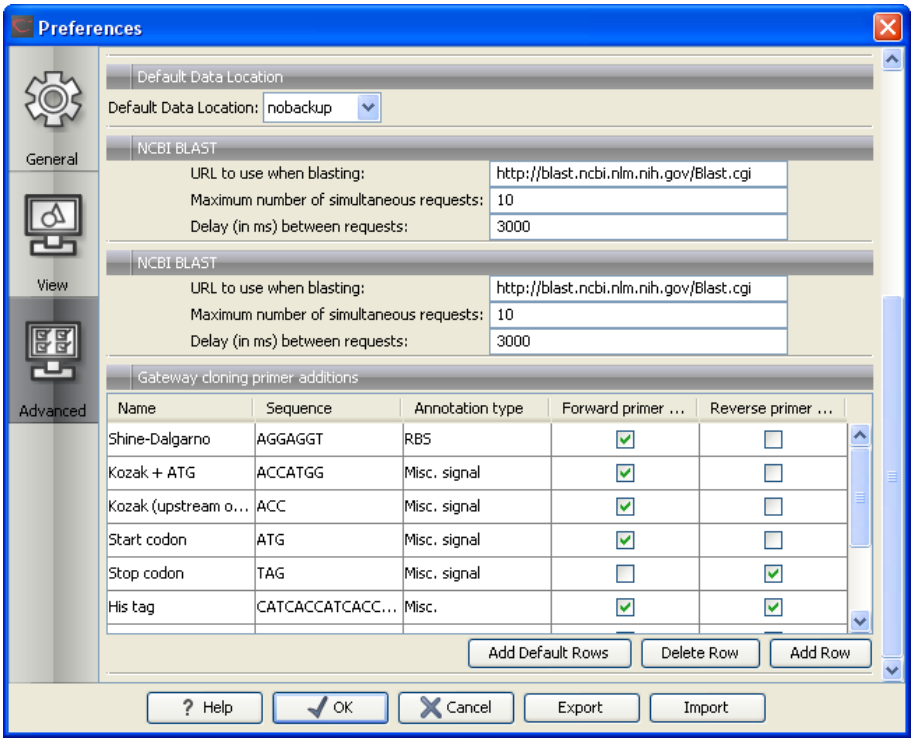



Figure 19.21: *Configuring the list of primer additions available when adding attB sites.*

**Annotation type** The annotation type used for the annotation that is added to the fragment.

**Forward primer addition** Whether this addition should be visible in the list of additions for the forward primer.

**Reverse primer addition** Whether this addition should be visible in the list of additions for the reverse primer.

You can either change 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.

### 19.2.2 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, the so-called BP reaction:

> **Toolbox** | **Molecular Biology Tools** ( ) | **Cloning and Restriction Sites** ( ) | **Gateway Cloning** ( ) | **Create Entry Clone** ( )

This will open a dialog where you can select one or more sequences that will be the sequence of interest to be recombined into your donor vector. Note that the sequences you select should be flanked with attB sites (see section 19.2.1). You can select more than one sequence as input, and the corresponding number of entry clones will be created.

When you have selected your sequence(s), click **Next**.

This will display the dialog shown in figure 19.22.

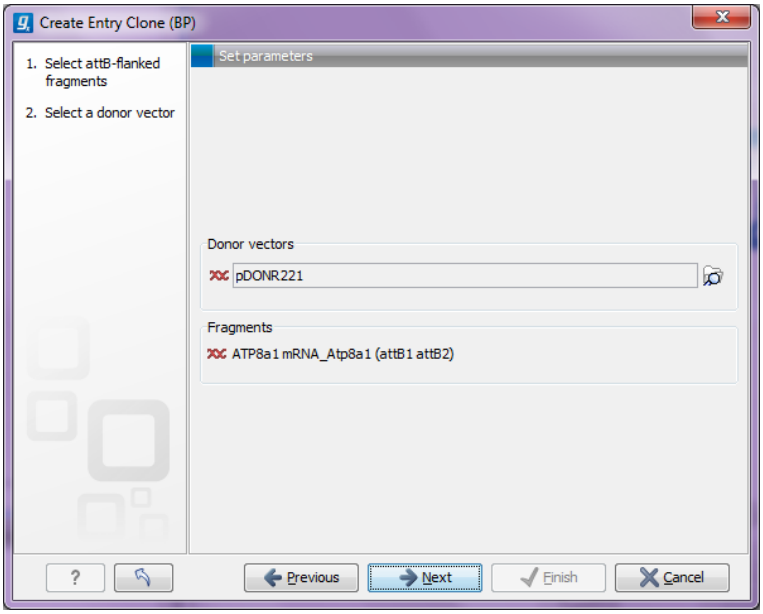



Figure 19.22: *Selecting one or more donor vectors.*

Clicking the **Browse** ( ) button opens a dialog where you can select a donor vector. You can download donor vectors from Invitrogen's web site: http://tools.invitrogen.com/downloads/Gateway%20vectors.ma4 and import into the *CLC Genomics Workbench*. Note that the Workbench looks for the specific sequences of the attP sites in the sequences that you select in this dialog (see how to change the definition of sites in appendix G). Note that the *CLC Genomics Workbench* only checks that valid attP sites are found - 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.

Below there is a preview of the fragments selected and the attB sites that they contain. 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.

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

The output is one entry clone per sequence selected. The attB and attP sites have been used for the recombination, and the entry clone is now equipped with attL sites as shown in figure 19.23.

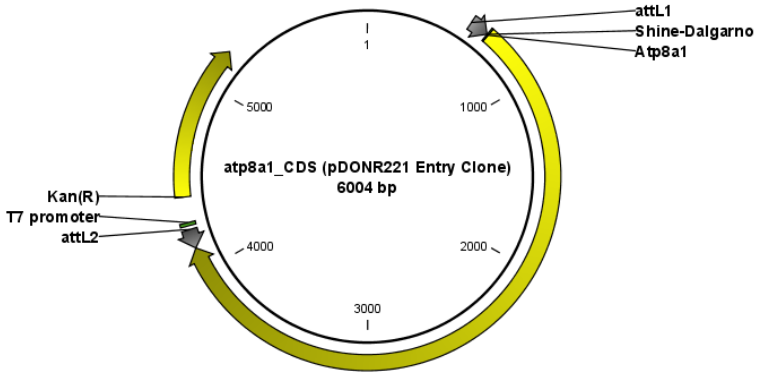



Figure 19.23: *The resulting entry vector opened in a circular view.*

Note that the bi-product of the recombination is not part of the output.

### 19.2.3 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, the so-called LR reaction:

**Toolbox | Molecular Biology Tools ( ) | Cloning and Restriction Sites ( )| Gateway Cloning ( ) | Create Expression Clone ( )**

This will open a dialog where you can select one or more entry clones (see how to create an entry clone in section 19.2.2). If you wish to perform separate LR reactions with multiple entry clones, you should run the **Create Expression Clone** in batch mode (see section 8.1).

When you have selected your entry clone(s), click **Next**.

This will display the dialog shown in figure 19.24.

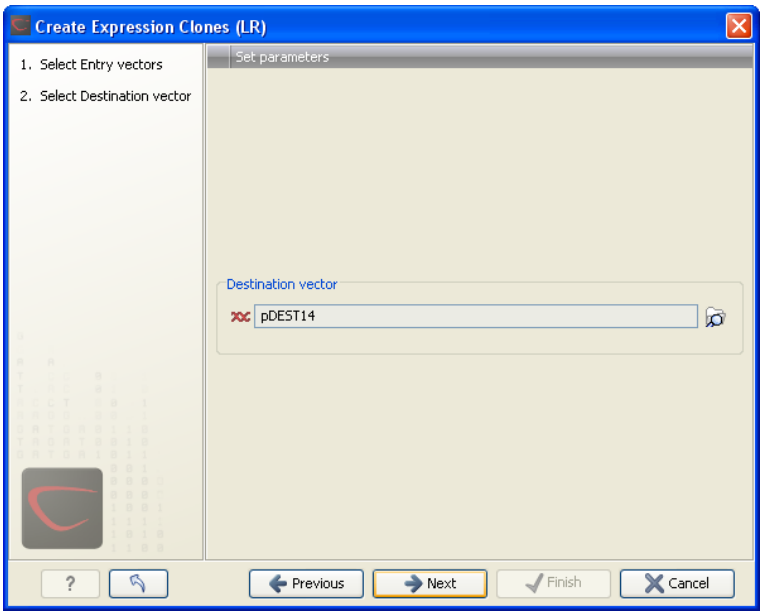



Figure 19.24: *Selecting one or more destination vectors.*

Clicking the **Browse** ( ) button opens a dialog where you can select a destination vector. You can download donor vectors from Invitrogen's web site: http://tools.invitrogen.com/downloads/Gateway%20vectors.ma4 and import into the *CLC Genomics Workbench*. 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 G). Note that the *CLC Genomics Workbench* only checks that valid attR sites are found - it does not check that they correspond to the attL sites of the selected fragments at this step. If the right combination of attL and attR sites is not found, no entry clones will be produced.

When performing multi-site gateway cloning, the *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 http://tools.invitrogen.com/downloads/gateway-multisite-seminar.html

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

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 19.25.

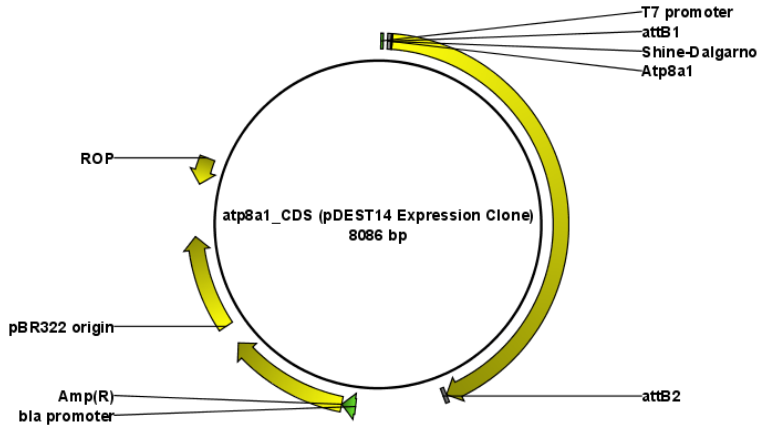



Figure 19.25: *The resulting expression clone opened in a circular view.*

You can choose to create a sequence list with the bi-products as well.

For a destination vector to be recognized, apart from the appropriate att sites, it must contain the *ccdB* gene. This 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 the relevant part of the sequence, right-click and choose 'Add Annotation'. Name the annotation 'ccdB'.

## 19.3 Restriction site analysis

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 will be useful, since it is a quick and easy way of showing restriction sites.
- In the **Toolbox** you will find the other way of doing restriction site analyses. This way provides more control of the analysis and gives you more output options, e.g. a table of restriction sites and you can perform the same restriction map analysis on several sequences in one step.

This chapter first describes the dynamic restriction sites, followed by a description of how to do restriction site analyses via the toolbox (you can run more extensive analysis via the restriction analysis tools available in the toolbox). This section also includes an explanation of how to

simulate a gel with the selected enzymes. The final section in this chapter focuses on enzyme lists which represent an easy way of managing restriction enzymes.

### 19.3.1 Dynamic restriction sites

If you open a sequence, a sequence list etc, you will find the **Restriction Sites** group in the **Side Panel**.

As shown in figure 19.26 you can display restriction sites as colored triangles and lines on the sequence. The **Restriction sites** group in the side panel shows a list of enzymes, represented by different colors corresponding to the colors of the triangles on the sequence. By selecting or deselecting the enzymes in the list, you can specify which enzymes' restriction sites should be displayed.

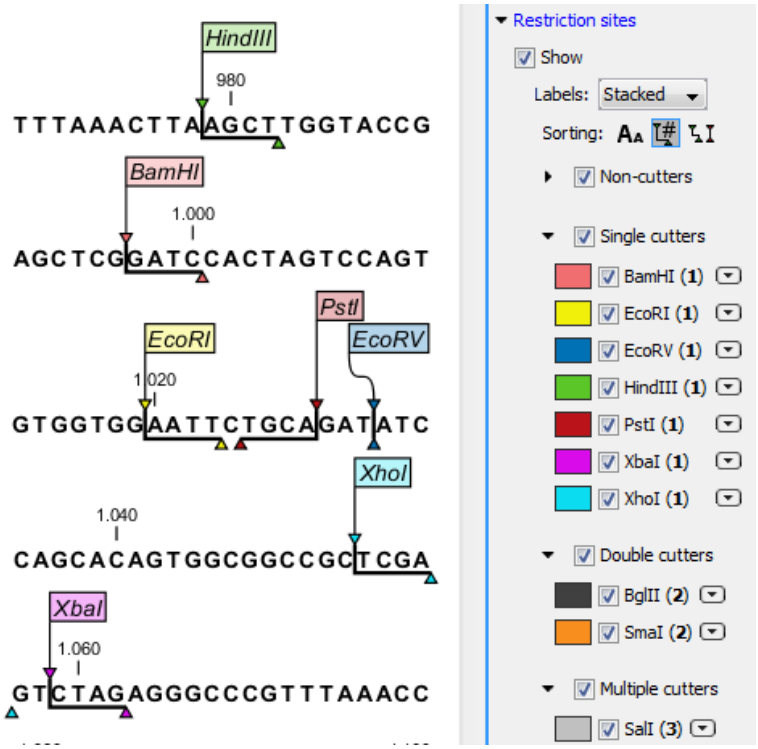



Figure 19.26: *Showing restriction sites of ten restriction enzymes.*

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 19.27). 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:

- **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 19.29).
- **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 19.28).

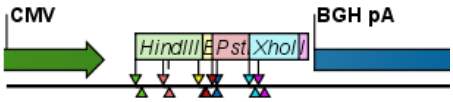



Figure 19.27: *Restriction site labels shown as flags.*

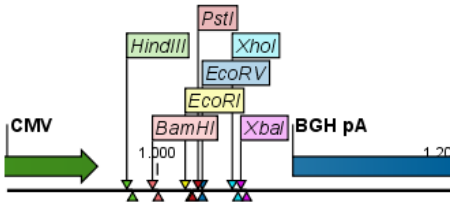



Figure 19.28: *Restriction site labels stacked.*

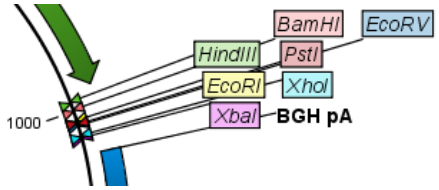



Figure 19.29: *Restriction site labels in radial layout.*

Note that in a circular view, the **Stacked** and **Radial** options also affect the layout of annotations.

### Sort enzymes

Just above the list of enzymes there are three buttons to be used for sorting the list (see figure 19.30):

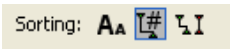



Figure 19.30: *Buttons to sort restriction enzymes.*

- **Sort enzymes alphabetically** (AA). Clicking this button will sort the list of enzymes alphabetically.
- **Sort enzymes by number of restriction sites** (#). 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 20 of the most popular enzymes, but you can easily modify this list and add more enzymes by clicking the **Manage enzymes button.** This will display the dialog shown in figure 19.31.

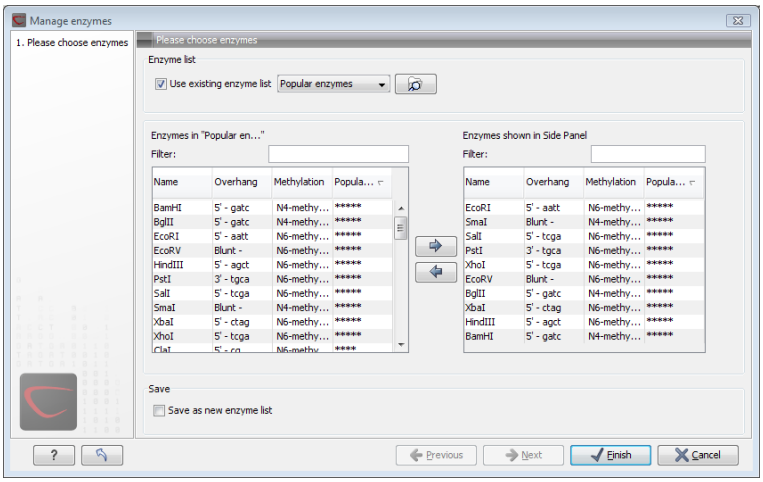



Figure 19.31: *Adding or removing enzymes from the Side Panel.*

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**. See section 19.5 for more about creating and modifying enzyme lists.

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 an existing enzyme list, this panel shows all the enzymes available [3].
- 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 (). If you e.g. wish to use EcoRV and BamHI, select these two enzymes and add them to the right side panel.

[3]The *CLC Genomics Workbench* comes with a standard set of enzymes based on http://rebase.neb.com/rebase/rebase.html. You can customize the enzyme database for your installation, see section F

If you wish to use all the enzymes in the list:

**Click in the panel to the left | press Ctrl + A (⌘ + A on Mac) | Add ( )**

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 e.g. a 3' overhang. In this case, you can sort the list by clicking the Overhang column heading, and all the enzymes producing 3' overhangs will be listed together for easy selection.

When looking for a specific enzyme, it is easier to use the Filter. If you wish to find e.g. HindIII sites, simply type HindIII into the filter, and the list of enzymes will shrink automatically to only include the HindIII enzyme. This can also be used to only show enzymes producing e.g. a 3' overhang as shown in figure 19.50.

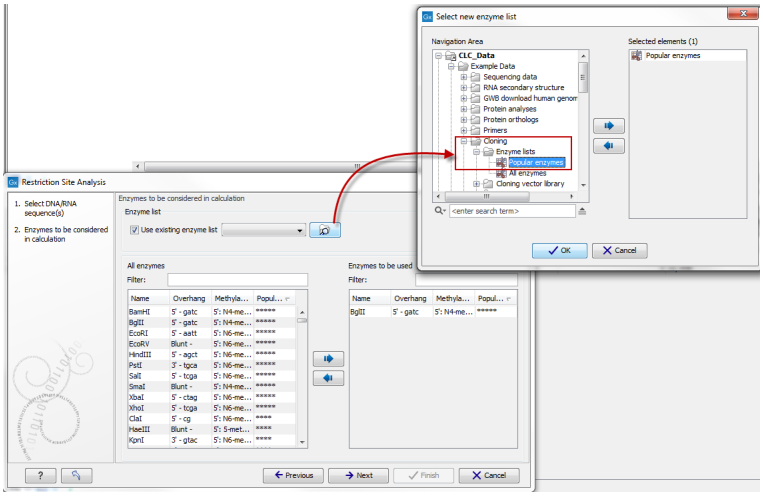


Figure 19.32: *Selecting enzymes.*

If you need more detailed information and filtering of the enzymes, either place your mouse cursor on an enzyme for one second to display additional information (see figure 19.51), or use the view of enzyme lists (see 19.5).

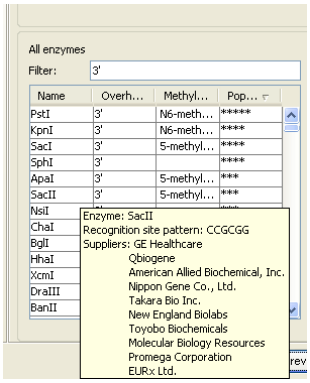



Figure 19.33: *Showing additional information about an enzyme like recognition sequence or a list of commercial vendors.*

At the bottom of the dialog, you can select to save this list of enzymes as a new file. In this way, you can save the selection of enzymes for later use.

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

Section 19.3.1 describes how to add more enzymes to the list in the Side Panel based on the name of the enzyme, overhang, methylation sensitivity etc. However, you will often find yourself in a situation where you need a more sophisticated and explorative approach.

An illustrative example: you have a selection on a sequence, and you wish to find enzymes cutting within the selection, but not outside. This problem often arises during design of cloning experiments. In this case, you do not know the name of the enzyme, so you want the Workbench to find the enzymes for you:

**right-click the selection | Show Enzymes Cutting Inside/Outside Selection (  )**

This will display the dialog shown in figure 19.34 where you can specify which enzymes should initially be considered.

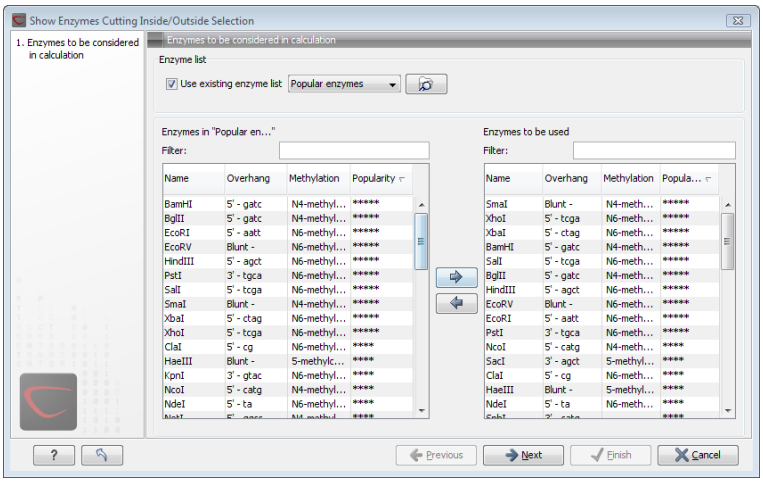



Figure 19.34: *Choosing enzymes to be considered.*

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**. See section 19.5 for more about creating and modifying enzyme lists.

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 an existing enzyme list, this panel shows all the enzymes available [4].
- 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 (  ). If you e.g. wish to use EcoRV and BamHI, select these two enzymes and add them to the right side panel.

[4]The *CLC Genomics Workbench* comes with a standard set of enzymes based on http://rebase.neb.com/rebase/rebase.html. You can customize the enzyme database for your installation, see section F

If you wish to use all the enzymes in the list:

**Click in the panel to the left | press Ctrl + A (⌘ + A on Mac) | Add (➡)**

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 e.g. a 3' overhang. In this case, you can sort the list by clicking the Overhang column heading, and all the enzymes producing 3' overhangs will be listed together for easy selection.

When looking for a specific enzyme, it is easier to use the Filter. If you wish to find e.g. HindIII sites, simply type HindIII into the filter, and the list of enzymes will shrink automatically to only include the HindIII enzyme. This can also be used to only show enzymes producing e.g. a 3' overhang as shown in figure 19.50.

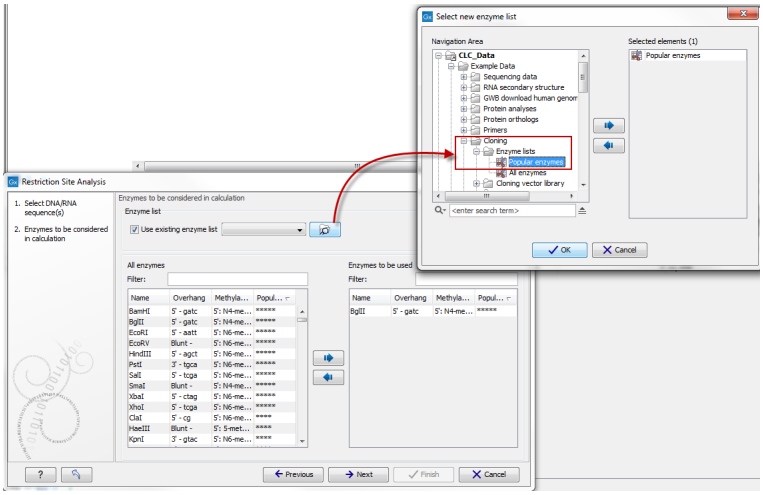



Figure 19.35: *Selecting enzymes.*

If you need more detailed information and filtering of the enzymes, either place your mouse cursor on an enzyme for one second to display additional information (see figure 19.51), or use the view of enzyme lists (see 19.5).

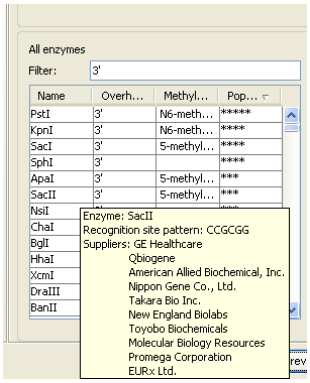



Figure 19.36: *Showing additional information about an enzyme like recognition sequence or a list of commercial vendors.*

Clicking **Next** will show the dialog in figure 19.37.

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.

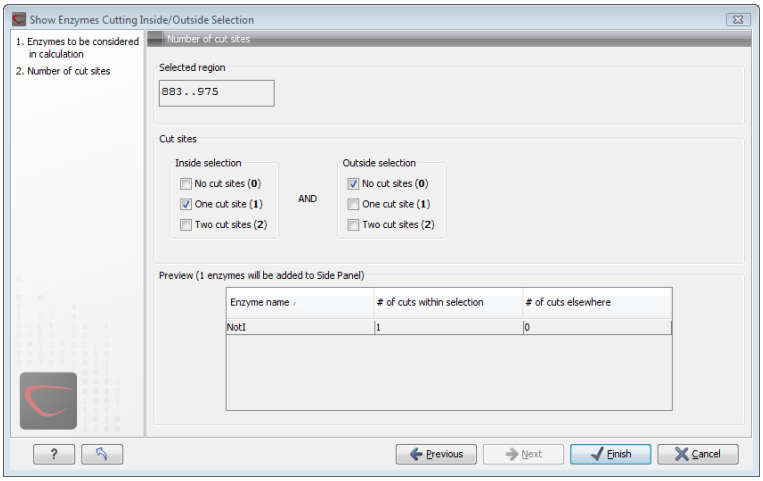



Figure 19.37: *Deciding number of cut sites inside and outside the selection.*

In the example described above, "One cut site (1)" should be selected to only show enzymes cutting once in the selection.

- **Outside selection**. Specify how many times you wish the enzyme to cut outside the selection (i.e. the rest of the sequence). In the example above, "No cut sites (0)" should be selected.

These panels offer a lot of flexibility for combining number of cut sites inside and outside the selection, respectively. To give a hint of how many enzymes will be added based on the combination of cut sites, the preview panel at the bottom lists the enzymes which will be added when you click **Finish**. Note that this list is dynamically updated when you change the number of cut sites. The enzymes shown in brackets [] are enzymes which are already present in the Side Panel.

If you have selected more than one region on the sequence (using Ctrl or ⌘ ), they will be treated as individual regions. This means that the criteria for cut sites apply to each region.

### Show enzymes with compatible ends

Besides what is described above, there is a third way of adding enzymes to the Side Panel and thereby displaying them on the sequence. It is based on the overhang produced by cutting with an enzyme and will find enzymes producing a compatible overhang:

**right-click the restriction site | Show Enzymes with Compatible Ends (**  **)**

This will display the dialog shown in figure 19.38.

At the top you can choose whether the enzymes considered should have an exact match or not. 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.

We advice trying **Exact match** first, and use **All matches** as an alternative if a satisfactory result cannot be achieved.

At the bottom of the dialog, the list of enzymes producing compatible overhangs is shown. Use the arrows to add enzymes which will be displayed on the sequence which you press **Finish**.

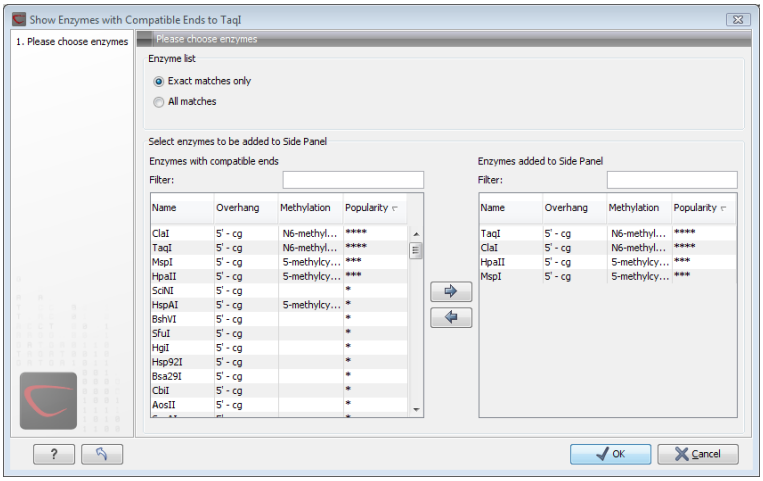



Figure 19.38: *Enzymes with compatible ends.*

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.

### 19.3.2 Restriction site analysis from the Toolbox

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 ( )**

This will display the dialog shown in figure 19.39.

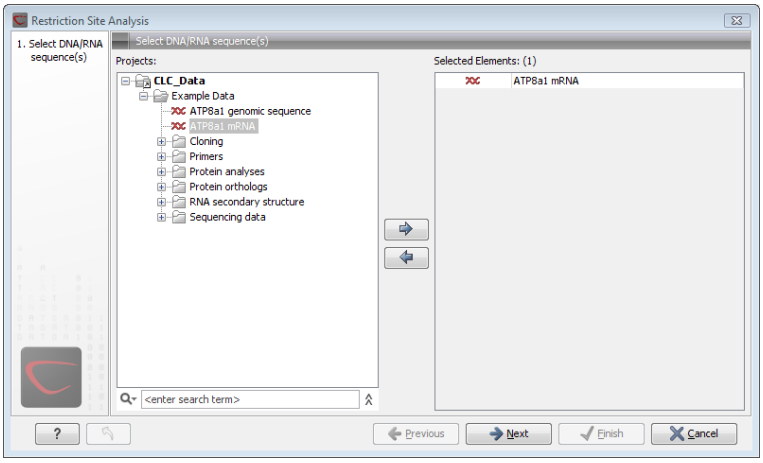



Figure 19.39: *Choosing sequence ATP8a1 mRNA for restriction map analysis.*

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.

### Selecting, sorting and filtering enzymes

Clicking **Next** lets you define which enzymes to use as basis for finding restriction sites on the sequence. 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**. See section 19.5 for more about creating and modifying enzyme lists.

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 an existing enzyme list, this panel shows all the enzymes available [5].
- 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 (  ). If you e.g. wish to use EcoRV and BamHI, select these two enzymes and add them to the right side panel.

If you wish to use all the enzymes in the list:

**Click in the panel to the left | press Ctrl + A (⌘ + A on Mac) | Add ()**

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 e.g. a 3' overhang. In this case, you can sort the list by clicking the Overhang column heading, and all the enzymes producing 3' overhangs will be listed together for easy selection.

When looking for a specific enzyme, it is easier to use the Filter. If you wish to find e.g. HindIII sites, simply type HindIII into the filter, and the list of enzymes will shrink automatically to only include the HindIII enzyme. This can also be used to only show enzymes producing e.g. a 3' overhang as shown in figure 19.50.

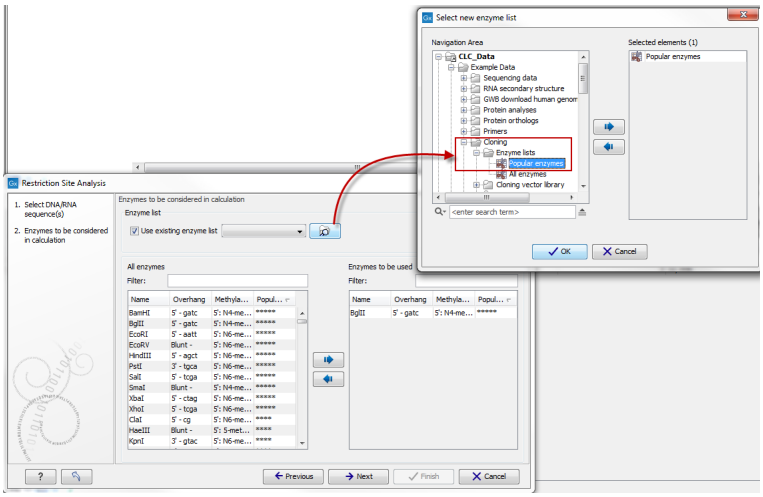



Figure 19.40: *Selecting enzymes.*

If you need more detailed information and filtering of the enzymes, either place your mouse cursor on an enzyme for one second to display additional information (see figure 19.51), or use the view of enzyme lists (see 19.5).

[5] The *CLC Genomics Workbench* comes with a standard set of enzymes based on http://rebase.neb.com/rebase/rebase.html. You can customize the enzyme database for your installation, see section F

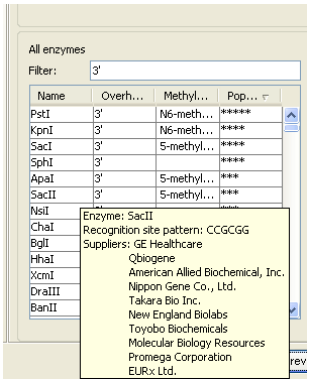



Figure 19.41: *Showing additional information about an enzyme like recognition sequence or a list of commercial vendors.*

### Number of cut sites

Clicking **Next** confirms the list of enzymes which will be included in the analysis, and takes you to the dialog shown in figure 19.42.

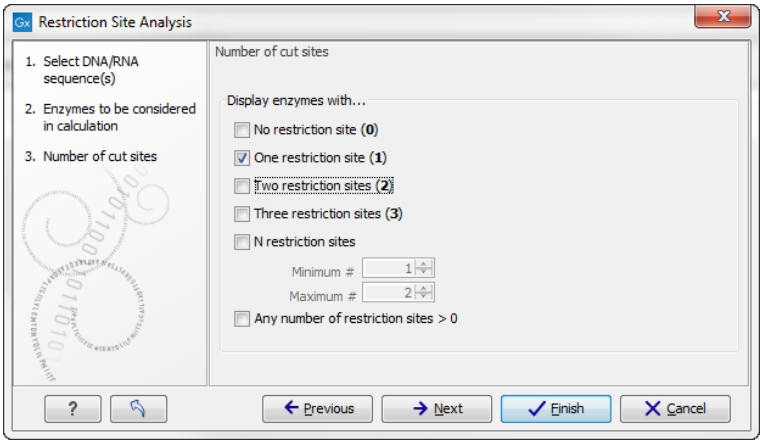



Figure 19.42: *Selecting number of cut sites.*

If you wish the output of the restriction map analysis only to include restriction enzymes which cut the sequence a specific number of times, use the checkboxes in this dialog:

- No restriction site (**0**)
- One restriction site (**1**)
- Two restriction sites (**2**)
- Three restriction site (**3**)
- N restriction sites
  - Minimum
  - Maximum
- Any number of restriction sites > 0

The default setting is to include the enzymes which cut the sequence one or two times.

You can use the checkboxes to perform very specific searches for restriction sites: e.g. if you wish to find enzymes which do not cut the sequence, or enzymes cutting exactly twice.

### Output of restriction map analysis

Clicking next shows the dialog in figure 19.43.

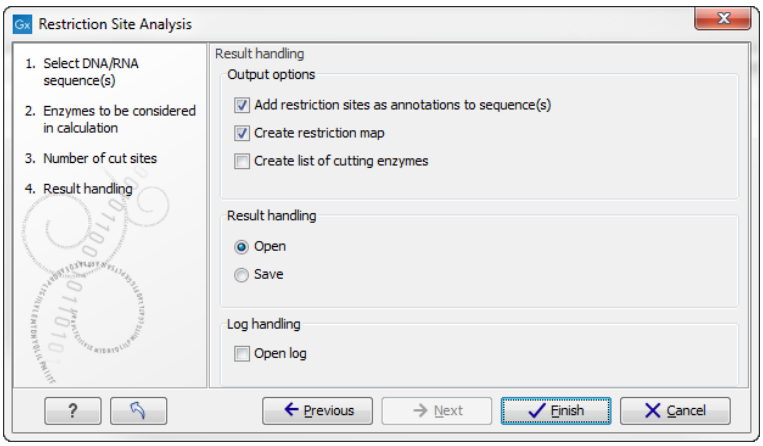



Figure 19.43: *Choosing to add restriction sites as annotations or creating a restriction map.*

This dialog 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 19.44) and save the annotations for later use.
- **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 19.45. 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.
  - As a **table of fragments** which shows the sequence fragments that would be the result of cutting the sequence with the selected enzymes (see figure19.46).
  - As a **virtual gel** simulation which shows the fragments as bands on a gel (see figure 19.48).
    For more information about gel electrophoresis, see section 19.4.

The following sections will describe these output formats in more detail.

In order to complete the analysis click **Finish** (see section 8.2 for information about the Save and Open options).

### Restriction sites as annotation on the sequence

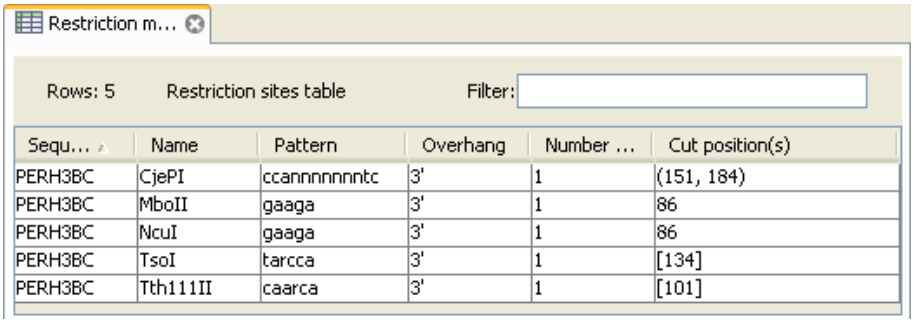
If you chose to add the restriction sites as annotation to the sequence, the result will be similar to the sequence shown in figure 19.44. See section 10.3 for more information about viewing annotations.




Figure 19.44: *The result of the restriction analysis shown as annotations.*

### Table of restriction sites

The restriction map can be shown as a table of restriction sites (see figure 19.45).




| Sequ... | Name | Pattern | Overhang | Number ... | Cut position(s) |
|---|---|---|---|---|---|
| PERH3BC | CjePI | ccannnnnnntc | 3' | 1 | (151, 184) |
| PERH3BC | MboII | gaaga | 3' | 1 | 86 |
| PERH3BC | NcuI | gaaga | 3' | 1 | 86 |
| PERH3BC | TsoI | tarcca | 3' | 1 | [134] |
| PERH3BC | Tth111II | caarca | 3' | 1 | [101] |

Figure 19.45: *The result of the restriction analysis shown as annotations.*

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.
- **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 cuts more than once, the positions are separated by commas.
  - **[]** If the enzyme's recognition sequence is on the negative strand, the cut position is put in brackets (as the enzyme TsoI in figure 19.45 whose cut position is [134]).
  - **()** Some enzymes cut the sequence twice for each recognition site, and in this case the two cut positions are surrounded by parentheses.

### Table of restriction fragments

The restriction map can be shown as a table of fragments produced by cutting the sequence with the enzymes:

**Click the Fragments button (  ) at the bottom of the view**

The table is shown in see figure 19.46.

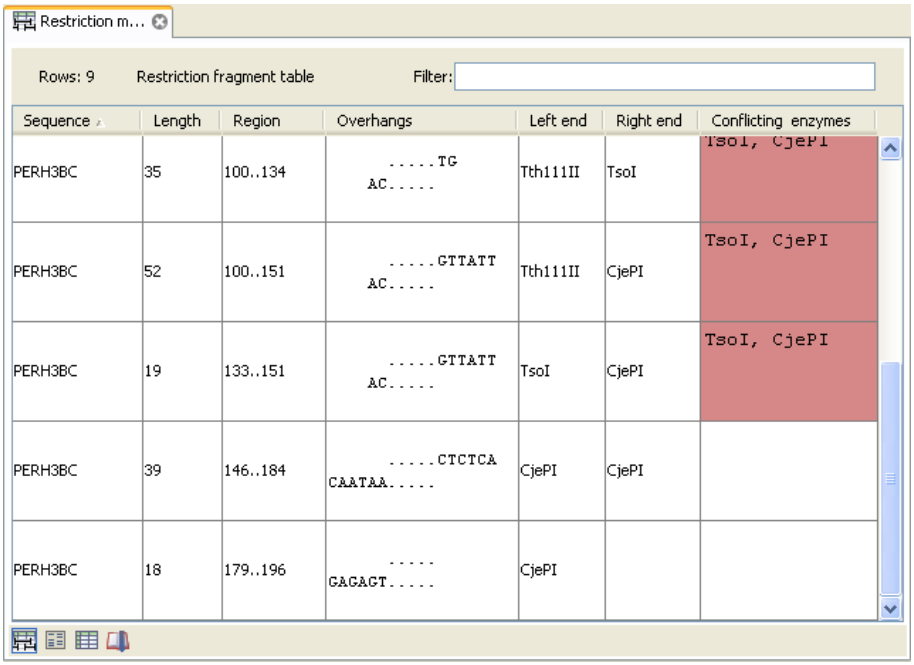



Figure 19.46: *The result of the restriction analysis shown as annotations.*

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 [6]. 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**. 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.

[6] Furthermore, if this is the case, you will see the names of the other enzymes in the **Conflicting Enzymes** column

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.

#### Gel

The restriction map can also be shown as a gel. This is described in section 19.4.1.

## 19.4 Gel electrophoresis

*CLC Genomics Workbench* enables the user to simulate the separation of nucleotide sequences on a gel. This feature is useful when e.g. 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 two main ways to simulate gel separation of nucleotide sequences:

- One or more sequences can be digested with restriction enzymes and the resulting fragments can be separated on a gel.
- A number of existing sequences can be separated on a gel.

There are several ways to apply these functionalities as described below.

### 19.4.1 Separate fragments of sequences on gel

This section explains how to simulate a gel electrophoresis of one or more sequences which are digested with restriction enzymes. There are two ways to do this:

- When performing the **Restriction Site Analysis** from the **Toolbox**, you can choose to create a restriction map which can be shown as a gel. This is explained in section **??**.
- From all the graphical views of sequences, you can right-click the name of the sequence and choose: **Digest Sequence with Selected Enzymes and Run on Gel (  )**. The views where this option is available are listed below:
  - Circular view (see section 10.2).
  - Ordinary sequence view (see section 10.1).
  - Graphical view of sequence lists (see section 10.6).
  - Cloning editor (see section 19.1).
  - Primer designer (see section 17.3).

  Furthermore, you can also right-click an empty part of the view of the graphical view of sequence lists and the cloning editor and choose **Digest All Sequences with Selected Enzymes and Run on Gel**.
  **Note!** When using the right-click options, the sequence will be digested with the enzymes that are selected in the **Side Panel**. This is explained in section 10.1.2.

The view of the gel is explained in section 19.4.3

### 19.4.2 Separate sequences on gel

To separate sequences without restriction enzyme digestion, first create a sequence list of the sequences in question (see section 10.6). Then click the **Gel** button ( ) at the bottom of the view of the sequence list.

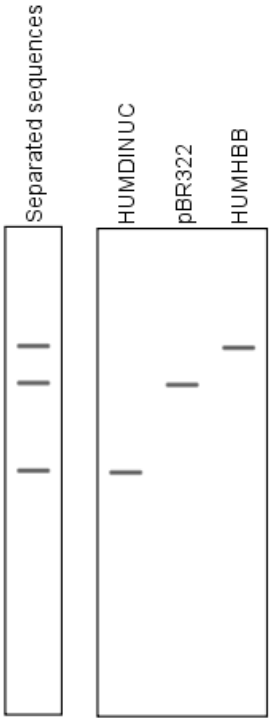



Figure 19.47: *A sequence list shown as a gel.*

For more information about the view of the gel, see the next section.

### 19.4.3 Gel view

In figure 19.48 you can see a simulation of a gel with its **Side Panel** to the right. This view will be explained in this section.

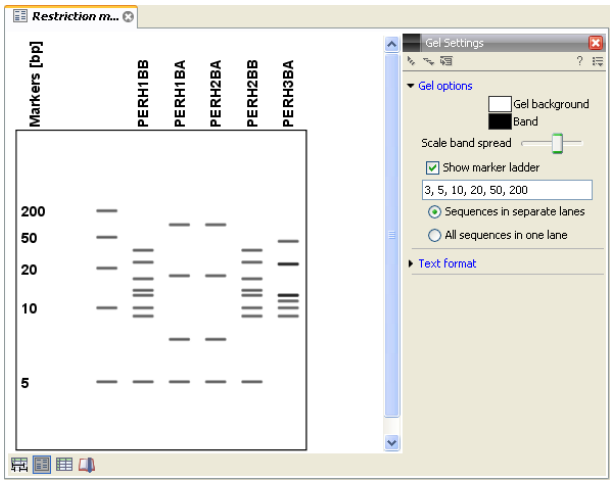



Figure 19.48: *Five lanes showing fragments of five sequences cut with restriction enzymes.*

#### Information on bands / fragments

You can get information about the individual bands by hovering the mouse cursor on the band of interest. This will display a tool tip with the following information:

- Fragment length
- Fragment region on the original sequence
- Enzymes cutting at the left and right ends, respectively

For gels comparing whole sequences, you will see the sequence name and the length of the sequence.

**Note!** You have to be in **Selection** ( ) or **Pan** ( ) mode in order to get this information.

It can be useful to add markers to the gel which enables you to compare the sizes of the bands. This is done by clicking **Show marker ladder** in the **Side Panel**.

Markers can be entered into the text field, separated by commas.

#### Modifying the layout

The background of the lane and the colors of the bands can be changed in the **Side Panel**. Click the colored box to display a dialog for picking a color. The slider **Scale band spread** can be used to adjust the effective time of separation on the gel, i.e. how much the bands will be spread over the lane. In a real electrophoresis experiment this property will be determined by several factors including time of separation, voltage and gel density.

You can also choose how many lanes should be displayed:

- **Sequences in separate lanes**. This simulates that a gel is run for each sequence.
- **All sequences in one lane**. This simulates that one gel is run for all sequences.

You can also modify the layout of the view by zooming in or out. Click **Zoom in** ( ) or **Zoom out** ( ) in the Toolbar and click the view.

Finally, you can modify the format of the text heading each lane in the **Text format** preferences in the **Side Panel**.

## 19.5 Restriction enzyme lists

*CLC Genomics Workbench* includes all the restriction enzymes available in the **REBASE** database[7]. 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 e.g. all enzymes available in the laboratory freezer, all enzymes used to create a given restriction map or all enzymes that are available form the preferred vendor.

[7] You can customize the enzyme database for your installation, see section F

In the example data (see section 1.6.2) 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*.

This section describes how you can create an enzyme list, and how you can modify it.

### 19.5.1 Create enzyme list

*CLC Genomics Workbench* uses enzymes from the **REBASE** restriction enzyme database at http://rebase.neb.com[8].

To create an enzyme list of a subset of these enzymes:

**File** | **New** | **Enzyme list (  )**

This opens the dialog shown in figure 19.49

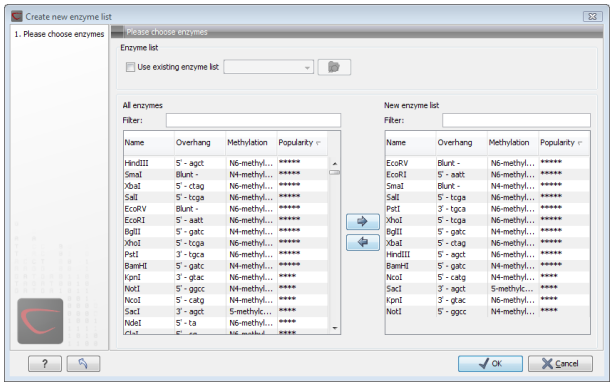



Figure 19.49: *Choosing enzymes for the new enzyme list.*

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**. See section 19.5 for more about creating and modifying enzyme lists.

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 an existing enzyme list, this panel shows all the enzymes available [9].
- 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 (  ). If you e.g. wish to use EcoRV and BamHI, select these two enzymes and add them to the right side panel.

If you wish to use all the enzymes in the list:

**Click in the panel to the left** | **press Ctrl + A (⌘ + A on Mac)** | **Add ()**

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 e.g. a 3'

[8] You can customize the enzyme database for your installation, see section F

[9] The *CLC Genomics Workbench* comes with a standard set of enzymes based on http://rebase.neb.com/rebase/rebase.html. You can customize the enzyme database for your installation, see section F

overhang. In this case, you can sort the list by clicking the Overhang column heading, and all the enzymes producing 3' overhangs will be listed together for easy selection.

When looking for a specific enzyme, it is easier to use the Filter. If you wish to find e.g. HindIII sites, simply type HindIII into the filter, and the list of enzymes will shrink automatically to only include the HindIII enzyme. This can also be used to only show enzymes producing e.g. a 3' overhang as shown in figure 19.50.

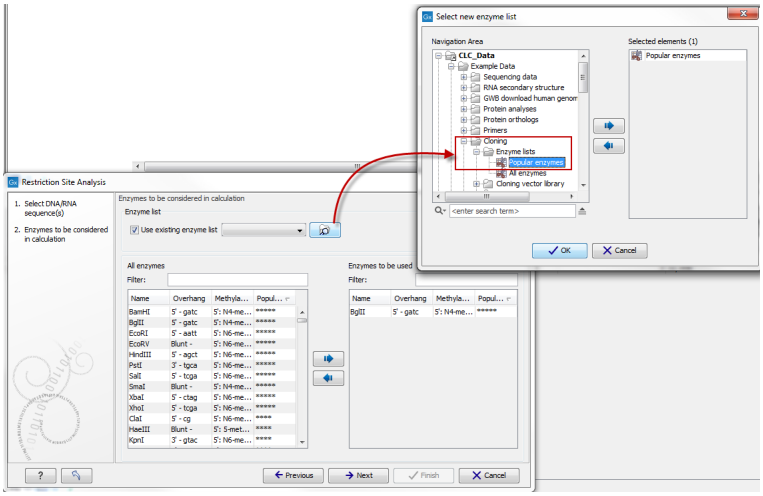



Figure 19.50: *Selecting enzymes.*

If you need more detailed information and filtering of the enzymes, either place your mouse cursor on an enzyme for one second to display additional information (see figure 19.51), or use the view of enzyme lists (see 19.5).

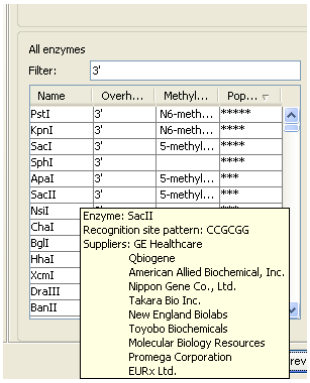



Figure 19.51: *Showing additional information about an enzyme like recognition sequence or a list of commercial vendors.*

Click **Finish** to open the enzyme list.

### 19.5.2 View and modify enzyme list

An enzyme list is shown in figure 19.52. The list can be sorted by clicking the columns, and you can use the filter at the top right corner to search for specific enzymes, recognition sequences etc.

If you wish to remove or add enzymes, click the **Add/Remove Enzymes** button at the bottom of the view. This will present the same dialog as shown in figure 19.49 with the enzyme list shown

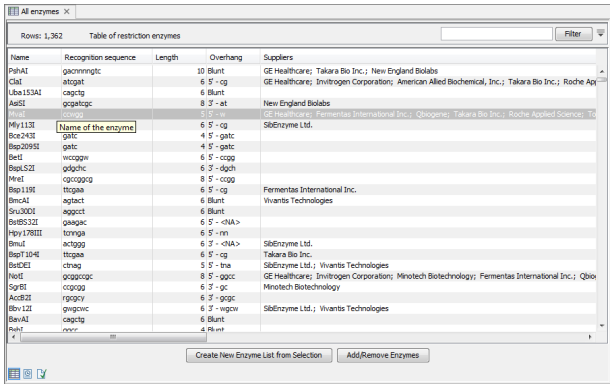



Figure 19.52: *An enzyme list.*

to the right.

If you wish to extract a subset of an enzyme list:

> **open the list** | **select the relevant enzymes** | **right-click** | **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. E.g. 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.

# Chapter 20

# Sequence alignment

**Contents**


**20.1 Create an alignment . . . 437**
20.1.1 Gap costs . . . 438
20.1.2 Fast or accurate alignment algorithm . . . 438
20.1.3 Aligning alignments . . . 439
20.1.4 Fixpoints . . . 440
**20.2 View alignments . . . 442**
20.2.1 Bioinformatics explained: Sequence logo . . . 444
**20.3 Edit alignments . . . 446**
20.3.1 Move residues and gaps . . . 446
20.3.2 Insert gaps . . . 446
20.3.3 Delete residues and gaps . . . 447
20.3.4 Copy annotations to other sequences . . . 447
20.3.5 Move sequences up and down . . . 447
20.3.6 Delete, rename and add sequences . . . 448
20.3.7 Realign selection . . . 448
**20.4 Join alignments . . . 449**
20.4.1 How alignments are joined . . . 449
**20.5 Pairwise comparison . . . 450**
20.5.1 Pairwise comparison on alignment selection . . . 451
20.5.2 Pairwise comparison parameters . . . 451
20.5.3 The pairwise comparison table . . . 452
**20.6 Bioinformatics explained: Multiple alignments . . . 453**
20.6.1 Use of multiple alignments . . . 453
20.6.2 Constructing multiple alignments . . . 454


*CLC Genomics Workbench* can align nucleotides and proteins using a *progressive alignment* algorithm (see section 20.6 or read the White paper on alignments in the **Science** section of http://www.clcbio.com).

This chapter describes how to use the program to align sequences. The chapter also describes alignment algorithms in more general terms.

## 20.1 Create an alignment

Alignments can be created from sequences, sequence lists (see section 10.6), 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 20.1.

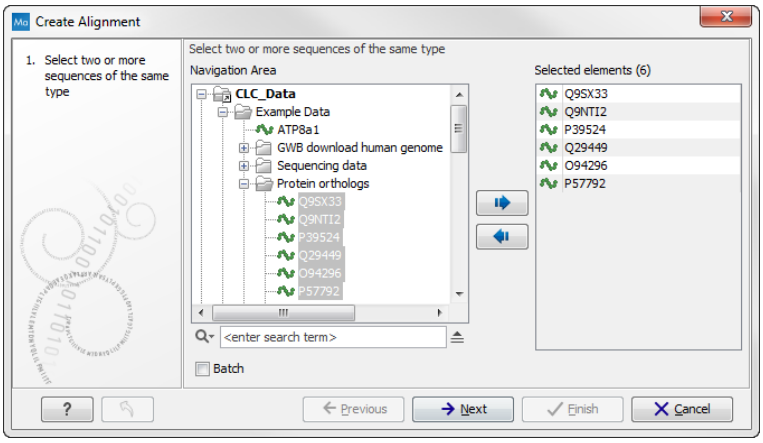



Figure 20.1: *Creating an alignment.*

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 20.2.

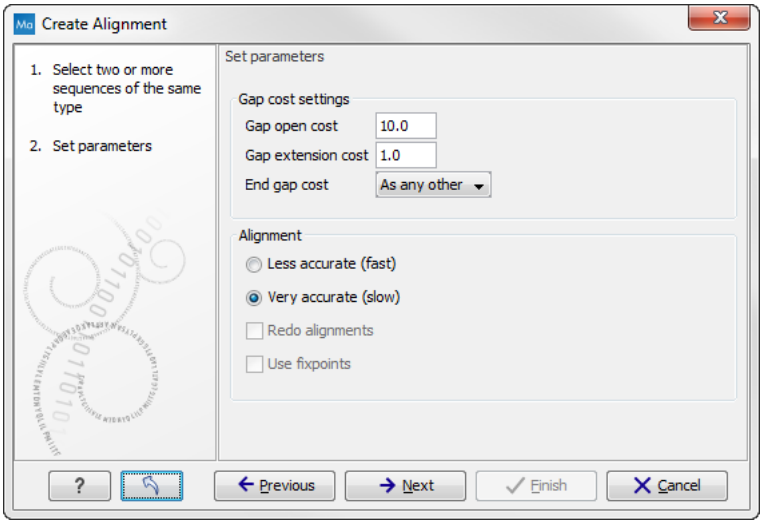



Figure 20.2: *Adjusting alignment algorithm parameters.*

### 20.1.1 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 20.3 and 20.4 illustrate the differences between the different gap scores at the sequence ends.

### 20.1.2 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.

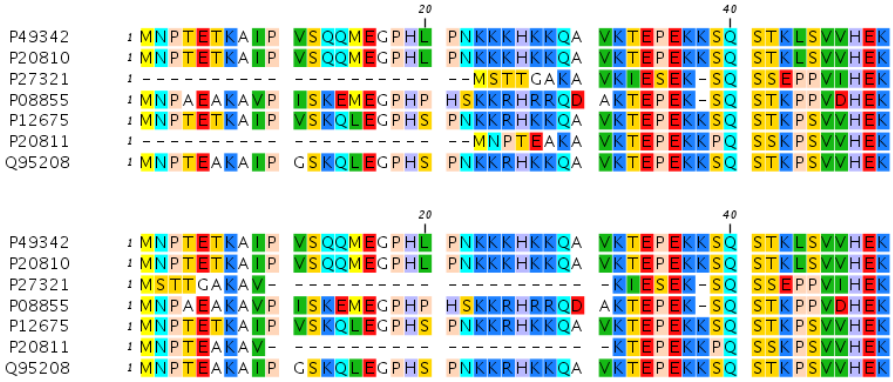



Figure 20.3: *The first 50 positions of two different alignments of seven calpastatin sequences. The top alignment is made with cheap end gaps, while the bottom alignment is made with end gaps having the same price as any other gaps. In this case it seems that the latter scoring scheme gives the best result.*

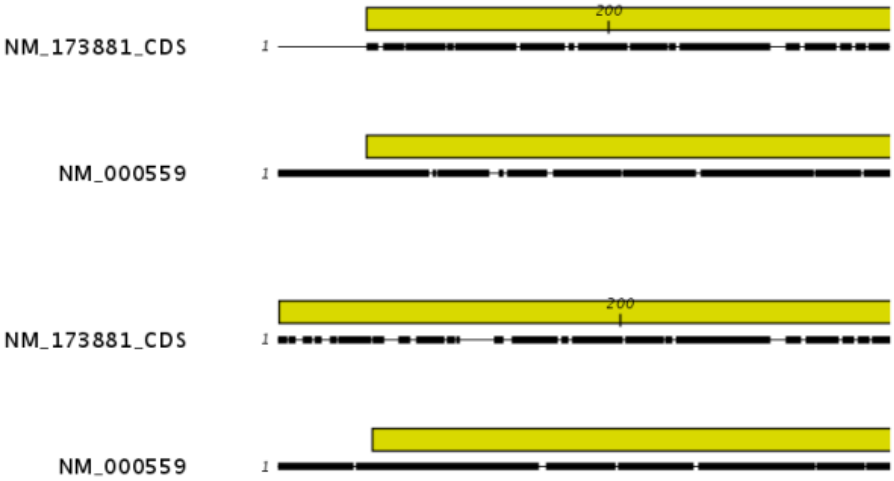



Figure 20.4: *The alignment of the coding sequence of bovine myoglobin with the full mRNA of human gamma globin. The top alignment is made with free end gaps, while the bottom alignment is made with end gaps treated as any other. The yellow annotation is the coding sequence in both sequences. It is evident 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.

### 20.1.3 Aligning alignments

If you have selected an existing alignment in the first step (20.1), 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 20.5.

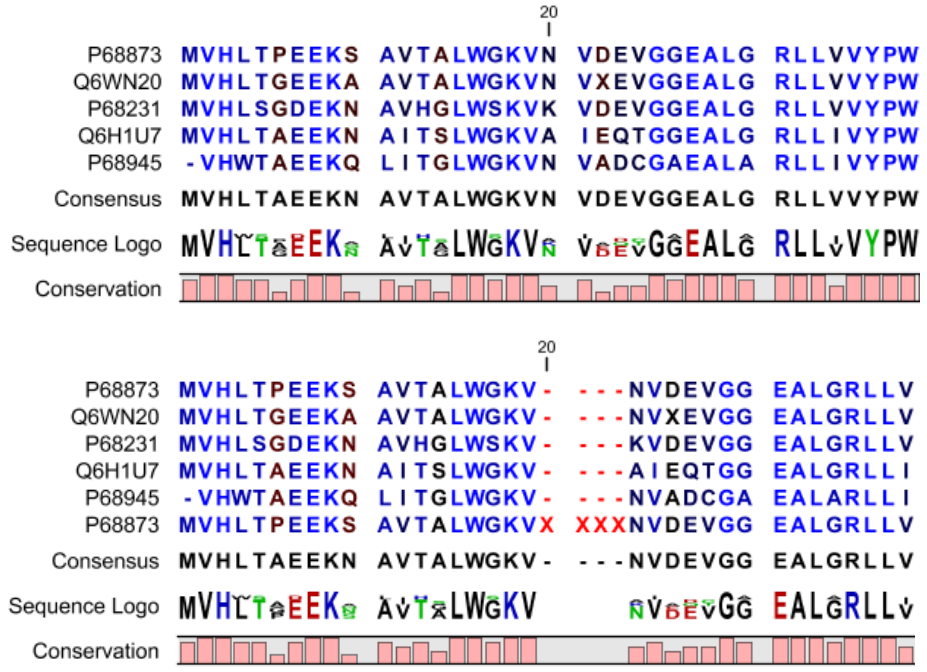



Figure 20.5: *The top figures shows the original alignment. In the bottom panel a single sequence with four inserted X's are aligned to the original alignment. This introduces gaps in all sequences of the original alignment. All other positions in the original alignment are fixed.*

This 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.

### 20.1.4 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.

Fixpoints are added to sequences or alignments before clicking "Create alignment". 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**

This will add an annotation labeled "Fixpoint" to the sequence (see figure 20.6). Use this procedure to add fixpoints to the other sequence(s) that should be forced to align to each other.

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 20.7 the result of an alignment using fixpoints is illustrated.

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

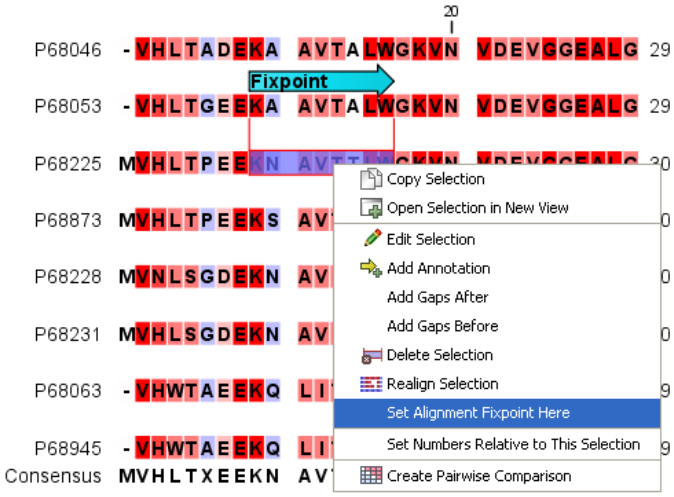



Figure 20.6: *Adding a fixpoint to a sequence in an existing alignment. At the top you can see a fixpoint that has already been added.*

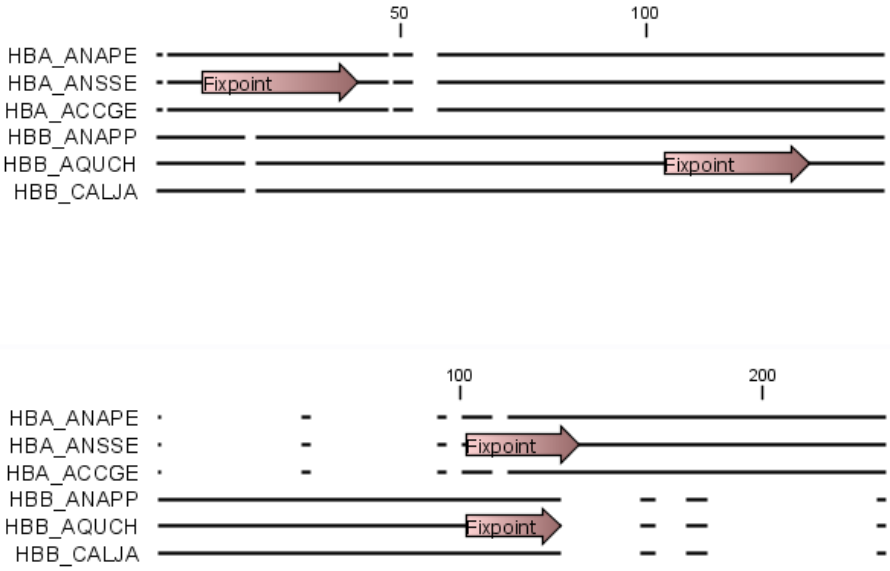



Figure 20.7: *Realigning using fixpoints. In the top view, fixpoints have been added to two of the sequences. In the view below, the alignment has been realigned using the fixpoints. The three top sequences are very similar, and therefore they follow the one sequence (number two from the top) that has a fixpoint.*

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**

## 20.2 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 10.1 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. E.g. 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 I and H.) 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 6.6.
    * **Height.** Specifies the height of the graph.
    * **Type.** The type of the graph.
      · **Line plot.** Displays the graph as a line plot.
      · **Bar plot.** Displays the graph as a bar plot.
      · **Colors.** Displays the graph 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 6.6).
    * **Height.** Specifies the height of the graph.
    * **Type.** The type of the graph.
      · **Line plot.** Displays the graph as a line plot.
      · **Bar plot.** Displays the graph as a line plot.
      · **Colors.** Displays the graph 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 20.2.1 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.

### 20.2.1 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 20.8). This is also the case for repressor sites as seen for the Cro repressor of bacteriophage $\lambda$.

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 20.8). 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 20.8 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.

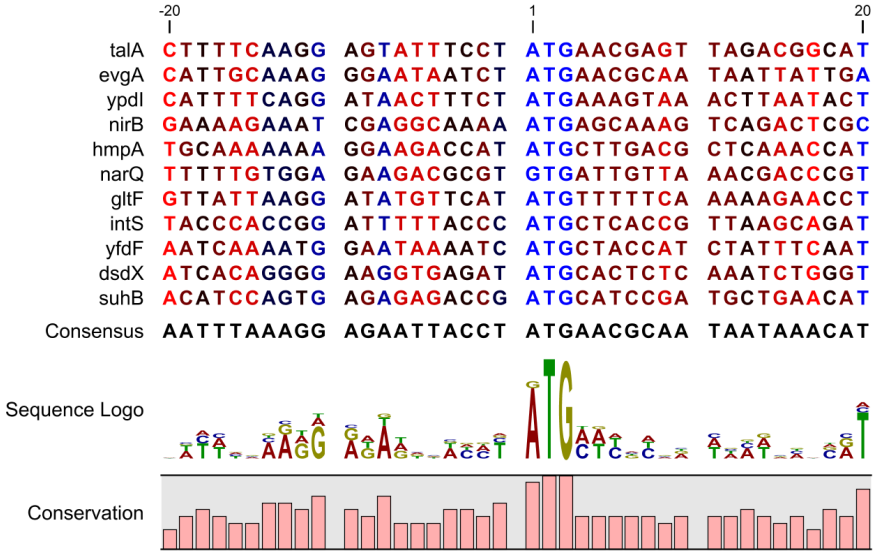



Figure 20.8: *Ungapped sequence alignment of eleven* E. coli *sequences defining a start codon. The start codons start at position 1. Below the alignment is shown the corresponding sequence logo. As seen, a GTG start codon and the usual ATG start codons are present in the alignment. This can also be visualized in the logo at position 1.*

### 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.

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]

## 20.3 Edit alignments

### 20.3.1 Move residues and gaps

The placement of gaps in the alignment can be changed by modifying the parameters when creating the alignment (see section 20.1). 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 20.9).

**Note!** Residues can only be moved when they are next to a gap.

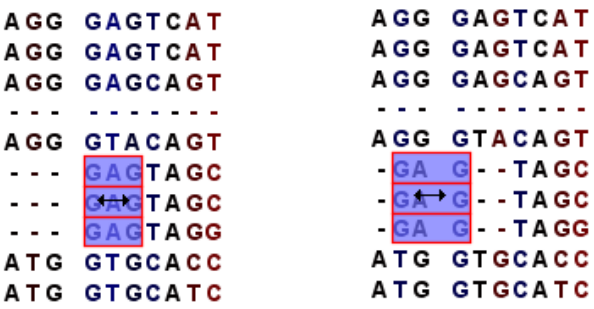



Figure 20.9: *Moving a part of an alignment. Notice the change of mouse pointer to a horizontal arrow.*

### 20.3.2 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 e.g. five residues, 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.

### 20.3.3 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'**

### 20.3.4 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. 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.

### 20.3.5 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**

### 20.3.6 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**

This can be undone by clicking **Undo** () in the Toolbar.

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 20.1).

The same procedure can be used for joining two alignments.

### 20.3.7 Realign selection

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** | **Realign selection**

This will open **Step 2** in the "Create alignment" dialog, allowing you to set the parameters for the realignment (see section 20.1).

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 by hand, you may end up being unhappy with the result. In this case you may of course undo your edits, but another option is to select the region and realign it.
- **Adjusting the number of gaps.** If you have a region in an alignment which has too many gaps in your opinion, you can select the region and realign it. By choosing a relatively high gap cost you will be able to reduce the number of gaps.
- **Combine with fixpoints.** If you have an alignment where two residues are not aligned, but you know that they should have been. You can now 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.

## 20.4 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 20.10.

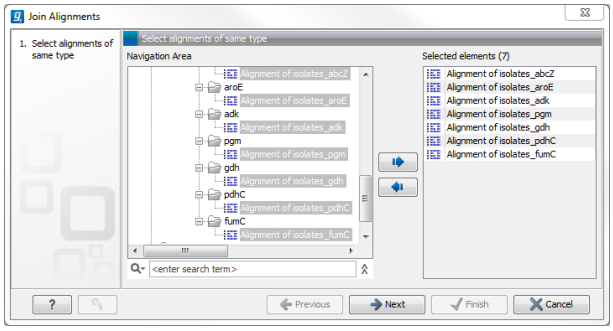



Figure 20.10: *Selecting two alignments to be joined.*

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 20.11.

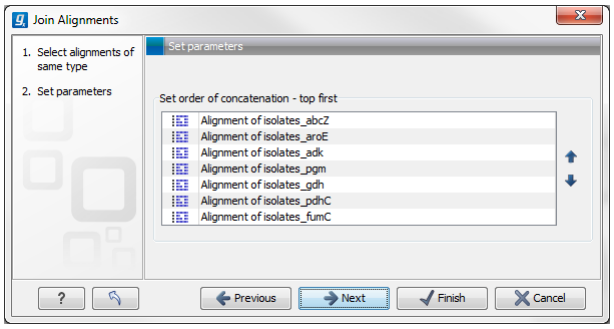



Figure 20.11: *Selecting order of concatenation.*

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 20.12.

### 20.4.1 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 20.12

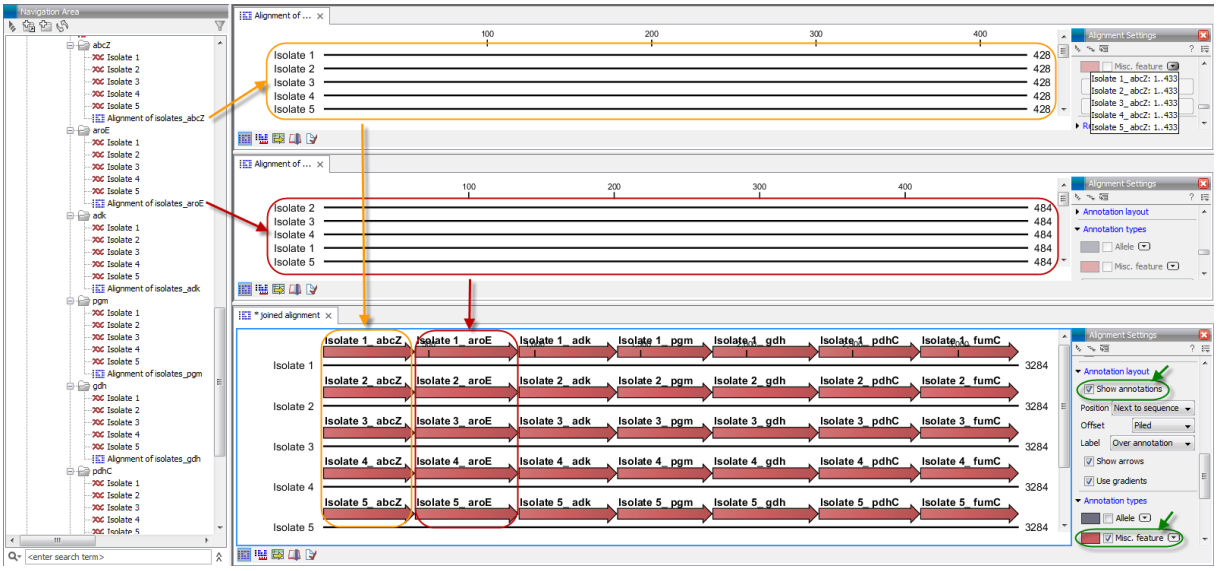


Figure 20.12: *The upper part of the figure shows two of the seven alignments for the genes "abcZ" and "aroE" respectively. Each alignment consists of sequences from one gene from five different isolates. The lower part of the figure shows the result of "Join Alignments". Seven genes have been joined 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.*

"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.

## 20.5 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 20.13:

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.

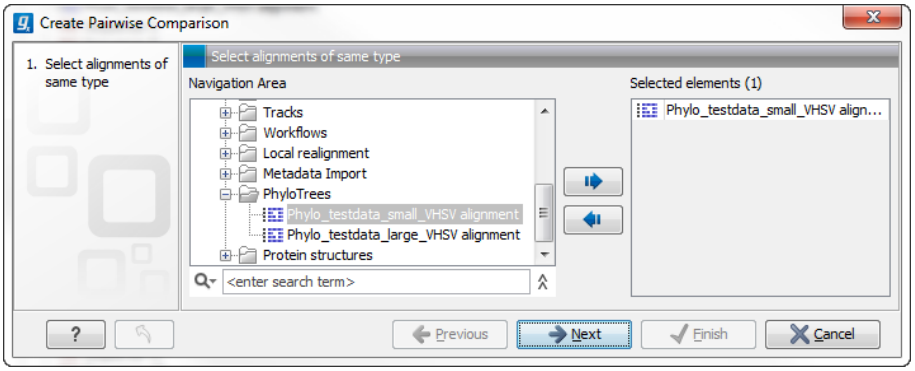



Figure 20.13: *Creating a pairwise comparison table.*

### 20.5.1 Pairwise comparison on alignment selection

A pairwise comparison can also be performed for a selected part of an alignment:

**right-click on an alignment selection | Pairwise Comparison (  )**

This leads directly to the dialog described in the next section.

### 20.5.2 Pairwise comparison parameters

There are five kinds of comparison that can be made between the sequences in the alignment, as shown in figure 20.14.

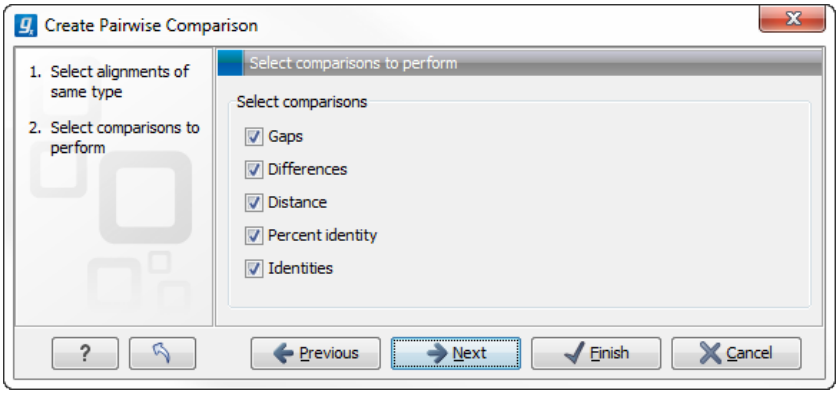


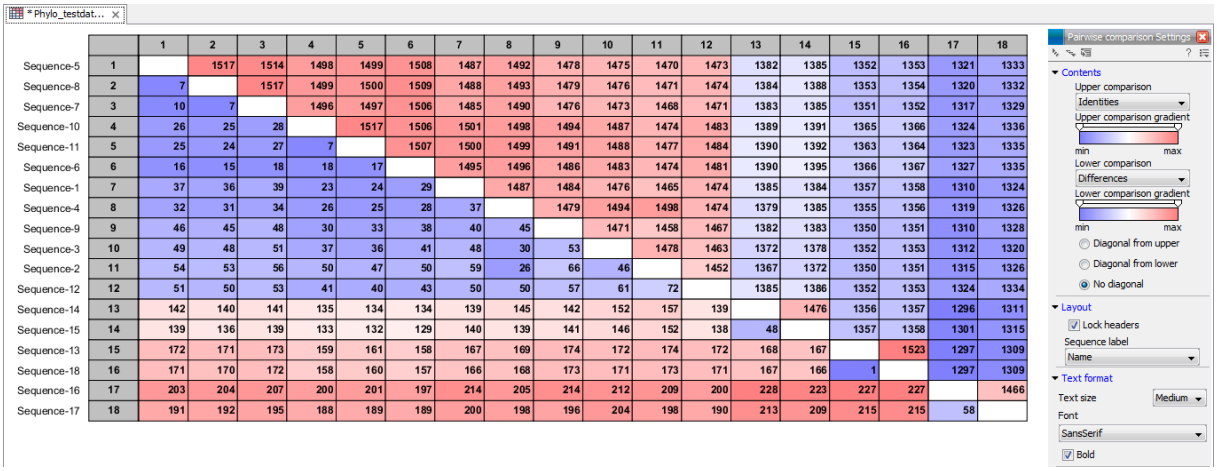

Figure 20.14: *Adjusting parameters for pairwise comparison.*

- **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.

### 20.5.3 The pairwise comparison table

The table shows the results of selected comparisons (see an example in figure 20.15). 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.




| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sequence-5 | 1 | | 1517 | 1514 | 1498 | 1499 | 1508 | 1487 | 1492 | 1478 | 1475 | 1470 | 1473 | 1382 | 1385 | 1352 | 1353 | 1321 | 1333 |
| Sequence-8 | 2 | 7 | | 1517 | 1499 | 1500 | 1509 | 1488 | 1493 | 1479 | 1476 | 1471 | 1474 | 1384 | 1388 | 1353 | 1354 | 1320 | 1332 |
| Sequence-7 | 3 | 10 | 7 | | 1496 | 1497 | 1506 | 1485 | 1490 | 1476 | 1473 | 1468 | 1471 | 1383 | 1385 | 1351 | 1352 | 1317 | 1329 |
| Sequence-10 | 4 | 26 | 25 | 28 | | 1517 | 1506 | 1501 | 1498 | 1494 | 1487 | 1474 | 1483 | 1389 | 1391 | 1365 | 1366 | 1324 | 1336 |
| Sequence-11 | 5 | 25 | 24 | 27 | 7 | | 1507 | 1500 | 1499 | 1491 | 1488 | 1477 | 1484 | 1390 | 1392 | 1363 | 1364 | 1323 | 1335 |
| Sequence-6 | 6 | 16 | 15 | 18 | 18 | 17 | | 1495 | 1496 | 1486 | 1483 | 1474 | 1481 | 1390 | 1395 | 1366 | 1367 | 1327 | 1335 |
| Sequence-1 | 7 | 37 | 36 | 39 | 23 | 24 | 29 | | 1487 | 1484 | 1476 | 1465 | 1474 | 1385 | 1384 | 1357 | 1358 | 1310 | 1324 |
| Sequence-4 | 8 | 32 | 31 | 34 | 26 | 25 | 28 | 37 | | 1479 | 1494 | 1498 | 1474 | 1379 | 1385 | 1355 | 1356 | 1319 | 1326 |
| Sequence-9 | 9 | 46 | 45 | 48 | 30 | 33 | 38 | 40 | 45 | | 1471 | 1458 | 1467 | 1382 | 1383 | 1350 | 1351 | 1310 | 1328 |
| Sequence-3 | 10 | 49 | 48 | 51 | 37 | 36 | 41 | 48 | 30 | 53 | | 1478 | 1463 | 1372 | 1378 | 1352 | 1353 | 1312 | 1320 |
| Sequence-2 | 11 | 54 | 53 | 56 | 50 | 47 | 50 | 59 | 26 | 66 | 46 | | 1452 | 1367 | 1372 | 1350 | 1351 | 1315 | 1326 |
| Sequence-12 | 12 | 51 | 50 | 53 | 41 | 40 | 43 | 50 | 50 | 57 | 61 | 72 | | 1385 | 1386 | 1352 | 1353 | 1324 | 1334 |
| Sequence-14 | 13 | 142 | 140 | 141 | 135 | 134 | 134 | 139 | 145 | 142 | 152 | 157 | 139 | | 1476 | 1356 | 1357 | 1296 | 1311 |
| Sequence-15 | 14 | 139 | 136 | 139 | 133 | 132 | 129 | 140 | 139 | 141 | 146 | 152 | 138 | 48 | | 1357 | 1358 | 1301 | 1315 |
| Sequence-13 | 15 | 172 | 171 | 173 | 159 | 161 | 158 | 167 | 169 | 174 | 172 | 174 | 172 | 168 | 167 | | 1523 | 1297 | 1309 |
| Sequence-18 | 16 | 171 | 170 | 172 | 158 | 160 | 157 | 166 | 168 | 173 | 171 | 173 | 171 | 167 | 166 | 1 | | 1297 | 1309 |
| Sequence-16 | 17 | 203 | 204 | 207 | 200 | 201 | 197 | 214 | 205 | 214 | 212 | 209 | 200 | 228 | 223 | 227 | 227 | | 1466 |
| Sequence-17 | 18 | 191 | 192 | 195 | 188 | 189 | 189 | 200 | 198 | 196 | 204 | 198 | 190 | 213 | 209 | 215 | 215 | 58 | |

Figure 20.15: *A pairwise comparison table.*

The following settings are present in the side panel:

- **Contents**
  - **Upper comparison** Selects the comparison to show in the upper triangle of the table.
  - **Upper comparison gradient** Selects the color gradient to use for the upper triangle.
  - **Lower comparison** Selects the comparison to show in the lower triangle. Choose the same comparison as in the upper triangle to show all the results of an asymmetric comparison.
  - **Lower comparison gradient** Selects the color gradient to use for the lower triangle.
  - **Diagonal from upper** Use this setting to show the diagonal results from the upper comparison.
  - **Diagonal from lower** Use this setting to show the diagonal results from the lower comparison.
  - **No Diagonal.** Leaves the diagonal table entries blank.
- **Layout**
  - **Lock headers** Locks the sequence labels and table headers when scrolling the table.
  - **Sequence label** Changes the sequence labels.
- **Text format**
  - **Text size** Changes the size of the table and the text within it.
  - **Font** Changes the font in the table.
  - **Bold** Toggles the use of boldface in the table.

## 20.6 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 20.16) 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.

### 20.6.1 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.

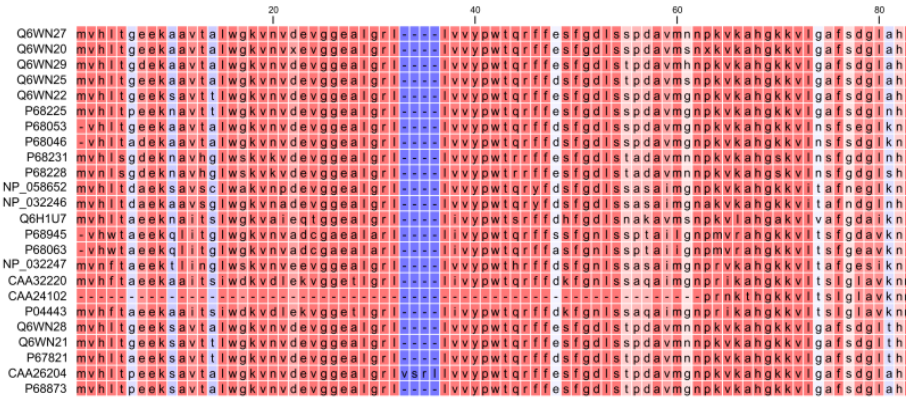



Figure 20.16: *The tabular format of a multiple alignment of 24 Hemoglobin protein sequences. Sequence names appear at the beginning of each row and the residue position is indicated by the numbers at the top of the alignment columns. The level of sequence conservation is shown on a color scale with blue residues being the least conserved and red residues being the most conserved.*

### 20.6.2 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

Presently, the most exciting development in multiple alignment methodology is the construction of *statistical alignment* algorithms [Hein, 2001], [Hein et al., 2000]. These algorithms employ a scoring function which incorporates the underlying phylogeny and use an explicit stochastic model of molecular evolution which makes it possible to compare different solutions in a statistically rigorous way. The optimization step, however, still relies on dynamic programming and practical use of these algorithms thus awaits further developments.

### Creative Commons License

All CLC bio's scientific articles are licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 2.5 License. You are free to copy, distribute, display, and use the work for educational purposes, under the following conditions: You must attribute the work in its original form and "CLC bio" has to be clearly labeled as author and provider of the work. You may not use this work for commercial purposes. You may not alter, transform, nor build upon this work.




See http://creativecommons.org/licenses/by-nc-nd/2.5/ for more information on how to use the contents.

# Chapter 21

# Phylogenetic trees

**Contents**


**21.1 Phylogenetic tree features . . . 455**
**21.2 Create Trees . . . 456**
21.2.1 K-mer Based Tree Construction . . . 457
21.2.2 Create tree . . . 458
21.2.3 Model Testing . . . 460
21.2.4 Maximum Likelihood Phylogeny . . . 462
21.2.5 Bioinformatics explained . . . 465
**21.3 Tree Settings . . . 470**
21.3.1 Minimap . . . 471
21.3.2 Tree layout . . . 471
21.3.3 Node settings . . . 473
21.3.4 Label settings . . . 474
21.3.5 Background settings . . . 475
21.3.6 Branch layout . . . 475
21.3.7 Bootstrap settings . . . 477
21.3.8 Metadata . . . 477
21.3.9 Node right click menu . . . 478
**21.4 Metadata and phylogenetic trees . . . 480**
21.4.1 Table Settings and Filtering . . . 481
21.4.2 Add or modify metadata . . . 481
21.4.3 Unknown metadata values . . . 483
21.4.4 Selection of specific nodes . . . 483


## 21.1 Phylogenetic tree features

Phylogenetics describes the taxonomical 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 21.2.5 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.

## 21.2 Create Trees

For a given set of aligned sequences (see section 20.1) 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 21.2.5). 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 21.2.5). 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 21.2.4 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 21.2.3) 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 21.2.5).

### 21.2.1 K-mer Based Tree Construction

The "K-mer Based Tree Construction" 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 21.1):

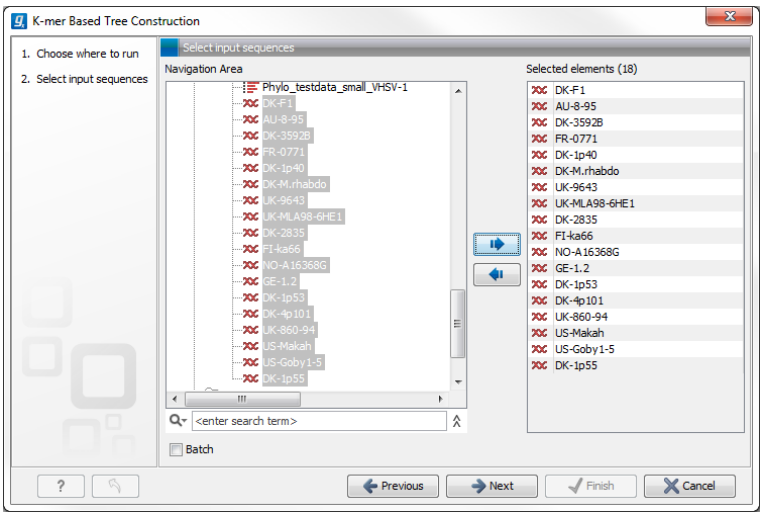



Figure 21.1: *Creating a tree with K-mer based tree construction. Select sequences.*

Next, select the construction method, specify the k-mer length and select a distance measure for tree construction (figure 21.2):

- **Tree construction**

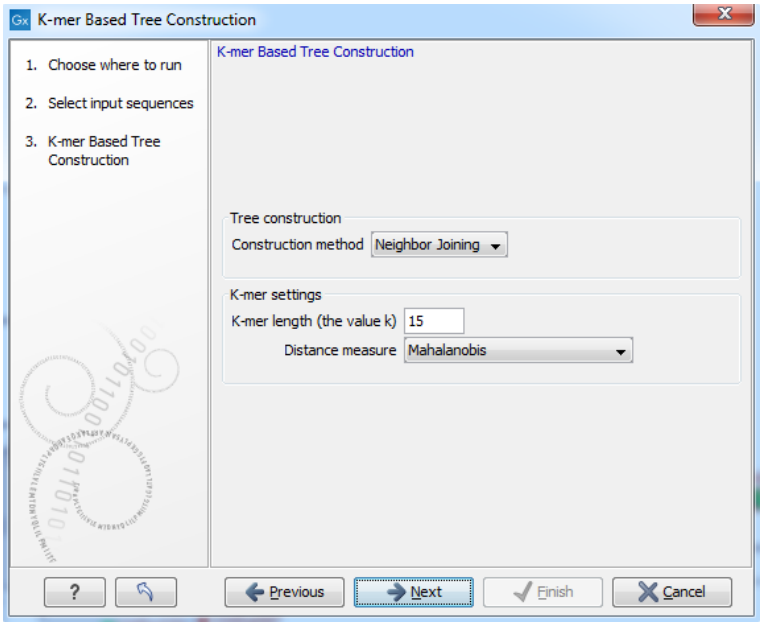



Figure 21.2: *Creating a tree with K-mer based tree construction. Select construction method, specify the k-mer length and select a distance measure.*

  - **Tree construction method** The user is asked to specify which distance-based method to use for tree construction. There are two options (see section 21.2.5):
    - The **UPGMA** method. Assumes constant rate of evolution.
    - The **Neighbor Joining** method. Well suited for trees with varying rates of evolution.

- **K-mer settings**
  - **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 21.2.5 for further details.

### 21.2.2 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 21.3:

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 21.4 shows the parameters that can be set for this distance-based tree creation:

- Tree construction (see section 21.2.5)

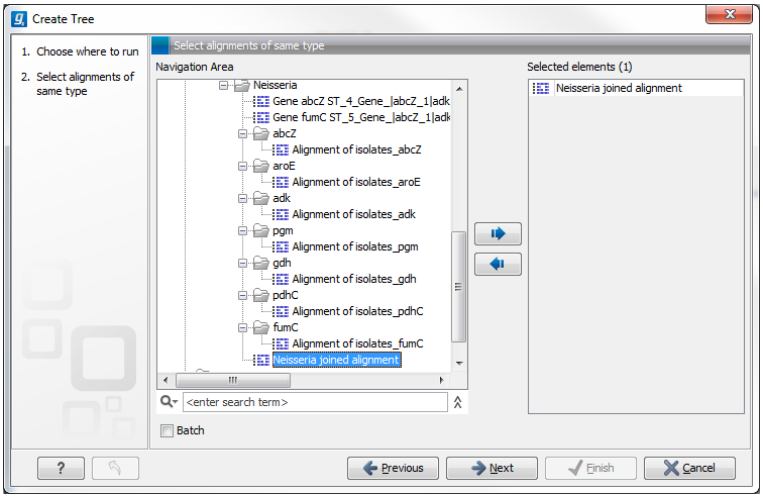



Figure 21.3: *Creating a tree.*

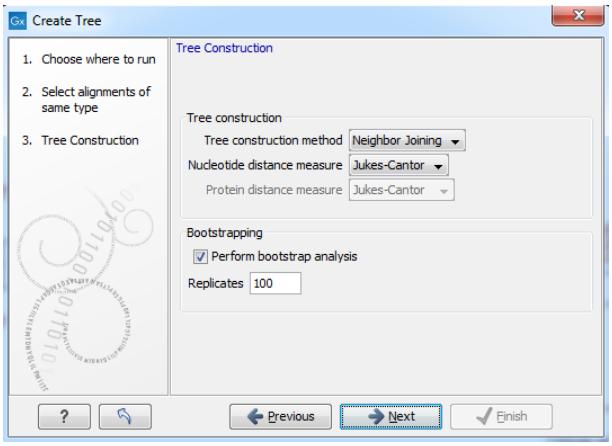



Figure 21.4: *Adjusting parameters for distance-based methods.*

  - Tree construction method
    - The **UPGMA** method. Assumes constant rate of evolution.
    - 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 21.2.5). 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 21.2.5.

### 21.2.3 Model Testing

As the "Model Testing" tool can help identify the best substitution model (21.2.5) to be used for "Maximum Likelihood Phylogeny" tree construction, it is recommended to do "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 21.5):

Specify the parameters to be used for model testing (figure 21.6):

- **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.

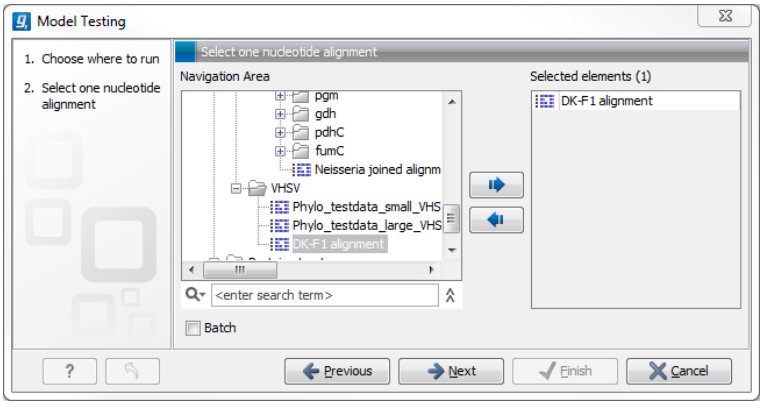



Figure 21.5: *Select alignment for model testing.*

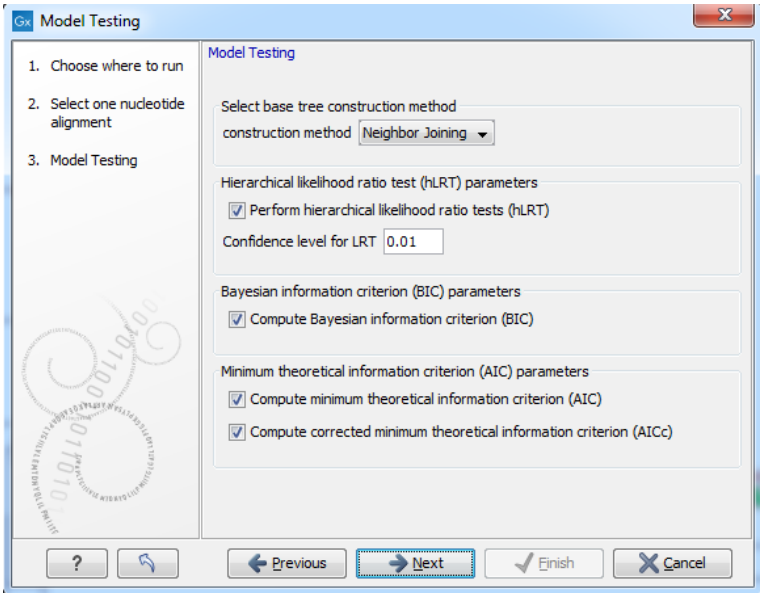



Figure 21.6: *Specify parameters for model testing.*

  - **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.

### 21.2.4 Maximum Likelihood Phylogeny

To generate a maximum likelihood based phylogenetic tree:

**Toolbox** | **Classical Sequence Analysis (  )** | **Alignments and Trees (  )**| **Maximum Likelihood Phylogeny (  )**

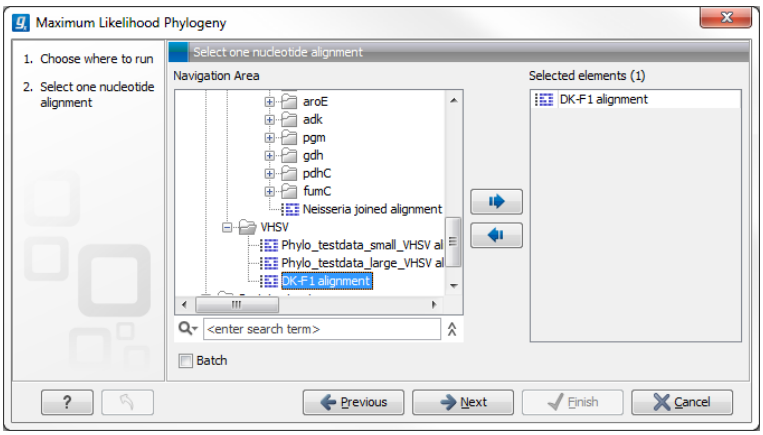



Figure 21.7: *Select the alingment for tree construction*

The following parameters can be set for the maximum likelihood based phylogenetic tree (see figure 21.8):

- **Starting tree**
  - **Construction method** Specify the tree construction method which should be used to create the initial tree. There are two possibilities:

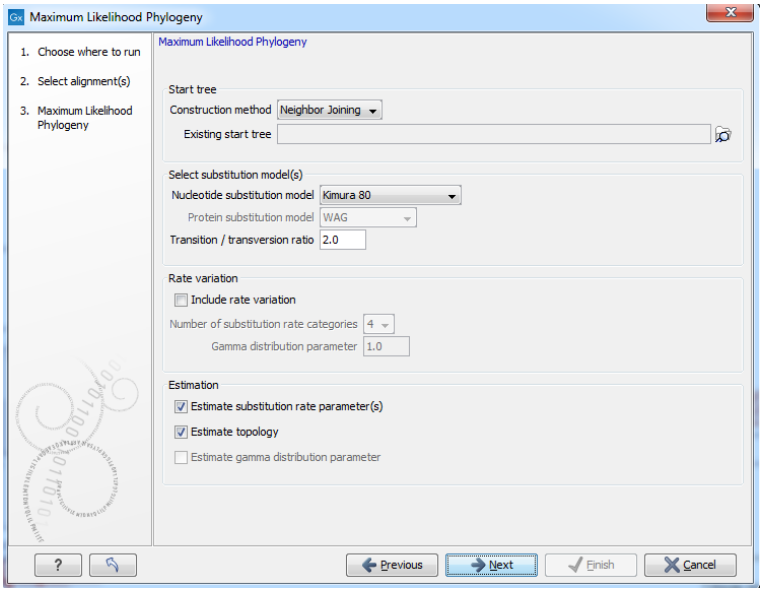



Figure 21.8: *Adjusting parameters for maximum likelihood phylogeny*

  - Neighbor Joining
  - UPGMA

- **Exiting 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 use the browser function to identify 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 21.2.5.

  - **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 21.9).

- **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 21.2.5). 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.

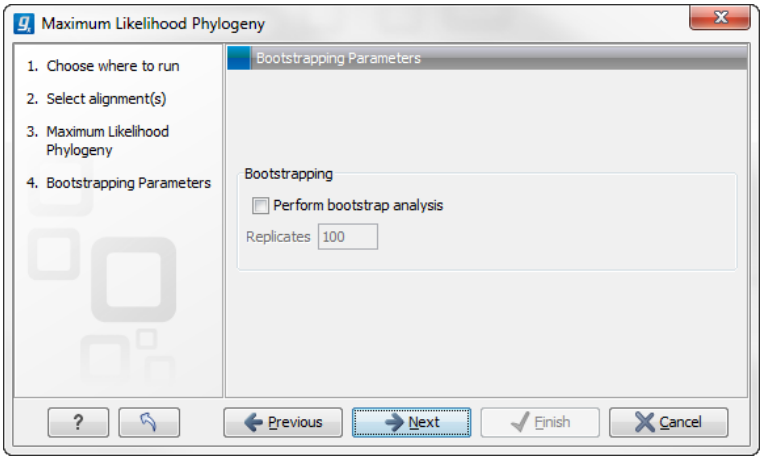



Figure 21.9: *Adjusting parameters for ML phylogeny*

### 21.2.5 Bioinformatics explained

#### The phylogenetic tree

The evolutionary hypothesis of a phylogeny can be graphically represented by a phylogenetic tree.

Figure 21.10 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 taxonomical units. In this case, the terminal nodes (also called leaves or tips of the tree) represent extant species of *Hominidae* and are the *operational taxonomical units* (OTUs). The internal nodes, which here represent extinct common ancestors of the great apes, are termed *hypothetical taxonomical units* since they are not directly observable.

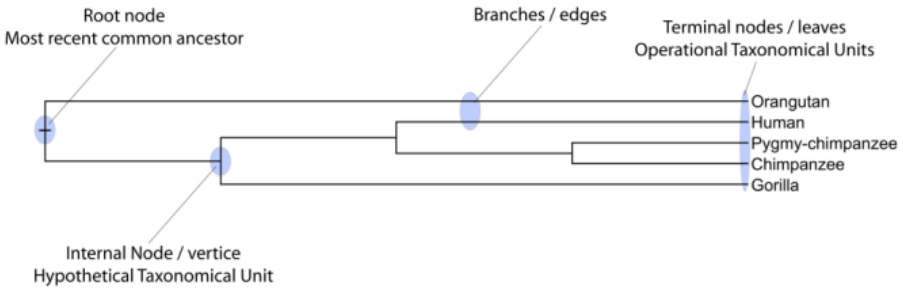



Figure 21.10: *A proposed phylogeny of the great apes (Hominidae). Different components of the tree are marked, see text for description.*

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 21.10 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.

### Modern usage of phylogenies

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 $\sigma_j$ 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 [Hï¿½hl 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 21.2.5). 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 and to help users choose the correct model for a data, set the "Model Testing" tool (see section 21.2.3) can be used to test a range of different models for nucleotide input 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.

## 21.3 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 preferred tree layout settings** (user defined tree settings) can be saved and applied via the top right **Save Tree Settings** (figure 21.11). 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.

**Tree Settings** contains the following categories:

- Minimap

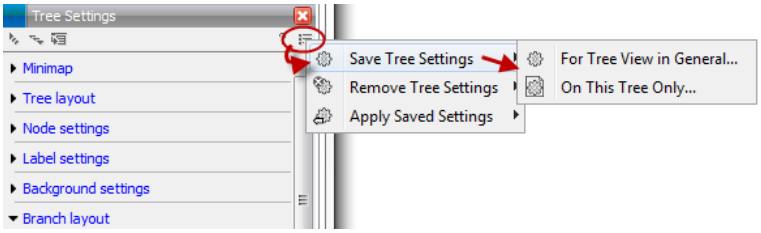



Figure 21.11: *Save, remove or apply preferred layout settings.*

- Tree layout
- Node settings
- Label settings
- Background settings
- Branch layout
- Bootstrap settings
- Metadata

### 21.3.1 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 21.12). To navigate the tree using the Minimap, click on the Minimap with the mouse and move the grey square around within the Minimap.

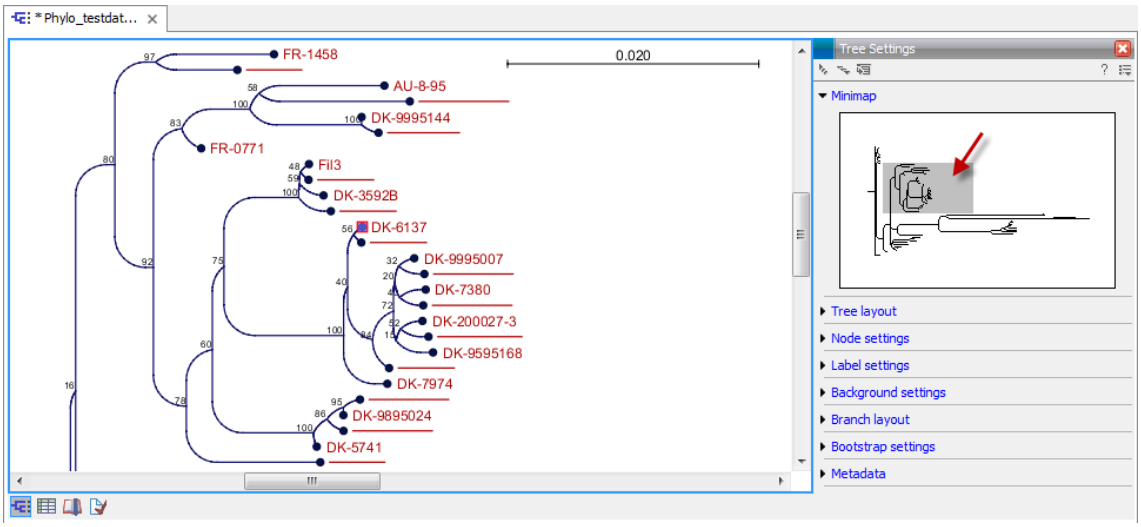



Figure 21.12: *Visualization of a phylogenetic tree. The grey square in the Minimap shows the part of the tree that is shown in the View Area.*

### 21.3.2 Tree layout

The **Tree Layout** can be adjusted in the Side Panel (figure 21.13).

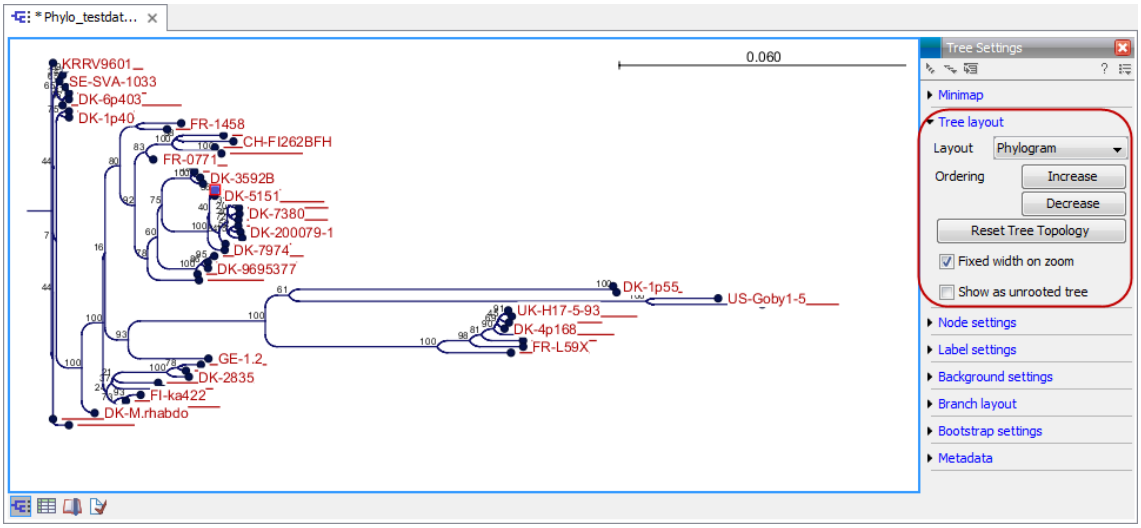



Figure 21.13: *The tree layout can be adjusted in the Side Panel. Five different layouts can be selected and the node order can be changed to increasing or decreasing. The tree topology and node order can be reverted to the original view with the button labeled "Reset Tree Topology".*

- **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 21.14) or **Decreasing**.
- **Reset Tree Topology** Resets to the default tree topology and node order (see figure 21.14).
- **Ordering** The nodes can be ordered after the branch length; either **Increasing** (shown in figure 21.14) or **Decreasing**.
- **Reset Tree Topology** Resets to the default tree topology and node order (see figure 21.14).
- **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.

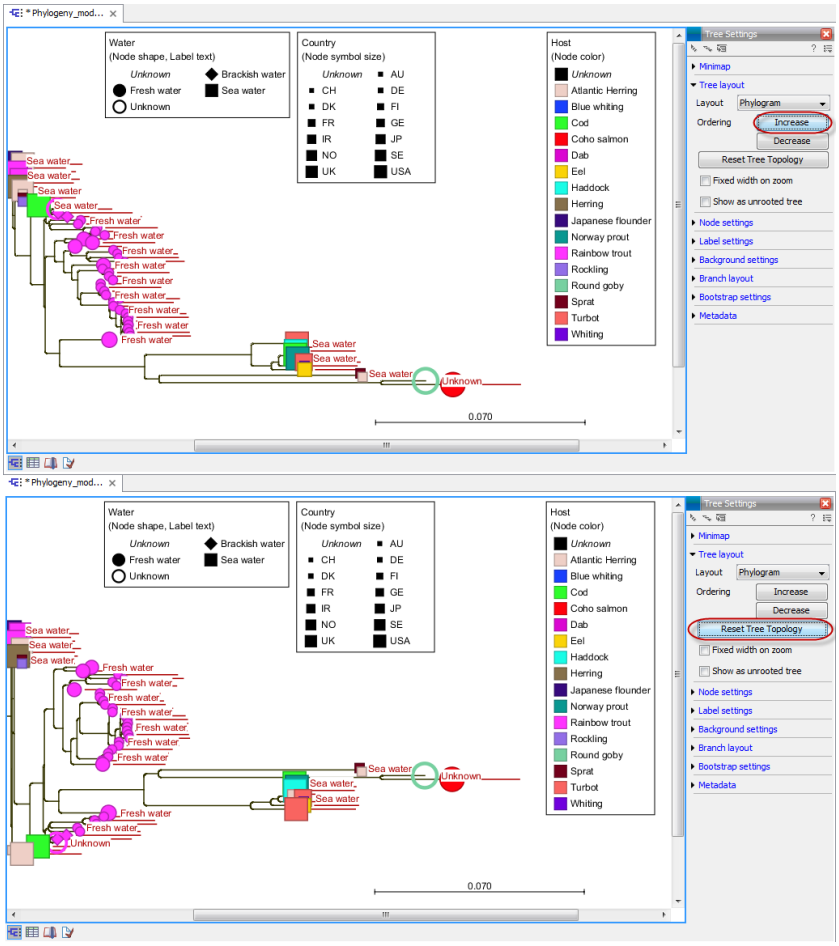



Figure 21.14: *The tree layout can be adjusted in the Side Panel. The top part of the figure shows a tree with increasing node order. In the bottom part of the figure the tree has been reverted to the original tree topology.*

### 21.3.3 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 21.15.

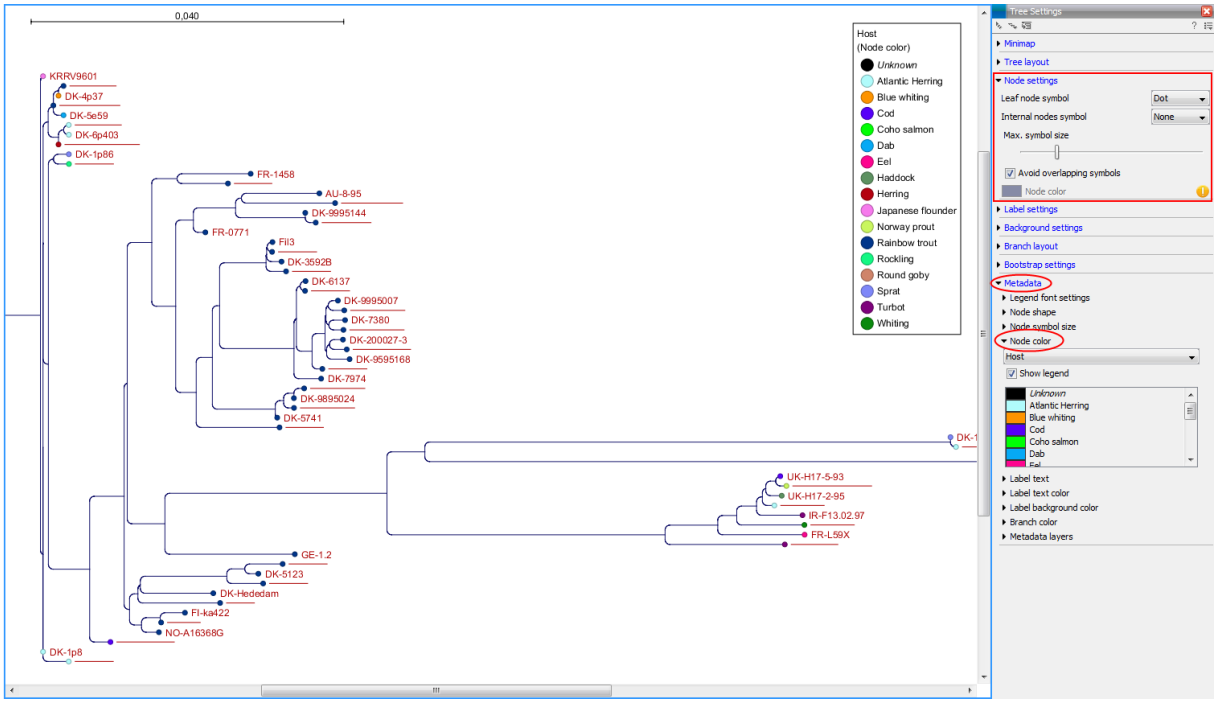



Figure 21.15: *The Node Layout settings. Node color is specified by metadata and is therefore inactive in this example.*

### 21.3.4 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 21.16).
- **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 21.3.9).
- **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.

When working with big trees there is typically not enough space to show all labels. As illustrated in figure 21.16, only some of the labels are shown. The hidden labels are illustrated with thin horizontal lines (figure 21.17).

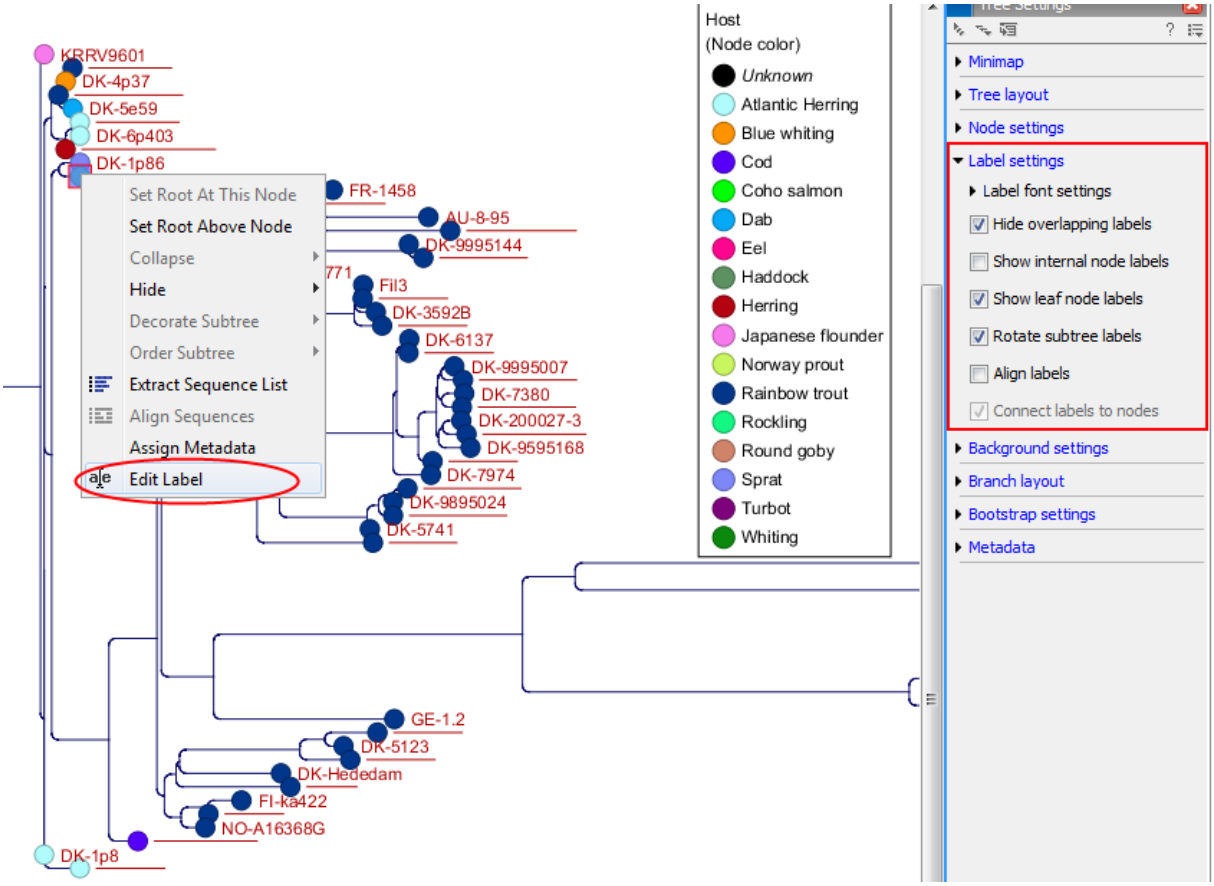



Figure 21.16: *"Edit label" in the right click menu can be used to customize the label text. The way node labels are displayed can be controlled through the labels settings in the right side panel.*

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 21.17 and figure 21.18). 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 21.3.5).

**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.

### 21.3.5 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.

### 21.3.6 Branch layout

- **Branch length font settings** Specify/adjust font type, size and typography (Bold, Italic or normal).
- **Line color** Select the default line color.

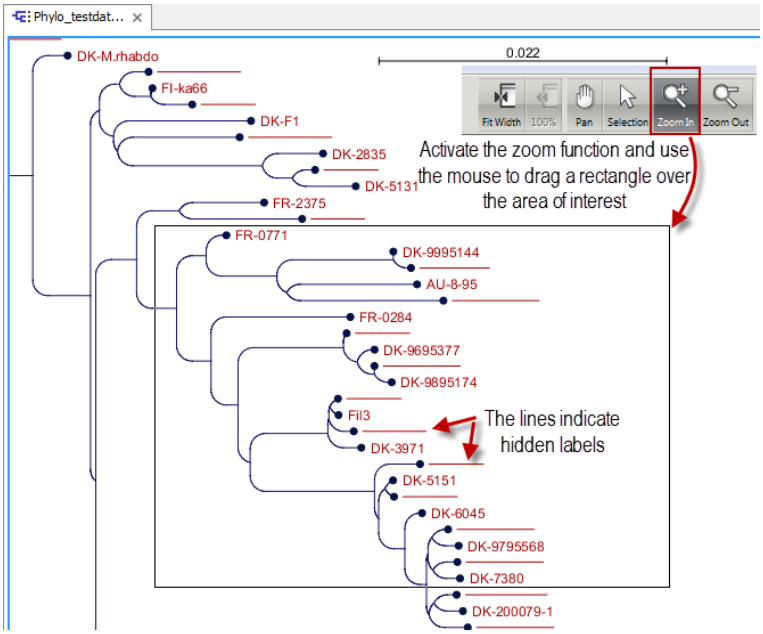



Figure 21.17: *The zoom function in the upper right corner of CLC Genomics Workbench can be used to zoom in on a particular region of the tree. When the zoom function has been activated, use the mouse to drag a rectangle over the area that you wish to zoom in at.*

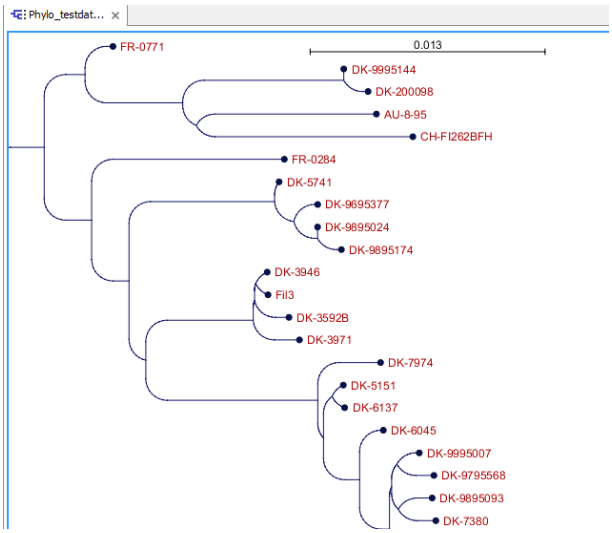



Figure 21.18: *After zooming in on a region of interest more labels become visible. In this example all labels are now visible.*

- **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 21.19.

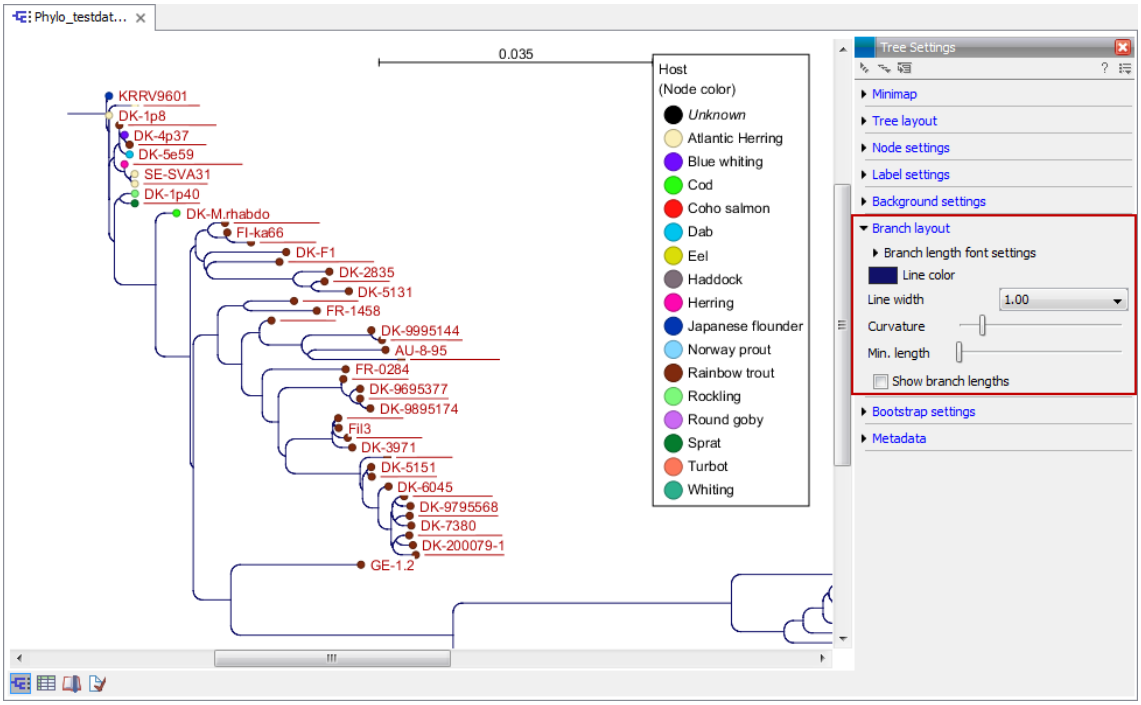



Figure 21.19: *Branch Layout settings.*

### 21.3.7 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.

- **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.

### 21.3.8 Metadata

Metadata associated with a phylogenetic tree (described in detail in section 21.4) 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 21.20.
- **Label text color** The label text can be colored and used to visualize metadata (see figure 21.20).
- **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 21.15).

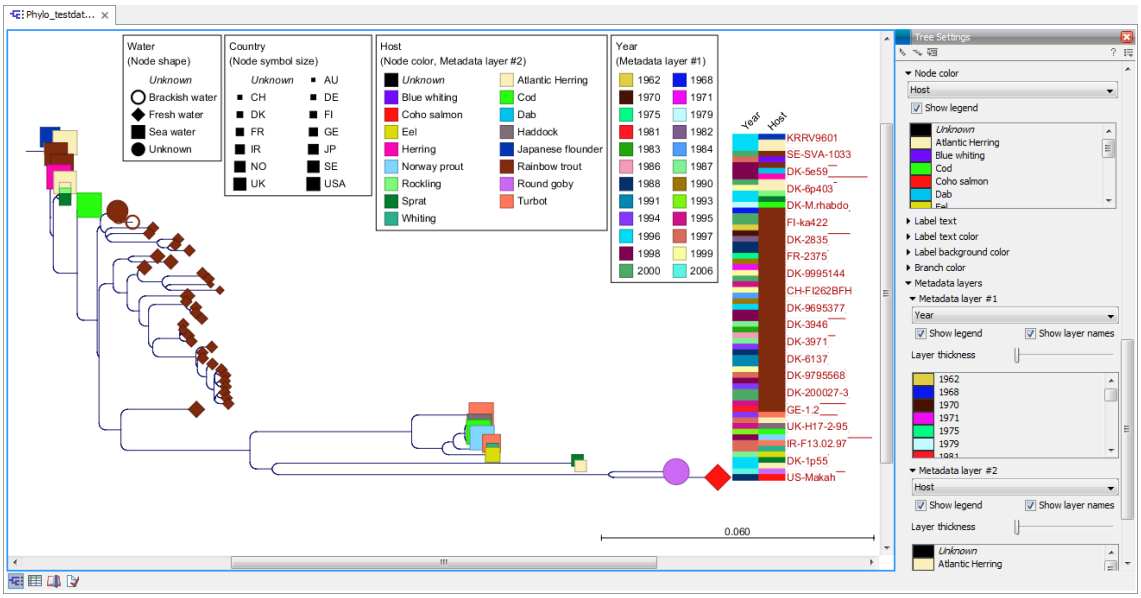



Figure 21.20: *Different types of metadata kan be visualized by adjusting node size, shape, and color. Two color-code metadata layers (Year and Host) are shown in the right side of the tree.*

### 21.3.9 Node right click menu

Additional options for layout and extraction of subtree data are available when right clicking the nodes (figure 21.16):

- **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.

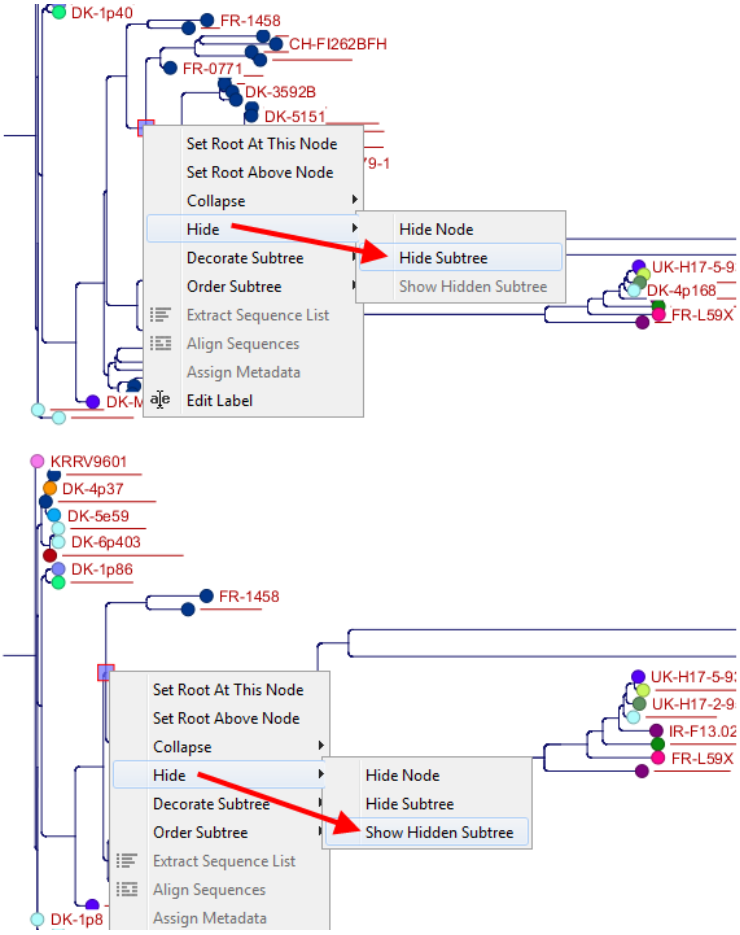



Figure 21.21: *A subtree can be hidden by selecting "Hide Subtree" and is shown again when selecting "Show Hidden Subtree" on a parent node.*

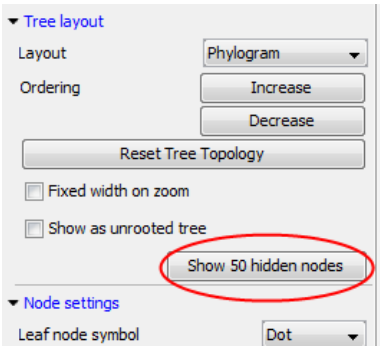



Figure 21.22: *When hiding nodes, a new button labeled "Show X hidden nodes" appears in the Side Panel under "Tree Layout". When pressing this button, all hidden nodes are brought back.*

- **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.
- **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 21.21). When hiding nodes, a new button appears labeled "Show X hidden nodes" in the Side Panel under "Tree Layout" (figure 21.22). 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.
- **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.
- **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**

## 21.4 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 21.23).

Additional metadata can be associated with a tree by clicking the **Import Metadata** button. This will open up the dialog shown in figure 21.24.

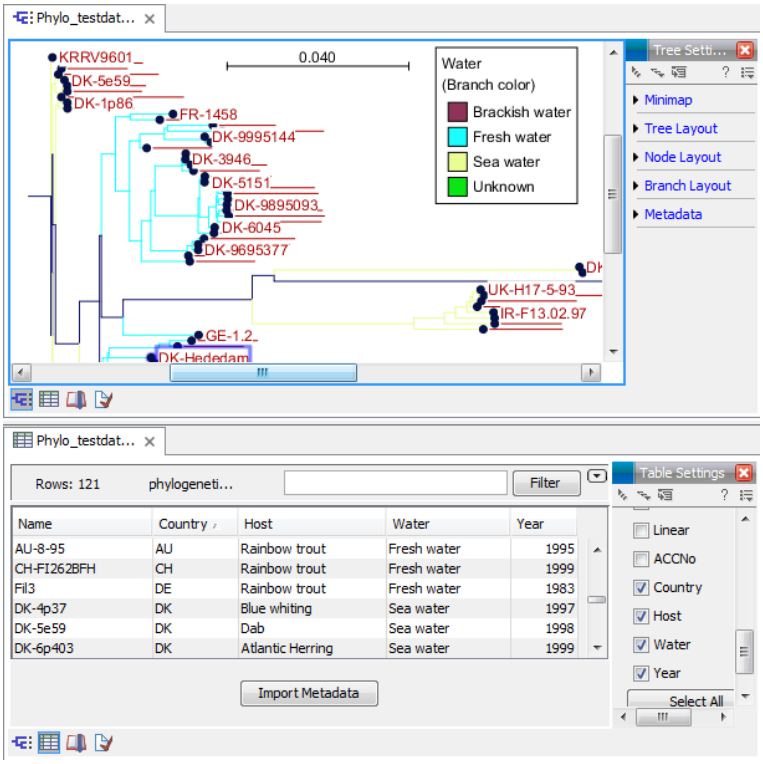



Figure 21.23: *Tabular metadata that is associated with an existing tree shown in a split view.*

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 21.24. 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.

### 21.4.1 Table Settings and Filtering

How to use the metadata table (see figure 21.25):

- **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 the CLC Genomics Workbench manual, Appendix D, Filtering tables).

### 21.4.2 Add or modify metadata

It is possible to add and modify metadata from both the tree view and the table view.

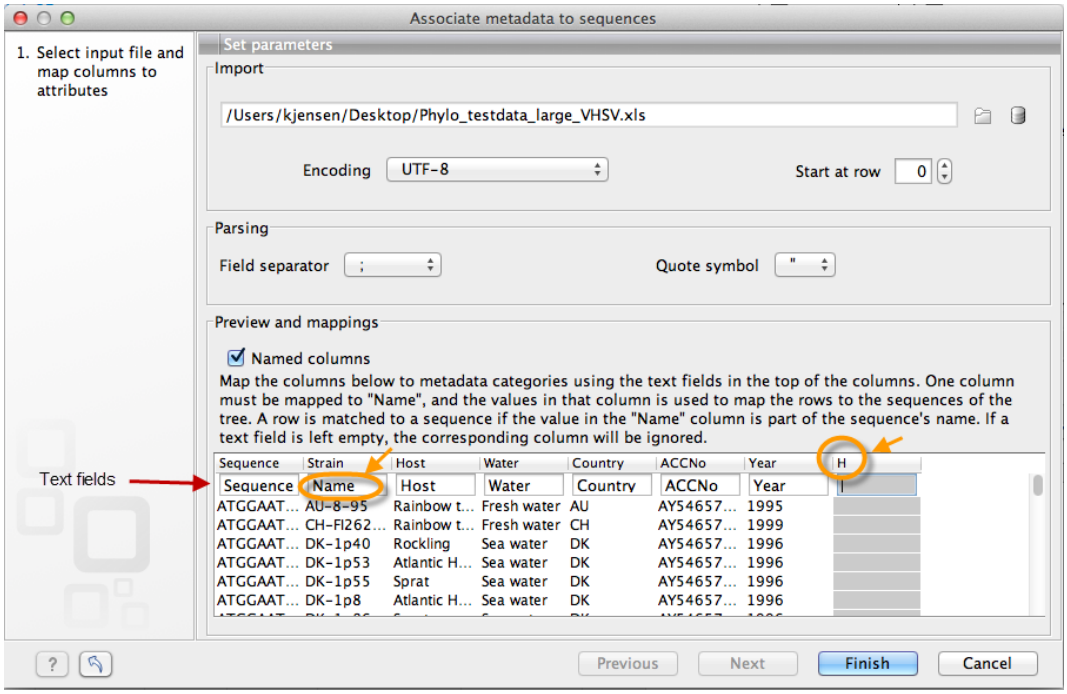



Figure 21.24: *Import of metadata for a tree. The second column named "Strain" is choosen as the common denominator by entering "Name" in the text field of the column. The column labeled "H" is ignored by not assigning a column heading to this column.*

Metadata can be added and edited in the metadata table by using the following right click options (see figure 21.26):

- **Assign Metadata** The right click option "Assign Metadata" can be used for four purposes.
  - 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*" (see an example in figure 21.27). To edit multiple entries at once, select multiple rows in the table, right click a selected cell in the column

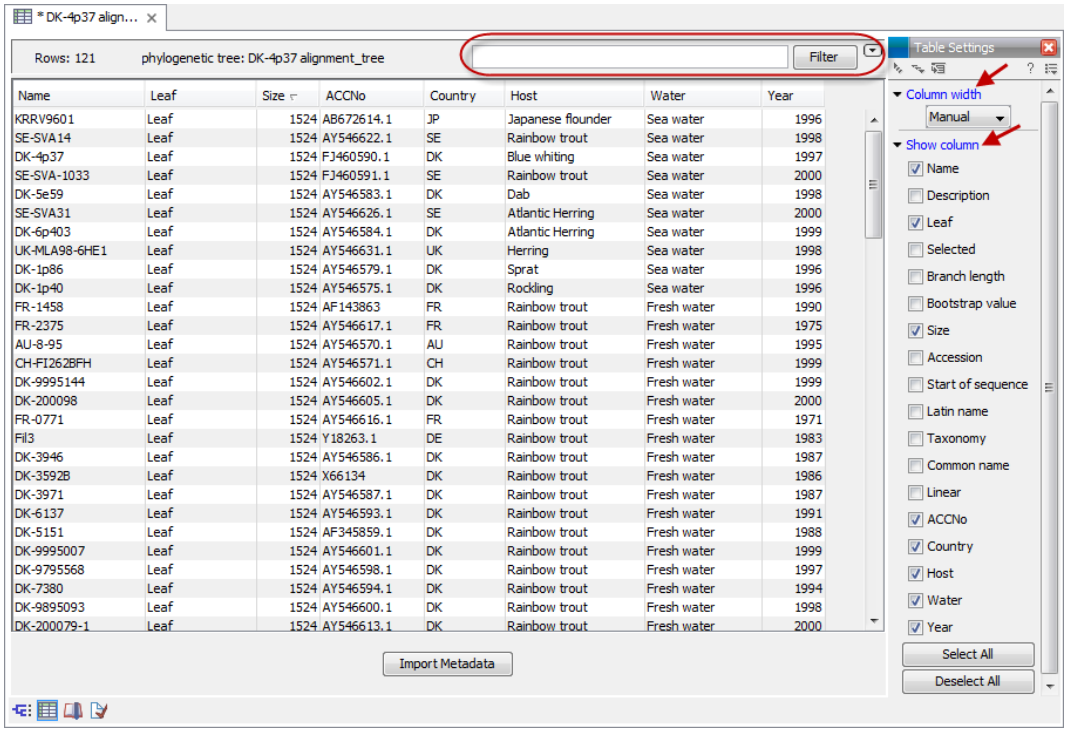



Figure 21.25: *Metadata table. The column width can be adjusted manually or automatically. Under "Show column" it is possible to select which columns should be shown in the table. Filtering using specific criteria can be performed (this is described in the CLC Genomics Workbench manual, Appendix D, Filtering tables).*

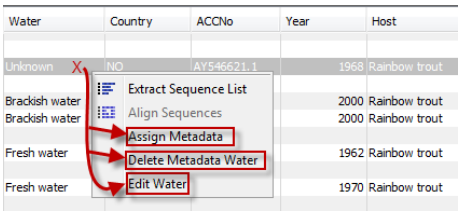



Figure 21.26: *Right click options in the metadata table.*

you want to edit and choose "Edit *column header*". This will change values in all selected rows in the column that was clicked.

### 21.4.3 Unknown metadata values

When visualizing a metadata category where one or more nodes in the tree have undefined values, these nodes will be visualized using a *default* value. This value will always be shown in italics in the top of the legend (see figure 21.28). To remove this entry in the legend, all nodes must have a value assigned in the corresponding metadata category.

### 21.4.4 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:

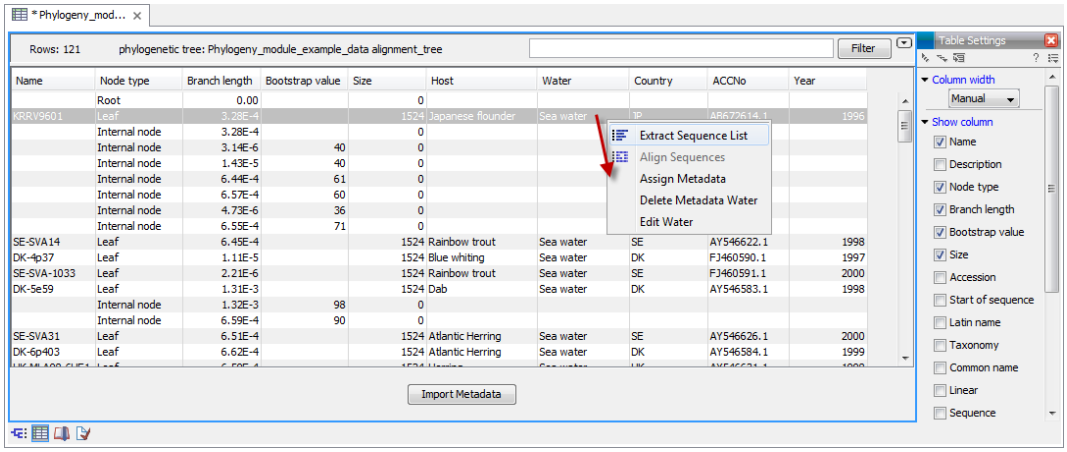



Figure 21.27: *To include an extra metadata column, use the right click option "Assign Metadata", provide "Name" (the column header) and "Value". To modify existing metadata, click on the specific field, select "Edit column header" and provide new value.*

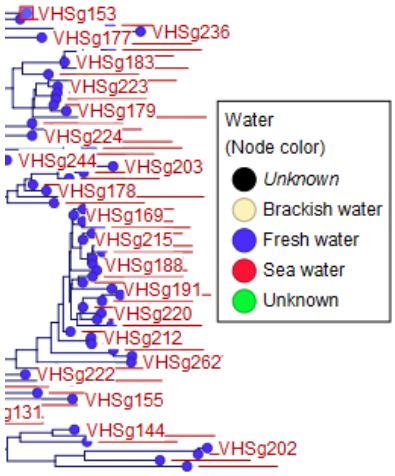



Figure 21.28: *A legend for a metadata category where one or more values are undefined.*

- *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 21.29).
- *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 20.6). Next, change relevant option in the multiple alignment wizard that pops up and click **Finish**. The multiple alignment will now be generated.

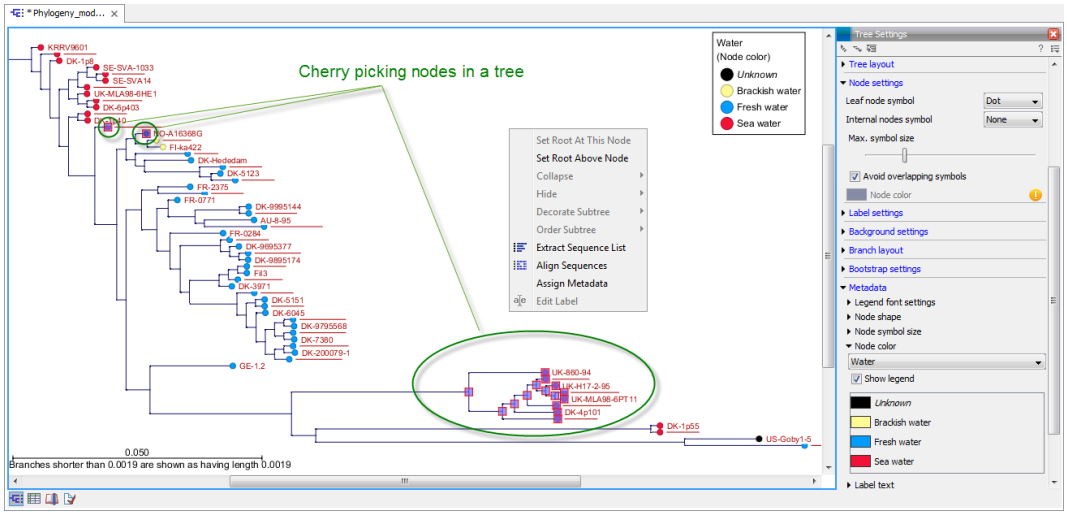



Figure 21.29: *Cherry picking nodes in a tree. The selected leaf sequences can be extracted by right clicking on one of the selected nodes and selecting "Extract Sequence List". It is also possible to Align Sequences directly by right clicking on the nodes or leaves.*

# Chapter 22

# RNA structure

## Contents


**22.1 RNA secondary structure prediction . . . . . 487**
22.1.1 Selecting sequences for prediction . . . . . 487
22.1.2 Structure output . . . . . 488
22.1.3 Partition function . . . . . 489
22.1.4 Advanced options . . . . . 490
22.1.5 Structure as annotation . . . . . 492
**22.2 View and edit secondary structures . . . . . 493**
22.2.1 Graphical view and editing of secondary structure . . . . . 493
22.2.2 Tabular view of structures and energy contributions . . . . . 496
22.2.3 Symbolic representation in sequence view . . . . . 499
22.2.4 Probability-based coloring . . . . . 501
**22.3 Evaluate structure hypothesis . . . . . 501**
22.3.1 Selecting sequences for evaluation . . . . . 501
22.3.2 Probabilities . . . . . 502
**22.4 Structure scanning plot . . . . . 502**
22.4.1 Selecting sequences for scanning . . . . . 503
22.4.2 The structure scanning result . . . . . 505
**22.5 Bioinformatics explained: RNA structure prediction by minimum free energy minimization . . . . . 506**
22.5.1 The algorithm . . . . . 506
22.5.2 Structure elements and their energy contribution . . . . . 508


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

## 22.1 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 22.5.

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`.

### 22.1.1 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 22.1.

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

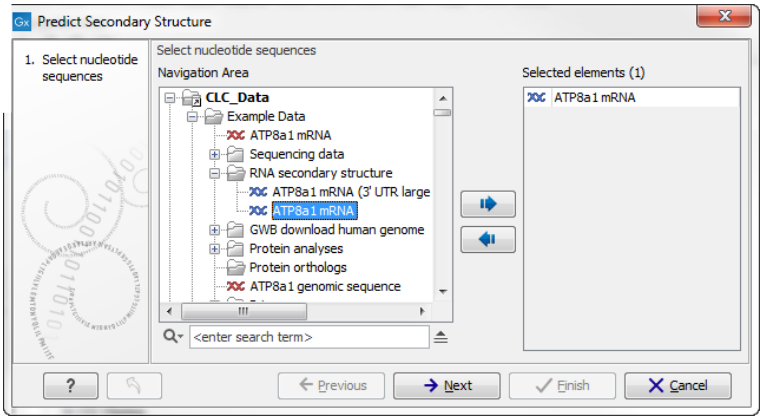



Figure 22.1: *Selecting RNA or DNA sequences for structure prediction (DNA is folded as if it was RNA).*

sequence lists from the selected elements. You can use both DNA and RNA sequences - DNA will be folded as if it were RNA. Click **Next** to adjust secondary structure prediction parameters. Clicking **Next** opens the dialog shown in figure 22.2.

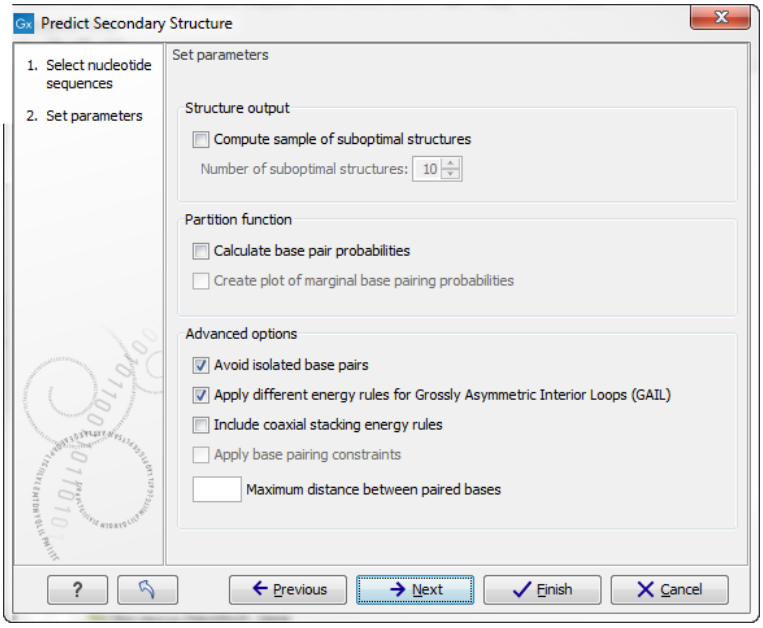



Figure 22.2: *Adjusting parameters for secondary structure prediction.*

### 22.1.2 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 labeled **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].

### 22.1.3 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 22.3.

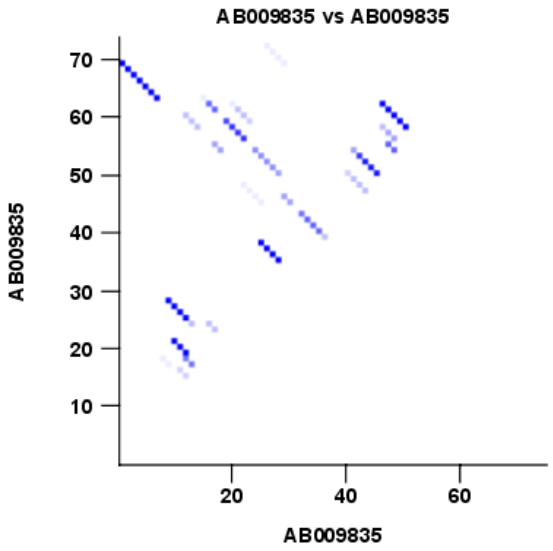



Figure 22.3: *The marginal base pair probability of all possible base pairs.*

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 22.2.3

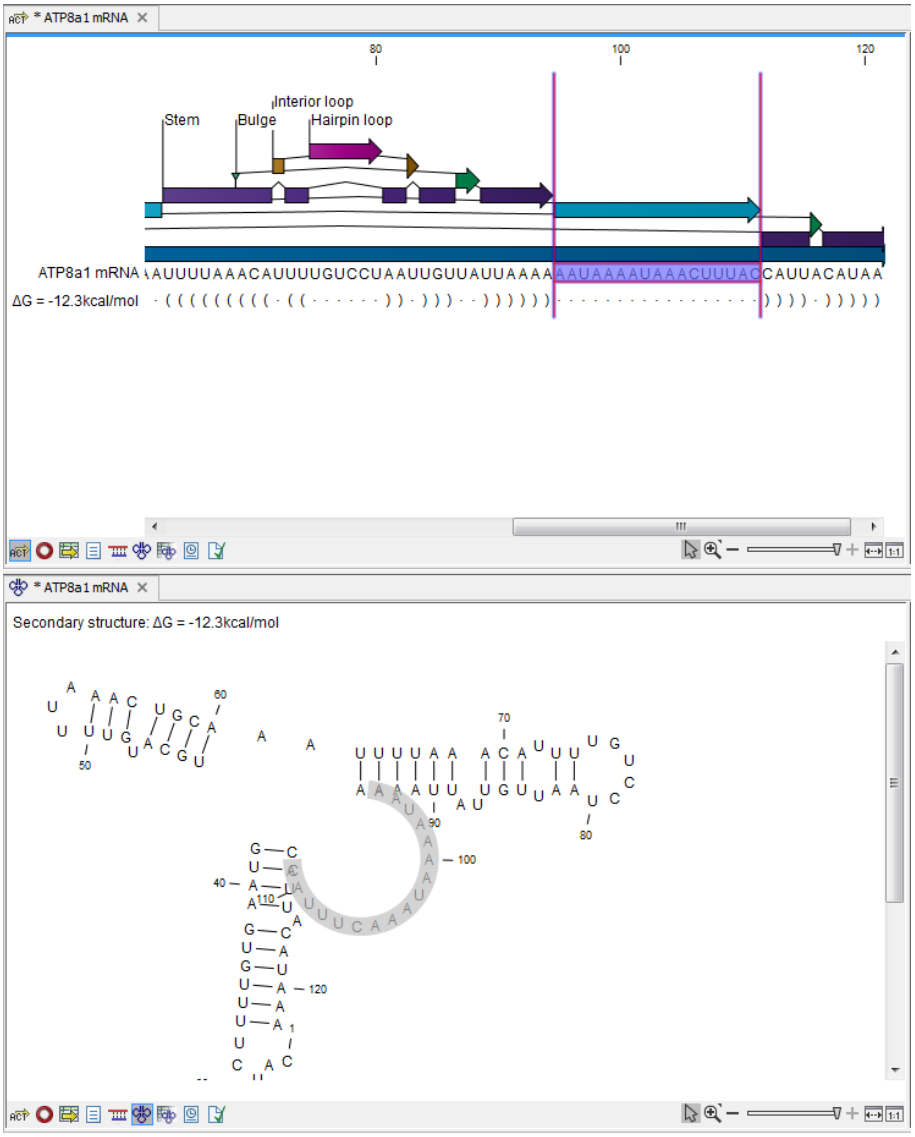



Figure 22.4: *Marginal probability of base pairs shown in linear view (top) and marginal probability of being unpaired shown in the secondary structure 2D view (bottom).*

and also in the secondary structure view. An example is shown in figure 22.4. Furthermore, the marginal probabilities are accessible from tooltips when hovering over the relevant parts of the structure.

### 22.1.4 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 22.1.3).

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 22.5).

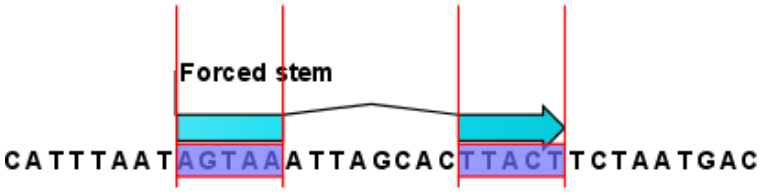



Figure 22.5: *Force a stem of the selected bases.*

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 22.6).

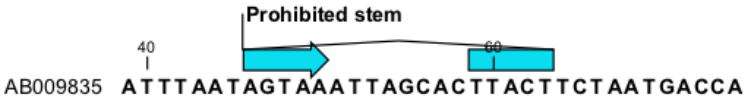



Figure 22.6: *Prohibit the selected bases from forming a stem.*

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:

> **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 22.7.




Figure 22.7: *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.

### 22.1.5 Structure as annotation

You can choose to add the elements of the best structure as annotations (see figure 22.8).

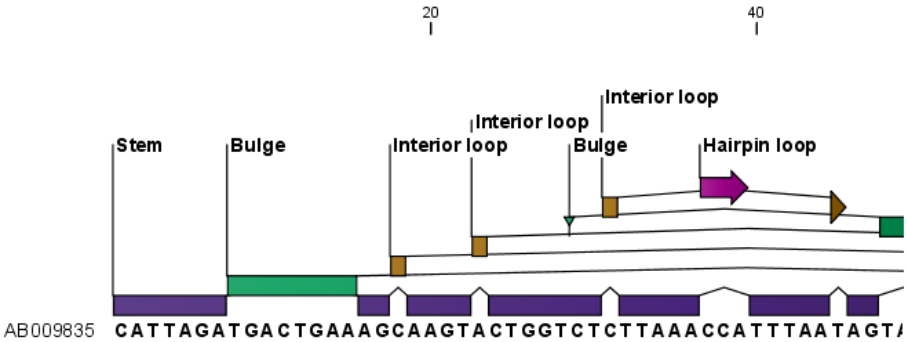



Figure 22.8: *Annotations added for each structure element.*

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 22.2.2.

## 22.2 View and edit secondary structures

When you predict RNA secondary structure (see section 22.1), 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 22.2. See an example in figure 22.8.
- Symbolic representation below the sequence (see section 22.2.3).
- A graphical view of the secondary structure (see section 22.2.1).
- 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 22.2.2.

### 22.2.1 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 22.9.

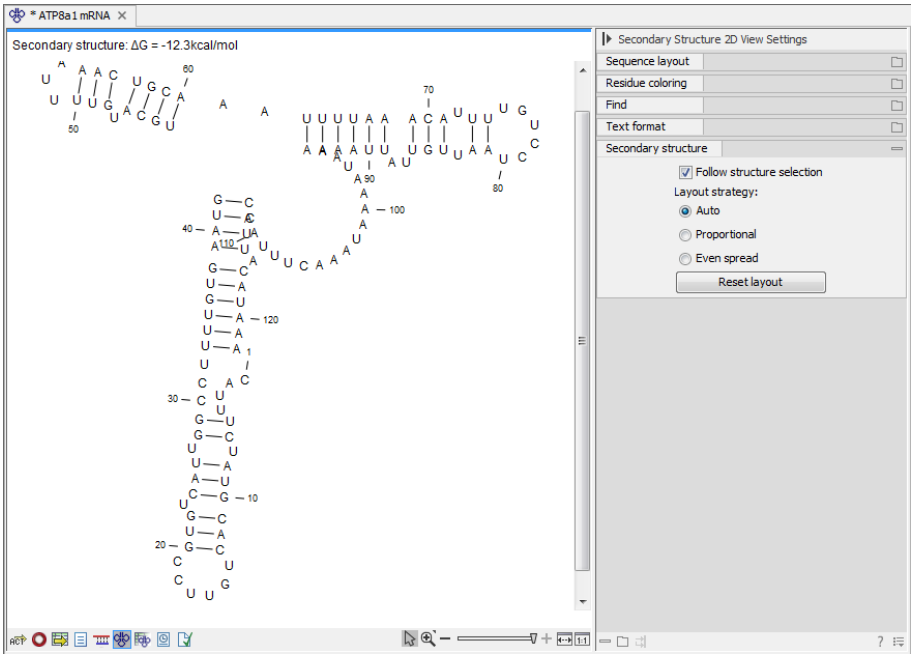



Figure 22.9: *The secondary structure view of an RNA sequence zoomed in.*

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 22.9. 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 10.1.1. 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 22.2.2 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 22.10).
  - **Proportional.** Arc lengths are proportional to the number of residues (see figure 22.11). Nothing is done to prevent overlap.
  - **Even spread.** Stems are spread evenly around loops as shown in figure 22.12.
- **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.

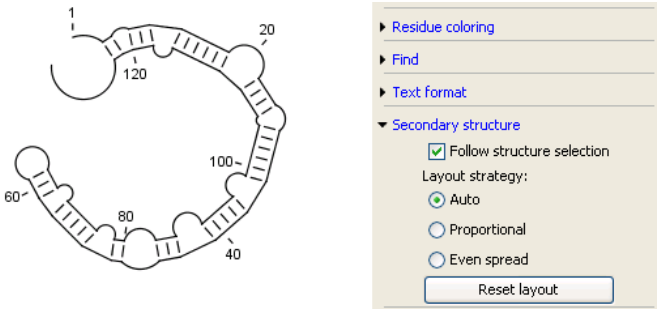



Figure 22.10: *Auto layout. Overlaps are minimized.*

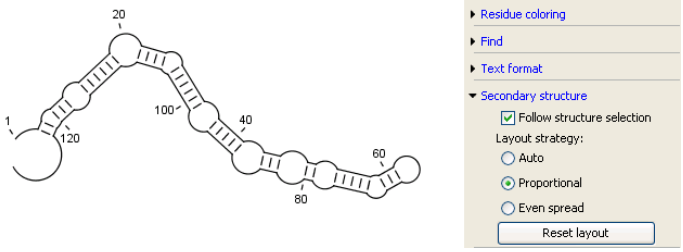



Figure 22.11: *Proportional layout. Length of the arc is proportional to the number of residues in the arc.*

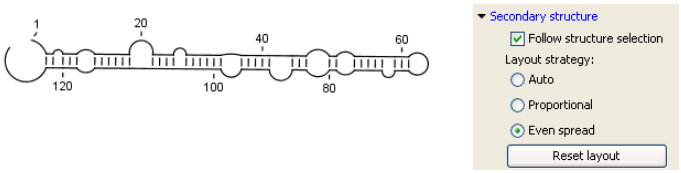



Figure 22.12: *Even spread. Stems are spread evenly around loops.*

### 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 22.13).

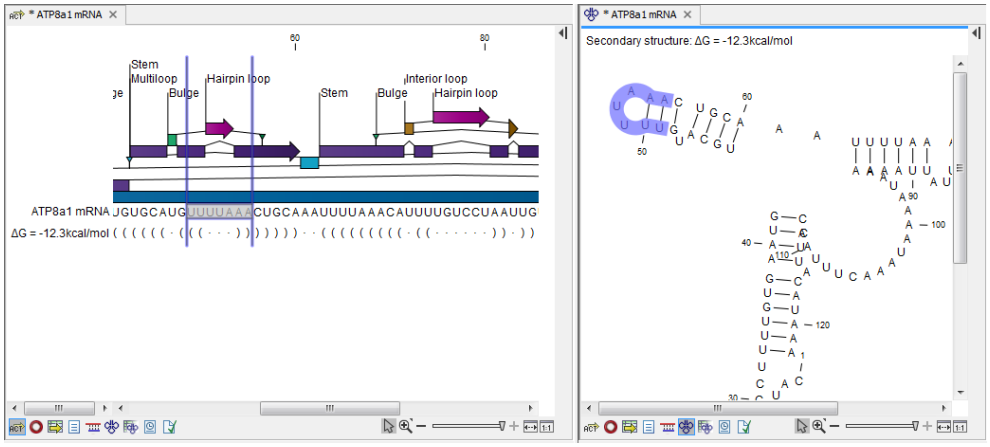



Figure 22.13: *A split view of the secondary structure view and a linear sequence view.*

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. 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 22.14).

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 22.15, the structure shown in figure 22.14 has been modified by dragging with the mouse.

Press **Reset layout** in the **Side Panel** to reset the layout to the way it looked when the structure

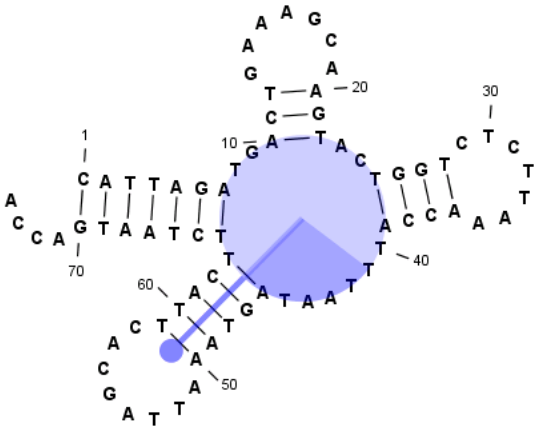


Figure 22.14: *The blue circle represents the anchor point for rotating the substructure.*

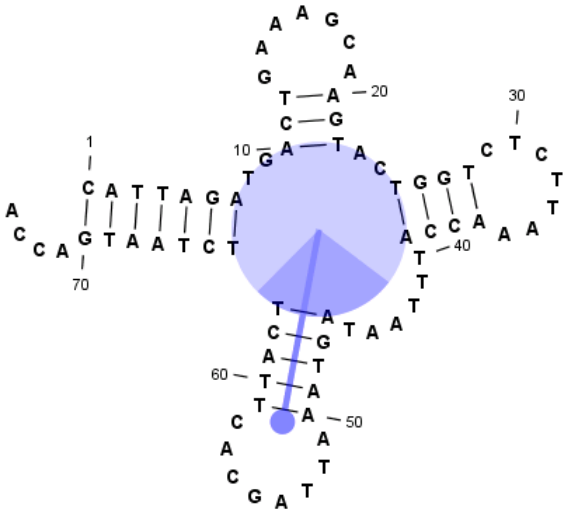


Figure 22.15: *The structure has now been rotated.*

was predicted.

### 22.2.2 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 22.1), 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 22.2.1).
- 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 22.16.

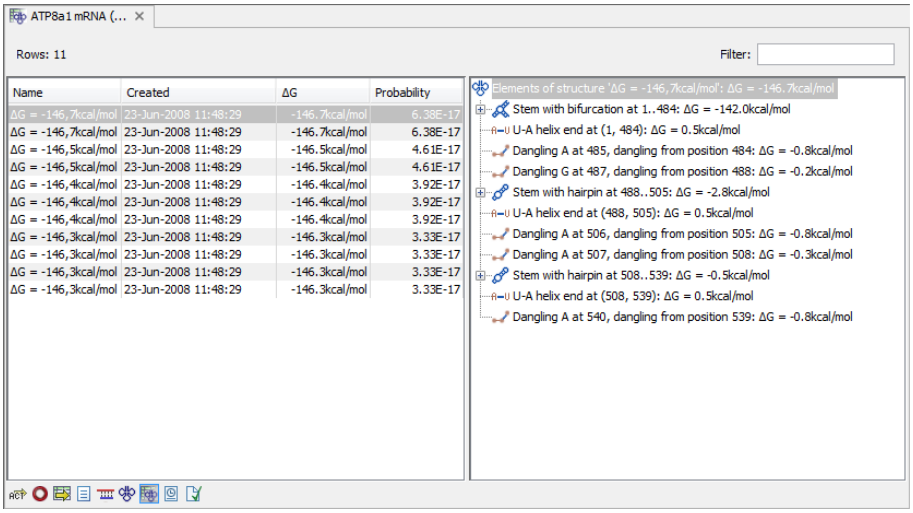



Figure 22.16: *The secondary structure table with the list of structures to the left, and to the right the substructures of the selected structure.*

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 22.5.2). Each substructure 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 $\Delta G$. 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 $\Delta G$. 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 22.17).

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

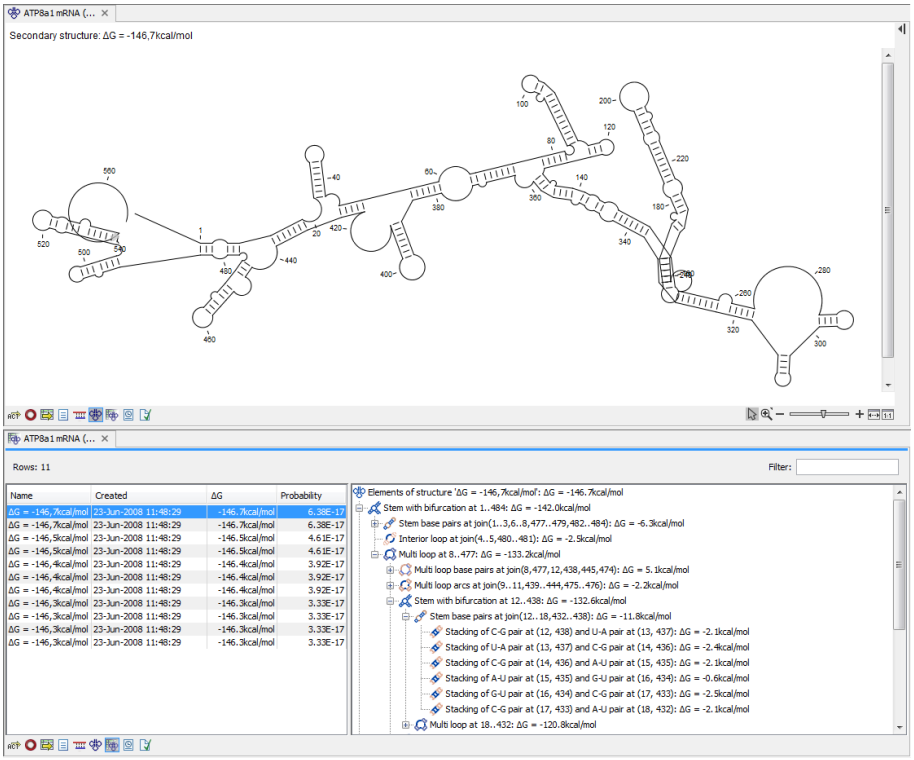



Figure 22.17: *A split view showing a structure table to the right and the secondary structure 2D view to the left.*

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 22.18).

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 22.18 the "Stem with bifurcation" is selected in the table, and this part of the structure is high-lighted in the Secondary Structure 2D view.

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.

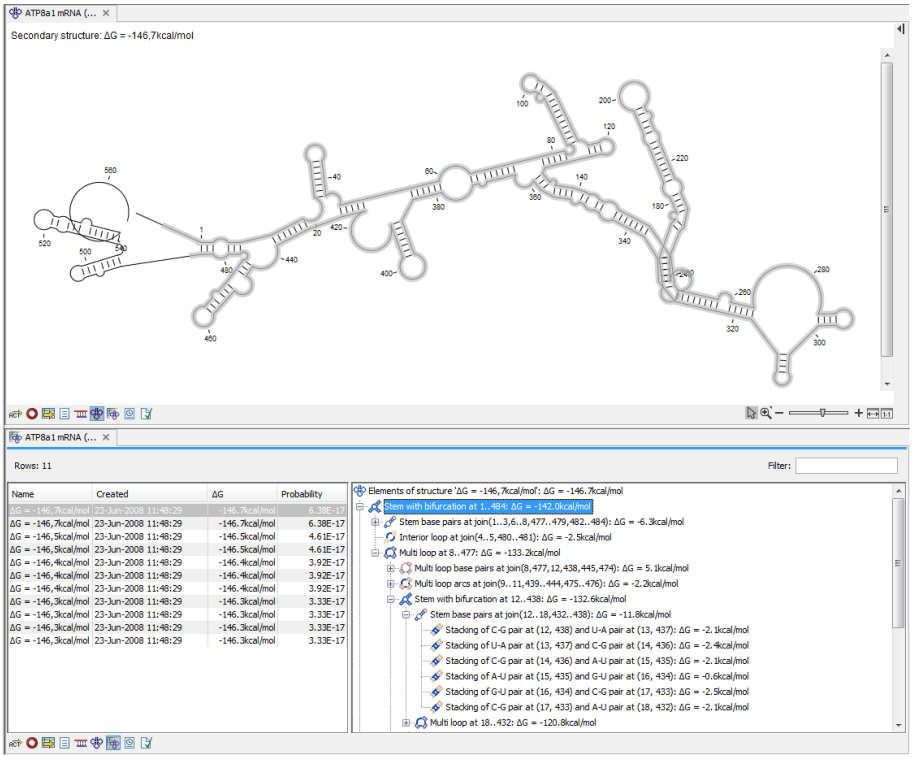



Figure 22.18: *Now the "Stem with bifurcation" node has been selected in the table and a corresponding selection has been made in the view of the secondary structure to the left.*

- **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.

### 22.2.3 Symbolic representation in sequence view

In the **Side Panel** of normal sequence views ( ), 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 22.19).

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.

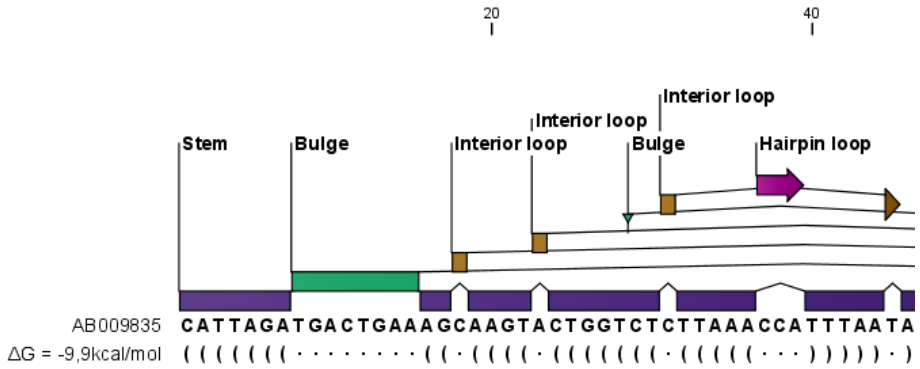



Figure 22.19: *The secondary structure visualized below the sequence and with annotations shown above.*

  - 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 22.19).
- **Unpaired symbol.** How bases which are not part of a base pair should be represented (see figure 22.19).
- **Height.** When you zoom out, this option determines the height of the symbols as shown in figure 22.20 (when zoomed in, there is no need for specifying the height).
- **Base pair probability.** See section 22.2.4 below).

When you zoom in and out, the appearance of the symbols change. In figure 22.19, the view is zoomed in. In figure 22.20 you see the same sequence zoomed out to fit the width of the sequence.

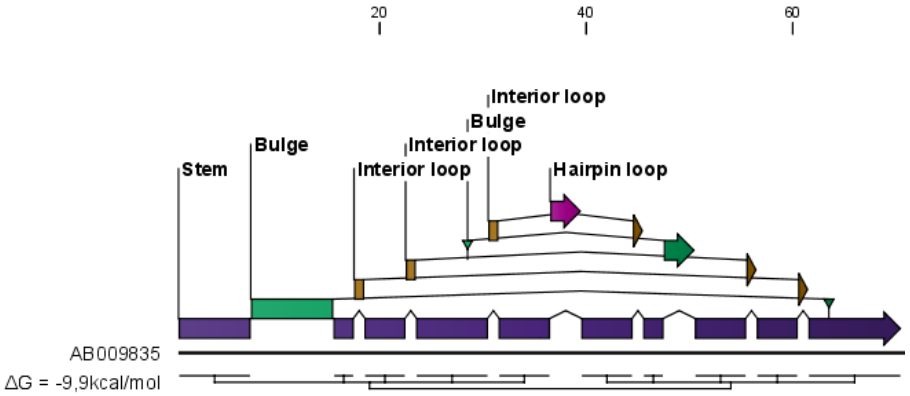



Figure 22.20: *The secondary structure visualized below the sequence and with annotations shown above. The view is zoomed out to fit the width of the sequence.*

### 22.2.4 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 22.4.

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%** () in order to see the coloring.

## 22.3 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 22.1.4. By adding several annotations complex structural hypotheses can be formulated (see 22.21).

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 $PF_{\text{full}}$ 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. 22.21.

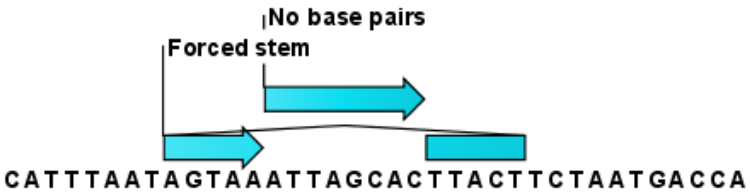



Figure 22.21: *Two constraints defining a structural hypothesis.*

### 22.3.1 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 22.22.

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

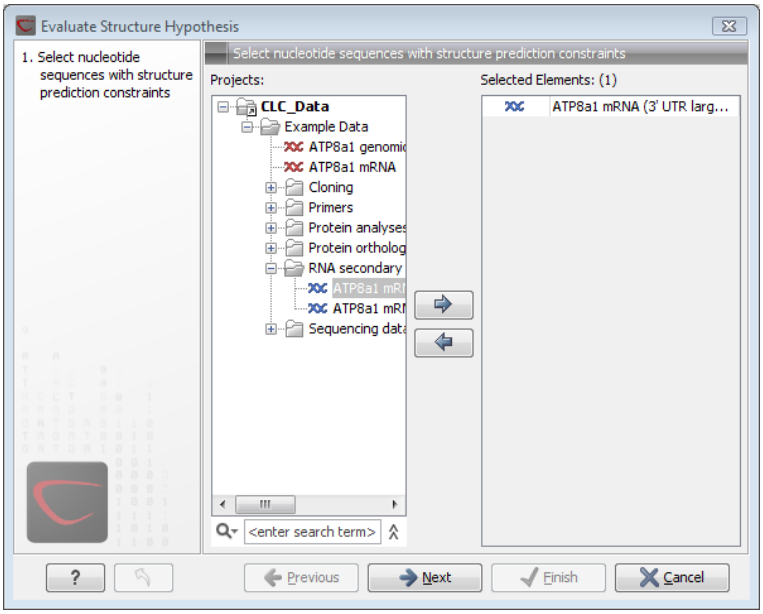



Figure 22.22: *Selecting RNA or DNA sequences for evaluating structure hypothesis.*

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 22.23).

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).
- **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].

### 22.3.2 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 22.24).

## 22.4 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

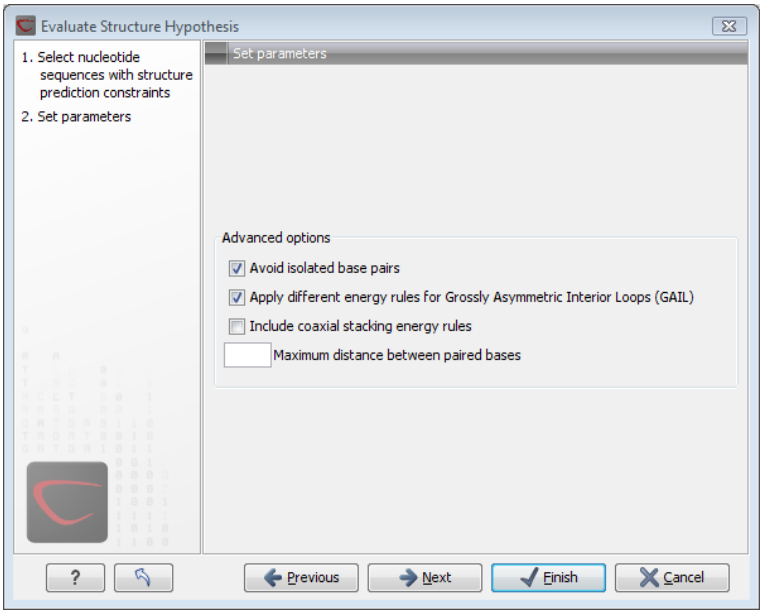



Figure 22.23: *Adjusting parameters for hypothesis evaluation.*

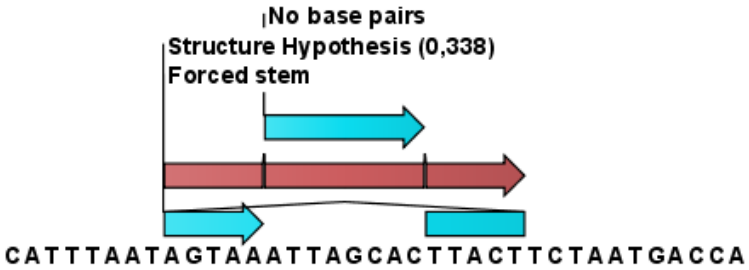



Figure 22.24: *This hypothesis has a probability of 0.338 as shown in the annotation.*

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].

### 22.4.1 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 22.25.

If you have selected sequences before choosing the Toolbox action, they are now listed in

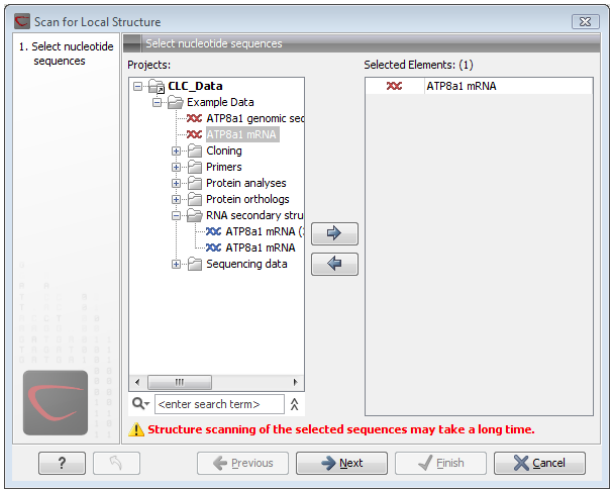



Figure 22.25: *Selecting RNA or DNA sequences for structure scanning.*

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 22.26).

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
- **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.

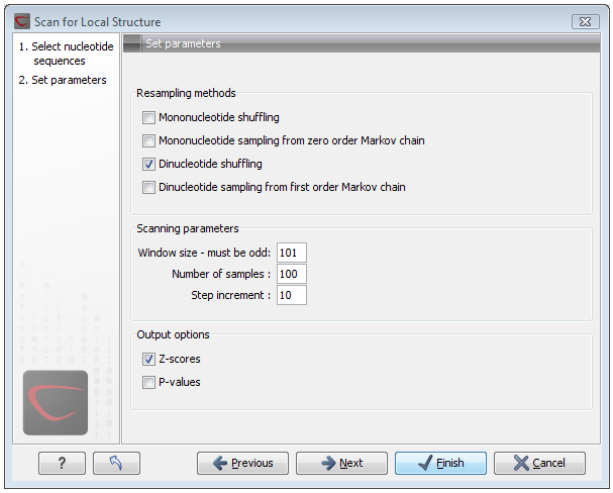



Figure 22.26: *Adjusting parameters for structure scanning.*

### 22.4.2 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 22.27).

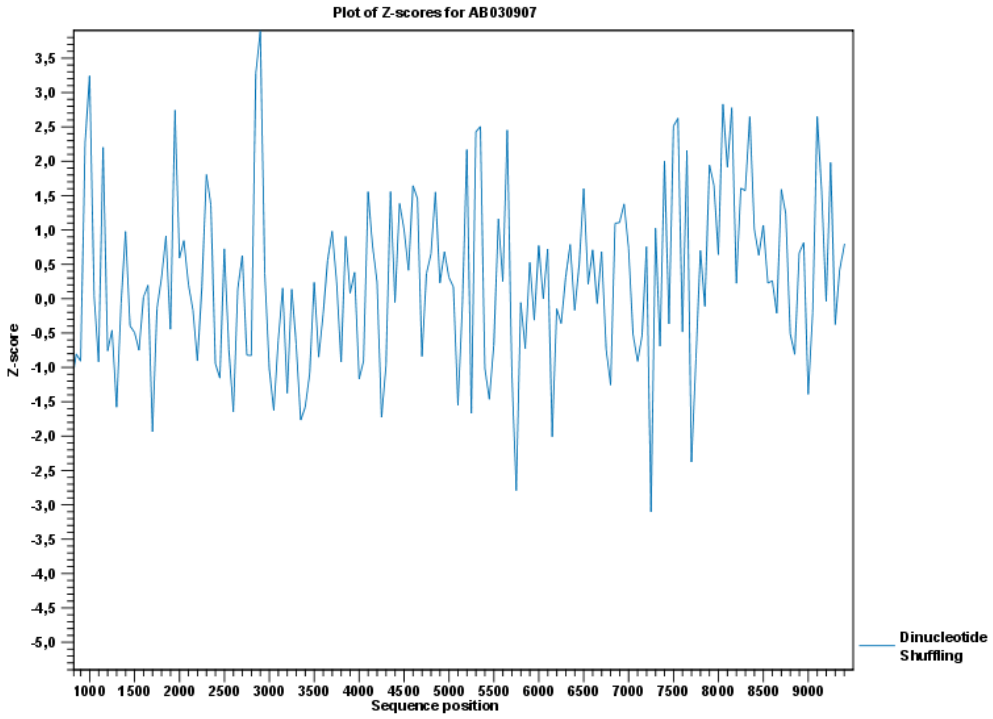



Figure 22.27: *A plot of the Z-scores produced by sliding a window along a sequence.*

## 22.5 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].

### 22.5.1 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 22.28.

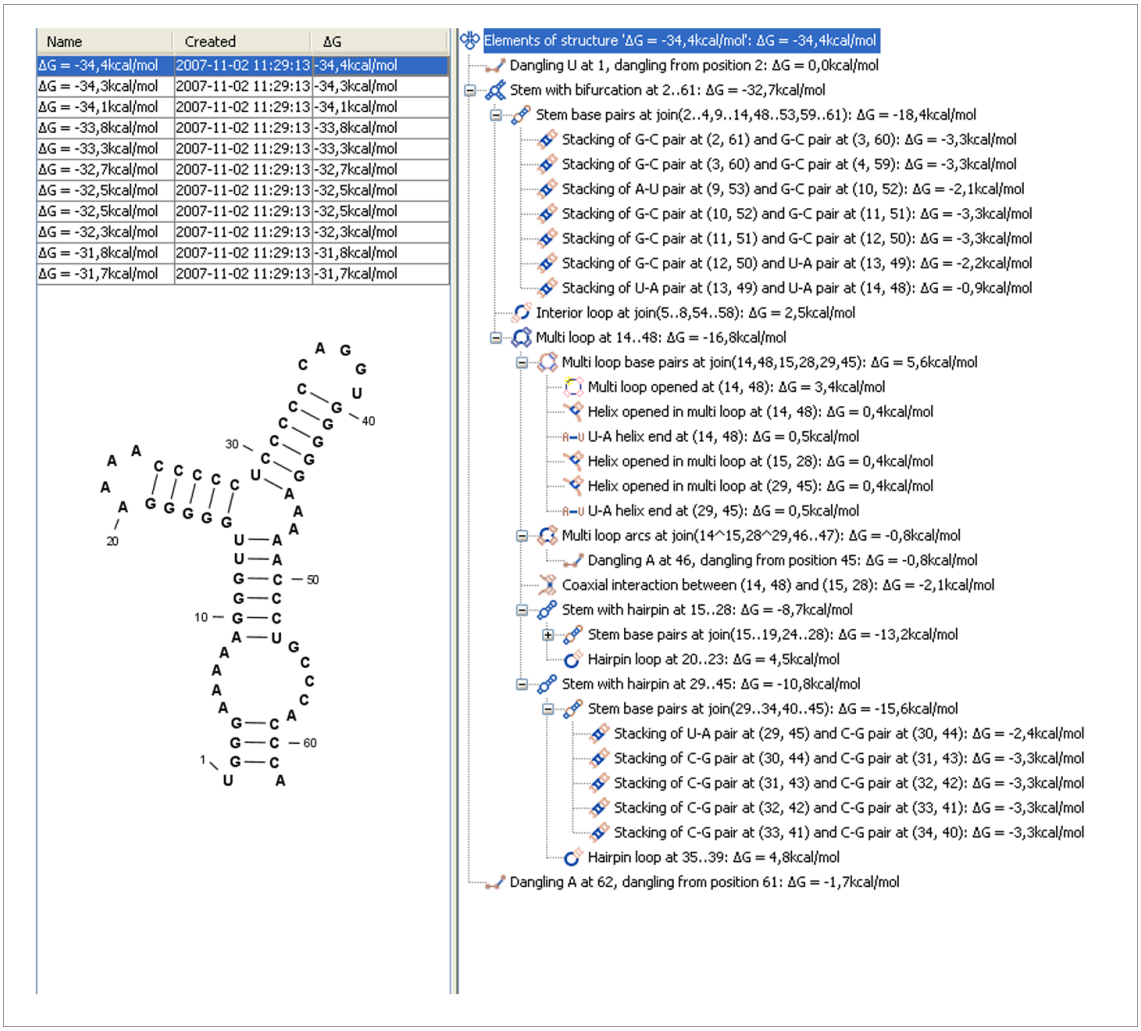



Figure 22.28: *A number of suboptimal structures have been predicted using* **CLC Genomics Workbench** *and are listed at the top left. At the right hand side, the structural components of the selected structure are listed in a hierarchical structure and on the left hand side the structure is displayed.*

### 22.5.2 Structure elements and their energy contribution

In this section, we classify the structure elements defining a secondary structure and describe their energy contribution.

#### 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.

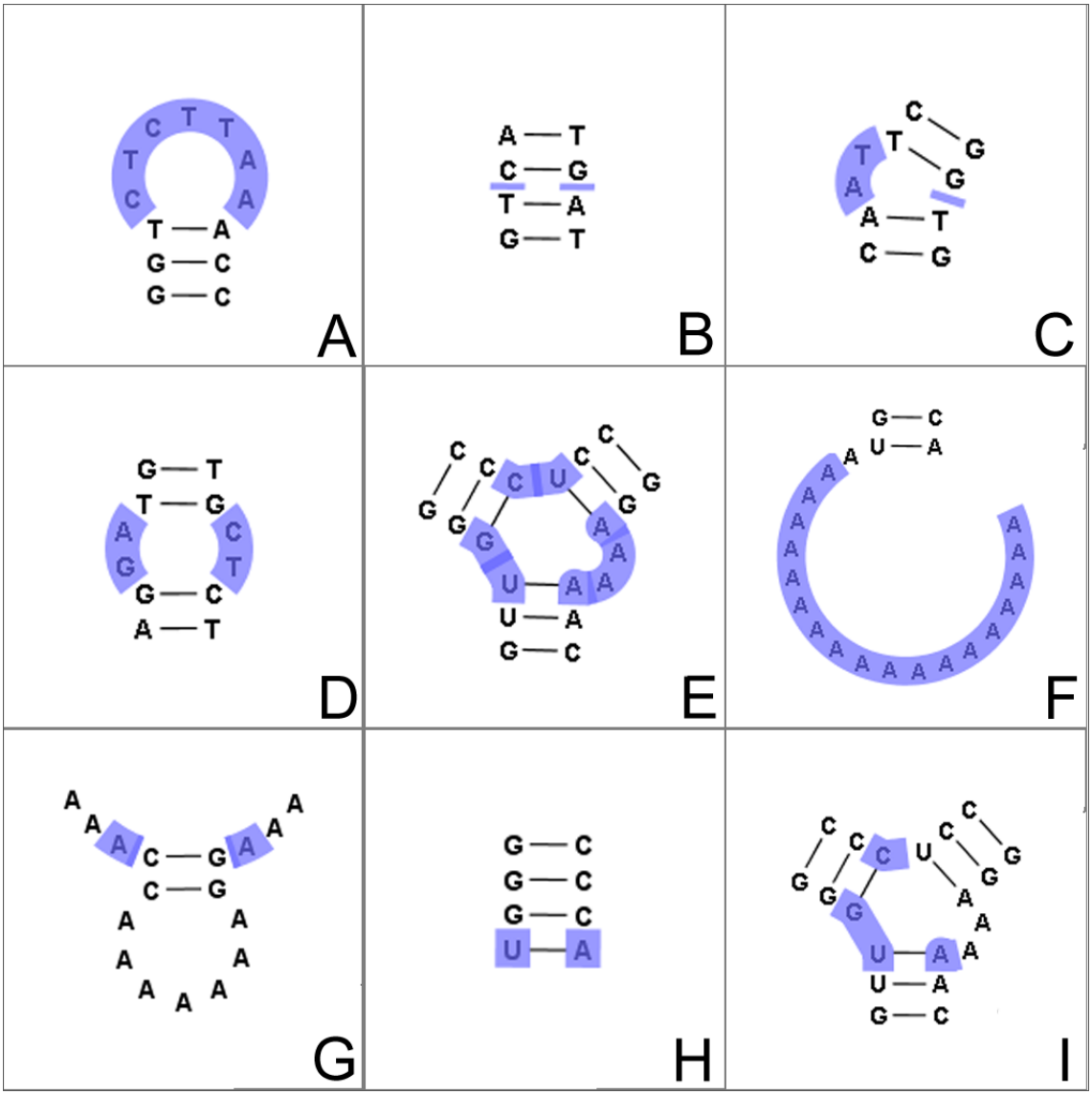



Figure 22.29: *The different structure elements of RNA secondary structures predicted with the free energy minimization algorithm in* **CLC Genomics Workbench**. *See text for a detailed description.*

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 22.29A).

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 22.29B). 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 $\geq 1$ on the other side (see figure 22.29C). 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 $\geq 1$ on both sides (see figure 22.29D). 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 22.29E). 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 22.29F). 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 22.29G). 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 22.29H).
- **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 22.29I). 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 22.30) and it has been shown that this can dramatically increase the fidelity of the secondary structure prediction [Mathews and Turner, 2006].

### Creative Commons License

All CLC bio's scientific articles are licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 2.5 License. You are free to copy, distribute, display, and use the work for educational




Figure 22.30: *Known structural features can be added as constraints to the secondary structure prediction algorithm in* **CLC Genomics Workbench**.

purposes, under the following conditions: You must attribute the work in its original form and "CLC bio" has to be clearly labeled as author and provider of the work. You may not use this work for commercial purposes. You may not alter, transform, nor build upon this work.

SOME RIGHTS RESERVED

See http://creativecommons.org/licenses/by-nc-nd/2.5/ for more information on how to use the contents.

# Part IV

# High-throughput sequencing

# Chapter 23

# Trimming, multiplexing and sequencing quality control

**Contents**


**23.1 Trim Sequences . . . 513**
23.1.1 Quality trimming . . . 514
23.1.2 Adapter trimming . . . 516
23.1.3 Length trimming . . . 522
23.1.4 Trim output . . . 523
**23.2 Demultiplex reads . . . 524**
**23.3 Sequencing data quality control . . . 533**
23.3.1 Report contents . . . 533
23.3.2 Adapters . . . 536
23.3.3 Running the quality control tool . . . 536
**23.4 Merge overlapping pairs . . . 537**
23.4.1 Using quality scores when merging . . . 539
23.4.2 Report of merged pairs . . . 540


## 23.1 Trim Sequences

*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 e.g. stretches of Ns
3. Adapter trimming
4. Base trim to remove a specified number of bases at either 3' or 5' end of the reads

5. 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.

**Adapters**
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. If adapters have not been trimmed off, please do so before proceeding with your analysis. The presence of adapters will lead to misleading results.
If you are working with sequences that still have adapters present, they can be trimmed using the **Trim Sequences** tool provided in the "NGS Core tools" folder in the toolbox.

**Illumina Adapters**
Illumina recently changed their adapter sequences and this may have consequences for the downstream data analysis if the new adapter sequences were used for the sequencing analysis and the old adapter sequences were used for trimming off the adapter sequences.

If you have Illumina sequencing data that have been generated with the new adapter sequences and have not been trimmed or have been trimmed incompletely, the adapter sequences can be removed within the *CLC Genomics Workbench* using the Illumina adapter sequences that can be found here:

http://support.illumina.com/downloads/illumina-customer-sequence-letter.html

To start trimming:

**Toolbox** | **NGS Core Tools** (  ) | **Trim Sequences** (  )

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**.

### 23.1.1 Quality trimming

This opens the dialog displayed in figure 23.1 where you can specify parameters for quality trimming.

The following parameters can be adjusted in the dialog:

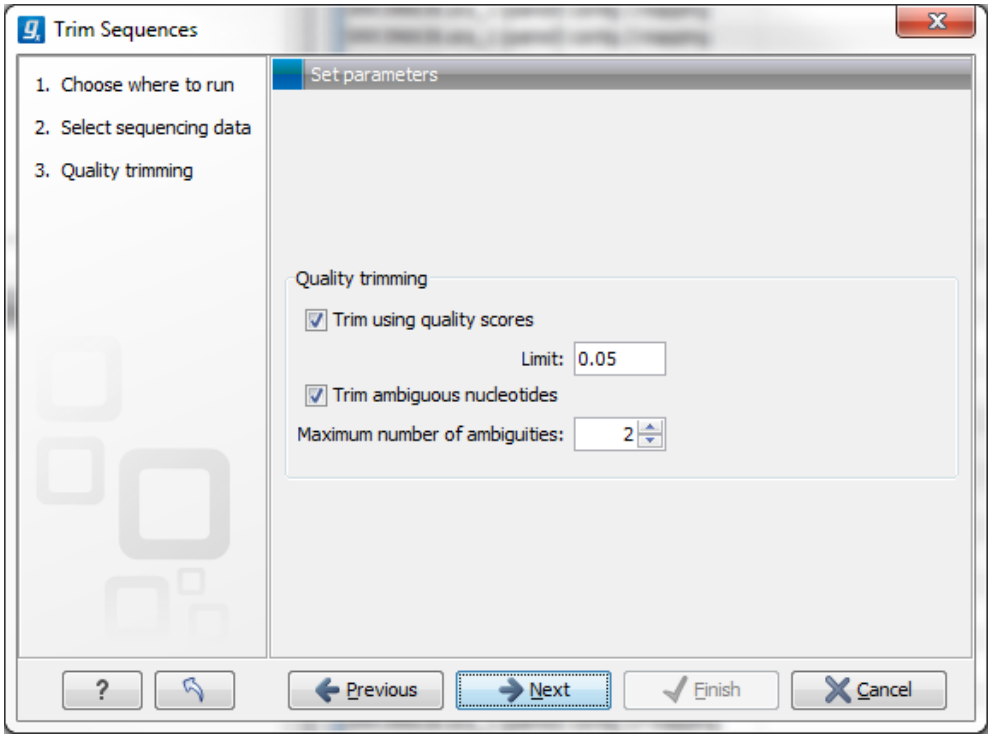



Figure 23.1: *Specifying quality trimming.*

- **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://www.clcbio.com/files/usermanuals/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 I).

### 23.1.2 Adapter trimming

Clicking **Next** will allow you to specify adapter trimming.

In order to trim for adapters, you have to create an adapter list first that must be supplied to the trim tool. A new adapter trim list can be created here:

**File** | **New** | **Trim Adapter List**

This will create a new empty trim adapter list. 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 (figure 23.2). Provide the name and sequence of the adapter that should be trimmed away and adjust the parameters if relevant. Click on the button labeled **Finish** to create the adapter trim list. You must now save the generated adapter trim list in the **Navigation Area**. You can do this by clicking on the tab and dragging and dropping the adapter trim list to the desired destination, or you can go to **File** in the menu bar and the choose **Save as**.

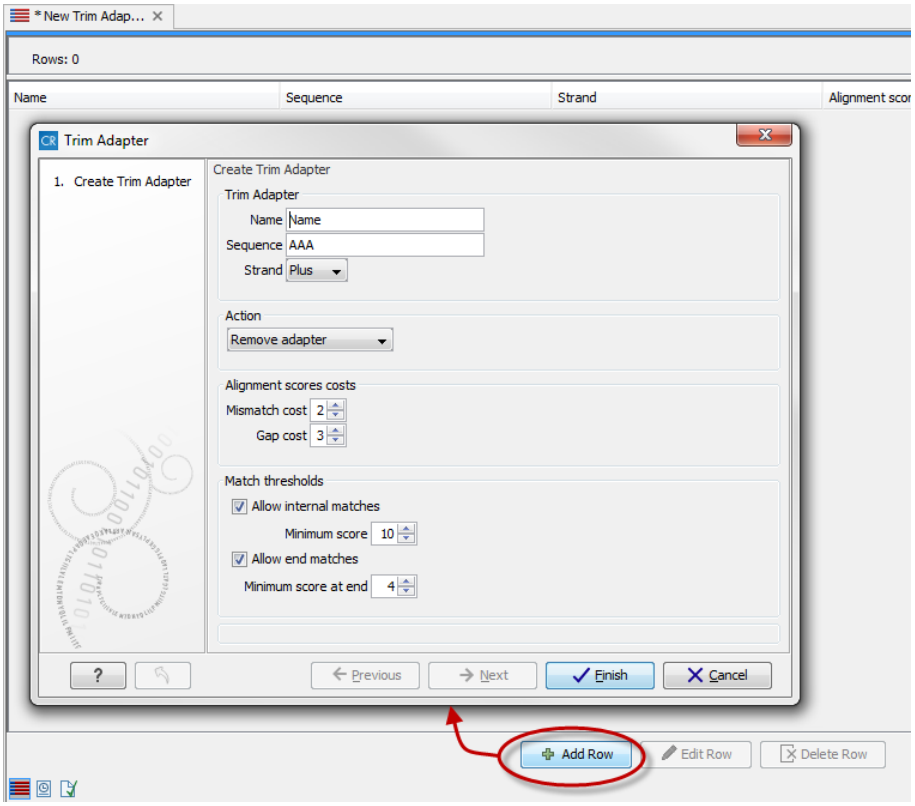



Figure 23.2: *Create a new Adapter Trim List by clicking on the button labeled "Add Row" found at the bottom of the New Trim Adapter view.*

Note: To create an Adapter List file with the adapter sequences that have traditionally been provided with the Genomics Workbench, please go to the Preferences panel. In the Data section of this panel you will find the adapter sequences. Please select the rows of your desired adapter sequences, then click the Convert Trim Adapters button. This will create an Adapter List for use in the Adapter trimming tool.

You can also create an adapter list by importing a comma separated value (.csv) file of your Adapters. This import can be performed with the standard import using either the Automatic Import option or Force Import as Type: Trim Adapter List. To import a csv file, the names of all adapters must be unique - the Workbench is unable to accept files with multiple rows containing the same adapter name. Additionally, the text between each comma that designates a new column should be quoted. The expected import format for Adapter Lists appears as shown in figure 23.3:

```
"Name","Sequence","Strand","Alignment score","Action"
"Adapter 1","AAATTTGC","Plus","Mismatch: 2, Gapcost: 3, Cutoff: 10, Cutoff at end: 4","Remove adapter"
"Adapter 2","AAACGCCT","Plus","Mismatch: 2, Gapcost: 3, Cutoff: 10, Cutoff at end: 4","Remove adapter"
```

Figure 23.3: *The expected import format for Adapter Lists.*

You can also create an Excel file (.xlsx or .xls) format. In this case, you include the same information per column as indicated above, but do not include the quotes within Excel.

At the bottom of the view, you have the following options:

- **Add Rows.** Add a new adapter. This will bring up a dialog as shown in figure 23.4.
- **Delete Row**. Delete the selected adapter.
- **Edit Row**. Edit the selected adapter. This can also be achieved by double-clicking the row in the table.

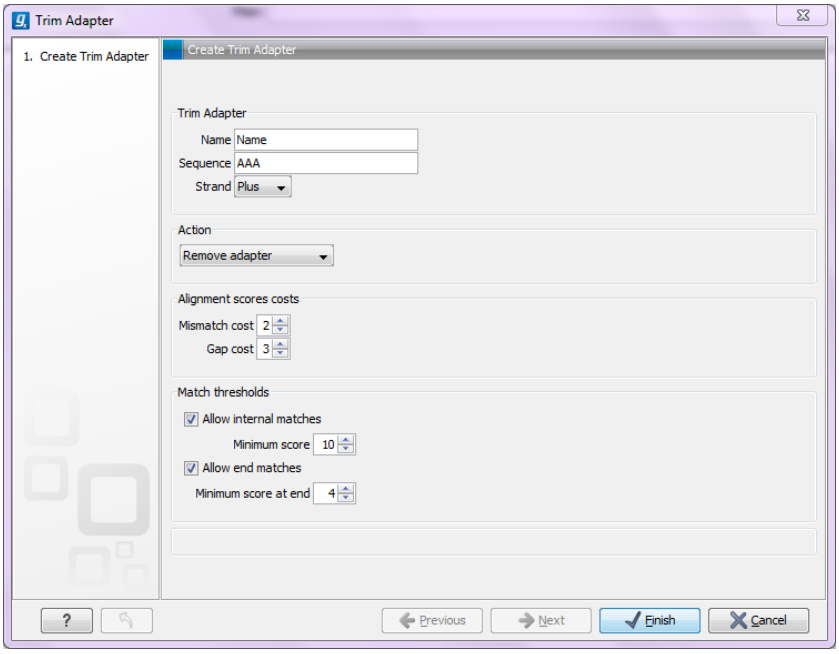



Figure 23.4: *Adding a new adapter for adapter trimming.*

The information to be added for each adapter is explained in the following sections, going into detail with the adapter trim. Once the adapters have been added to the list, it should be saved ( ), and you can select it as shown in figure 23.10.

**Action to perform when a match is found** For each read sequence in the input to trim, the Workbench performs a Smith-Waterman alignment [Smith and Waterman, 1981] with the adapter

sequence to see if there is a match (details described below). When a match is found, the user can specify three kinds of actions:

- **Remove adapter.** This will remove the adapter *and all the nucleotides 5' of the match*. All the nucleotides 3' of the adapter match will be preserved in the read that will be retained in the trimmed reads list. If there are no nucleotides 3' of the adapter match, the read is added to the **List of discarded sequences** (see section 23.1.4).
- **Discard when not found**. If a match is found, the adapter sequence is removed (including all nucleotides 5' of the match as described above) and the rest of the sequence is retained in the list of trimmed reads. If no match is found, the whole sequence is discarded and put in the list of discarded sequences. This kind of adapter trimming is useful for small RNA sequencing where the remnants of the adapter is an indication that this is indeed a small RNA.
- **Discard when found**. If a match is found, the read is discarded. If no match is found, the read is retained in the list of trimmed reads. This can be used for quality checking the data for linker contaminations etc.

**When is there a match?** To determine whether there is a match there is a set of scoring thresholds that can be adjusted for each adapter as shown in figure 23.4.

First, you can choose the costs for mismatch and gaps. A match is rewarded one point (this cannot be changed), and per default a mismatch costs 2 and a gap (insertion or deletion) costs 3. A few examples of adapter matches and corresponding scores are shown in figure 23.5.

```
    CGTATCAATCGATTACGCTATGAATG
a)      ||||||| ||||            11 matches - 2 mismatches = 7
       TTCAATCGGTTAC

    CGTATCAATCGATTACGCTATGAATG
       |||||||||| ||||          14 matches - 1 gap = 11
b)     ATCAATCGAT-CGCT

    CGTATCAATCGATTACGCTATGAATG
c)      |||||||                 7 matches - 3 mismatches = 1
       TTCAATCGGG
```

Figure 23.5: *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.*

In the panel below, you can set the **Minimum score** for a match to be accepted. Note that there is a difference between an **internal match** and an **end match**. The examples above are all internal matches where the alignment of the adapter falls within the read. Figure 23.5 shows a few examples with an adapter match at the end:

In the first two examples, the adapter sequence extends beyond the end of the read. This is what typically happens when sequencing e.g. small RNAs where you sequence part of the adapter. The third example shows an example which 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 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). Also note, that if **Remove**

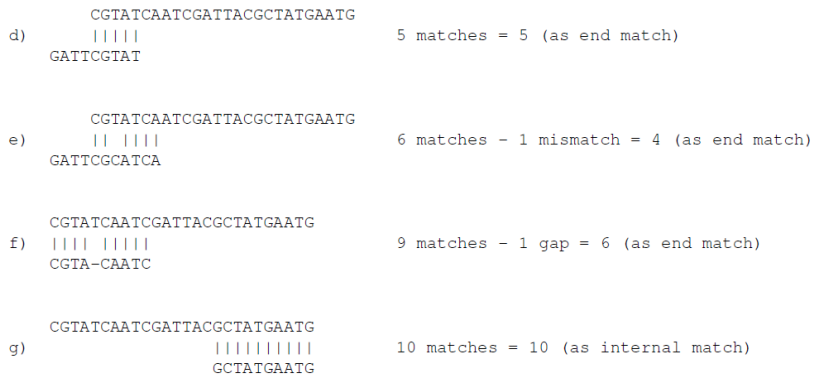



Figure 23.6: *Four examples showing a sequencing read (top) and an adapter (bottom). The examples are artificial.*

**adapter** is chosen for the last example, the full read will be discarded because everything 5' of the adapter is removed.

Below, the same examples are re-iterated showing the results when applying different scoring schemes. In the first round, the settings are:

- Allowing internal matches with a minimum score of 6
- Not allowing end matches
- Action: Remove adapter

The result (shown in figure 23.7) would be the following (the retained parts are green):

A different set of adapter settings could be:

- Allowing internal matches with a minimum score of 11
- Allowing end match with a minimum score of 4
- Action: Remove adapter

The result would be (shown in figure 23.8):

**Strand settings**

Each adapter is defined as either **Plus** or **Minus**. Note that all the definitions above regarding 3' end and 5' end also apply to the minus strand (i.e. selecting the Minus strand is equivalent to reverse complementing all the reads). The adapter in this case should be defined as you would see it on the plus strand of the reverse complemented read. The example below (figure 23.9) shows a few examples of an adapter defined on the minus strand. It shows hits for an adapter sequence defined as `CTGCTGTACGGCCAAGGCG`, searching on the minus strand.

You can see that if you reverse complemented the adapter you would find the hit on the plus strand, but then you would have trimmed the wrong end of the read. So it is important to define the adapter as it is, without reverse complementing.

**Trimming of 3' ends of the reads**

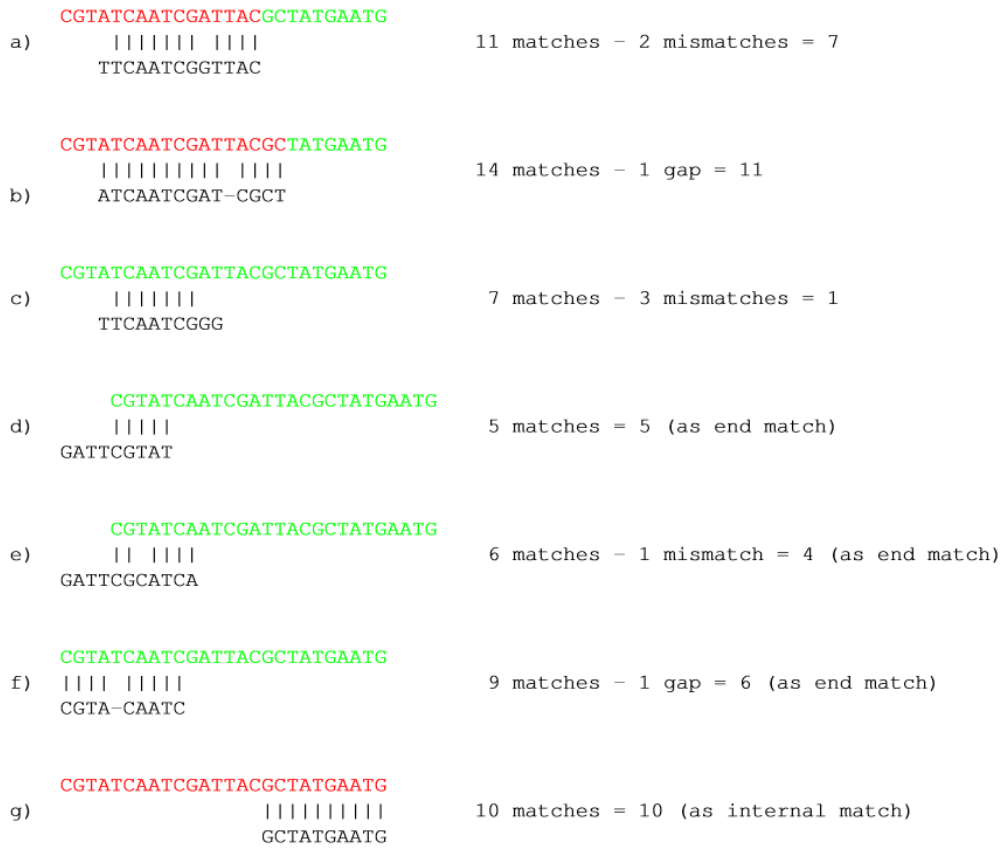



Figure 23.7: *The results of trimming with internal matches only. Red is the part that is removed and green is the retained part. Note that the read at the bottom is completely discarded.*

To trim an adapter and everything to the 3' end of the adapter you will need to search for the reverse complement of the adapter on the negative strand. This is achieved by creating a new Trim Adapter List from the reverse complement of your adapter sequence, choosing the minus strand of your reads and run adapter trimming with the new Trim Adapter List as input.

**Other adapter trimming options**

When you run the trim, you specify the adapter settings as shown in figure 23.10.

Select an trim adapter list (see section 23.1.2 on how to create an adapter list) that defines the adapters to use.

You can specify if the adapter trimming should be performed in **Color space**. Note that this option is only available for sequencing data imported using the SOLiD import (see section 6.3.3). When doing the trimming in color space, the Smith-Waterman alignment is simply done using colors rather than bases. The adapter sequence is still input in base space, and the Workbench then infers the color codes. Note that the scoring thresholds apply to the color space alignment (this means that a perfect match of 10 bases would get a score of 9 because 10 bases are represented by 9 color residues). Learn more about color space in section 25.4.

Checking the **Search on both strands** checkbox will search both the minus and plus strand for the adapter sequence. **Note!** If a match is found on the reverse strand the Trim action will reverse complement the read before trimming and output the trimmed reverse complement. Its

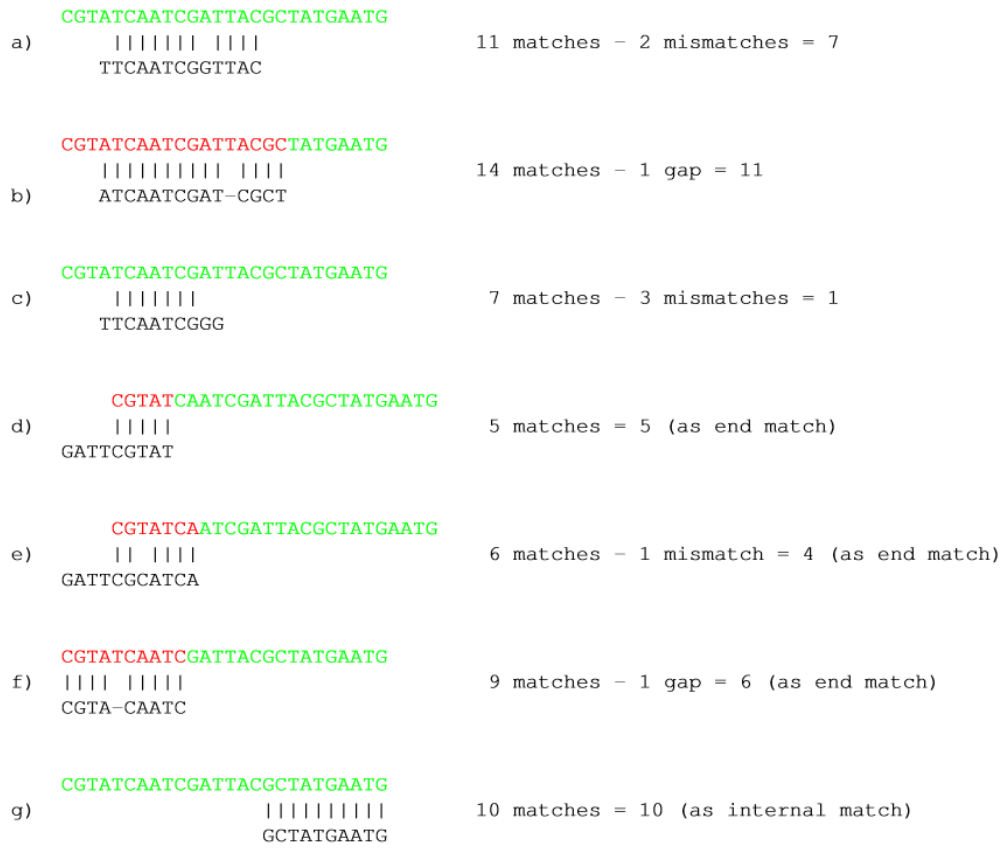



Figure 23.8: *The results of trimming with both internal and end matches. Red is the part that is removed and green is the retained part.*




Figure 23.9: *An adapter defined as CTGCTGTACGGCCAAGGCG searching on the minus strand. Red is the part that is removed and green is the retained part. The retained part is 3' of the match on the minus strand, just like matches on the plus strand.*

intended use is for removal of multiplexing barcodes and primers.

Below you find a preview listing the results of trimming with the current settings 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 strand and alignment score settings.
- **Reads discarded**. This is the number of reads that will be completely discarded. This can

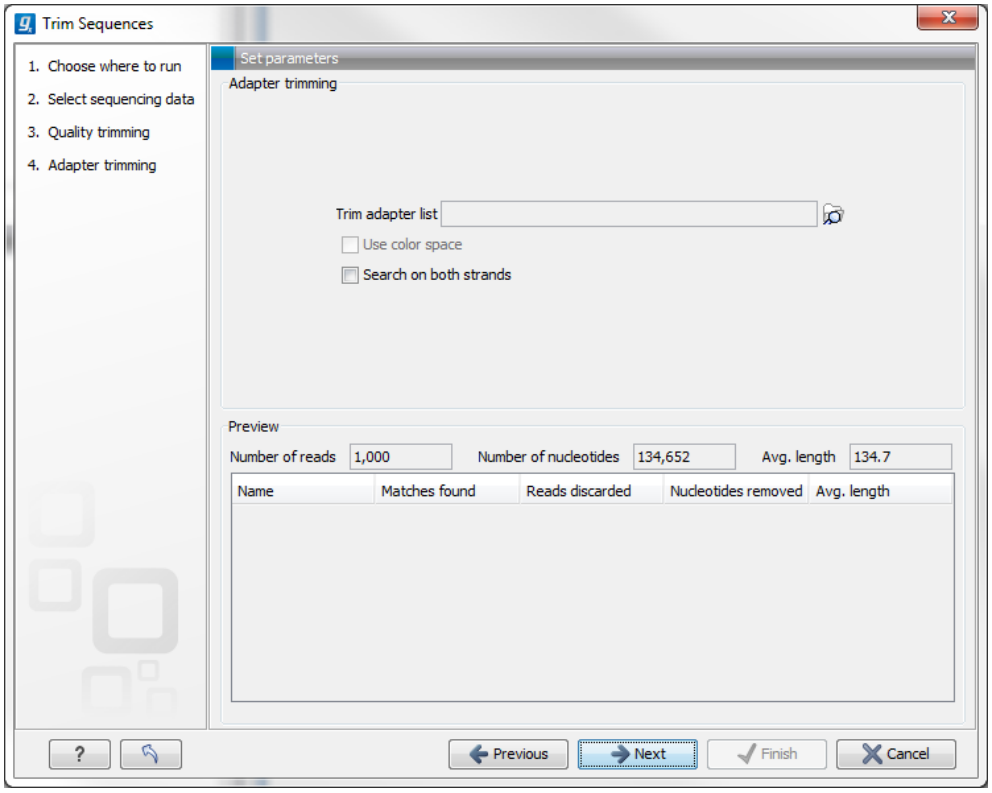



Figure 23.10: *Trimming your sequencing data for adapter sequences.*

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 adapter trim affects the results. If other kinds of trimming (quality or length trimming) is applied, this will not be reflected in the preview but still influence the results.

### 23.1.3 Length trimming

Clicking **Next** will allow you to specify length trimming as shown in figure 23.11.

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 choose to **Discard reads below length**. This can be used if you wish to simply discard reads because they are too short. Similarly, you can discard reads above a certain length. This will typically be useful when investigating e.g. small RNAs (note that this is an integral part of the small RNA analysis together with adapter trimming).

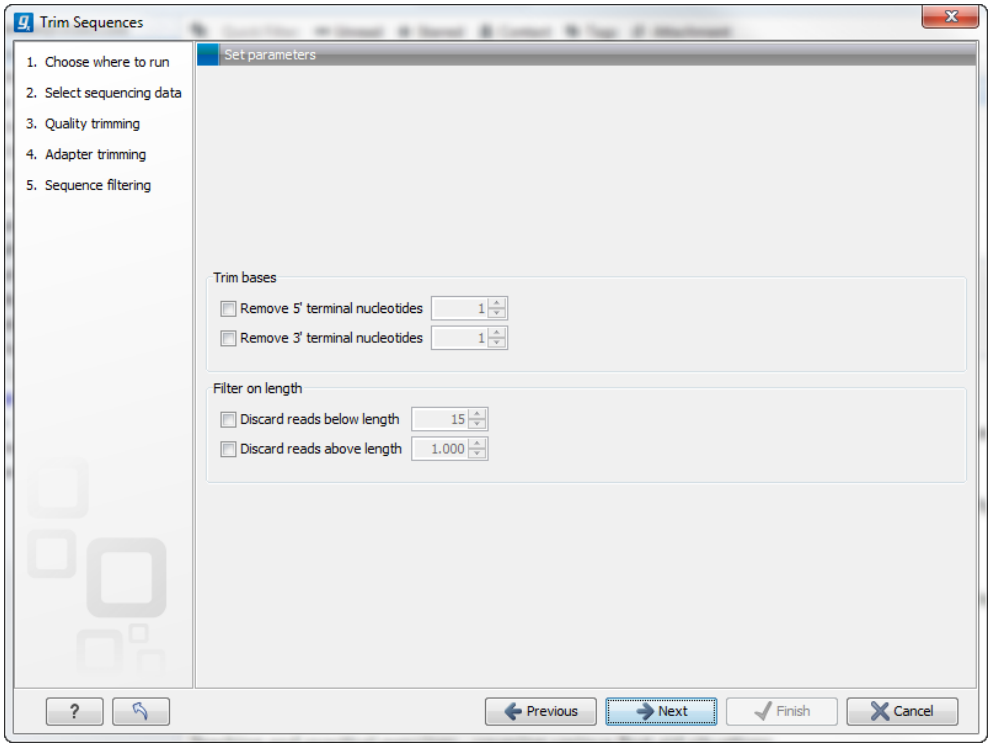



Figure 23.11: *Trimming on length.*

### 23.1.4 Trim output

Clicking **Next** will allow you to specify the output of the trimming as shown in figure 23.12.

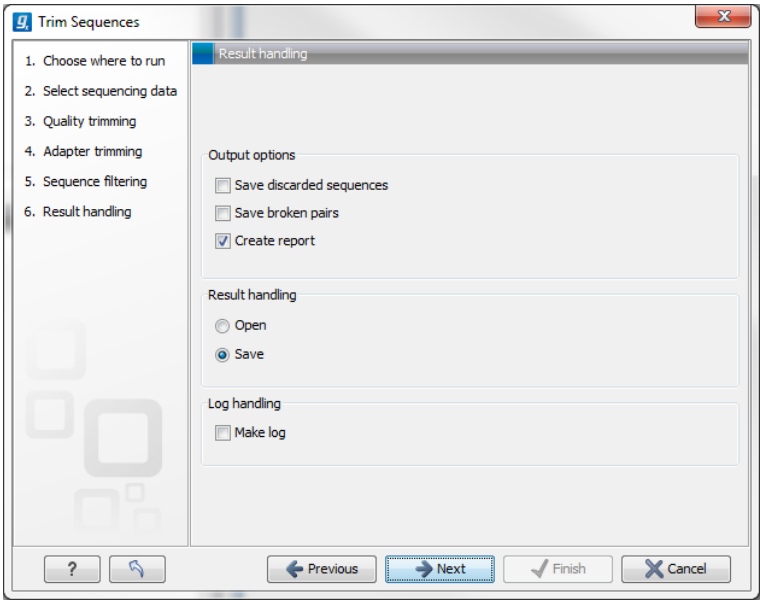



Figure 23.12: *Specifying the trim output. No matter what is chosen here, the list of trimmed reads will always be produced.*

No matter what is chosen here, the list of trimmed reads will always be produced. In addition the following can be output as well:

- **Create list of 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.
- **Create report**. An example of a trim report is shown in figure 23.13. 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.
    - **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).

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**. This will start 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.

## 23.2 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].

### 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

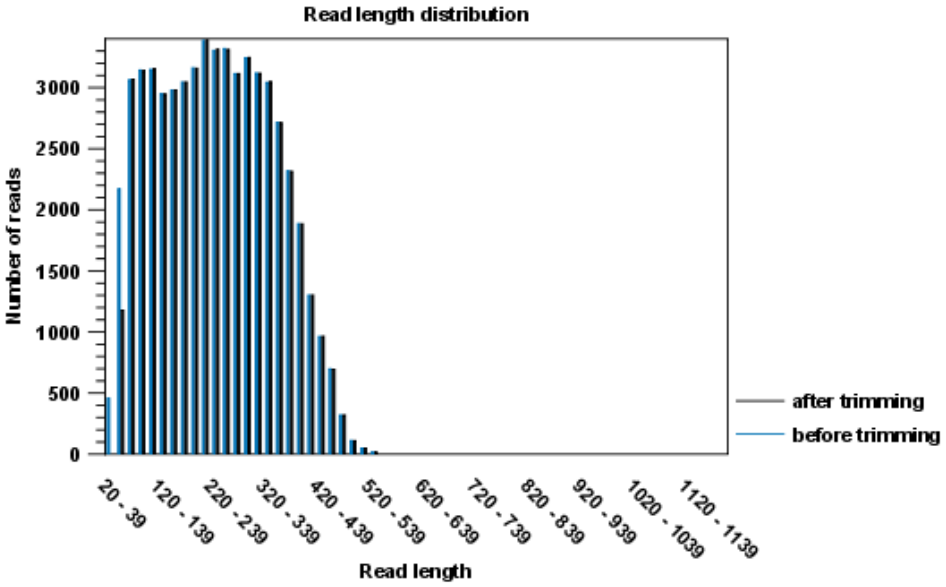



Figure 23.13: *A report with statistics on the trim results.*

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 23.14.




Figure 23.14: *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. Based on their barcodes this can be done using the demultiplexing functionality of the *CLC Genomics Workbench*. Using this tool, sequences are associated with a particular sample when they contain an exact match to a particular barcode. Sequences that do not contain an exact match to any of the barcode sequences provided are classified as not grouped and are put into a sequence list with the name "Not grouped".

Note that there is also an example using Illumina data at the end of this section.

Before processing the data, you need to import it as described in section 6.3.

Please note that demultiplexing is often carried out on the sequencing machine so that the sequencing reads are already separated according to sample. This is often the best option, if it is available to you. Of course, in such cases, the data will not need to be demuliplexed again after import into the *CLC Genomics Workbench*.

To demultiplex your data, please go to:

**Toolbox** | **NGS Core Tools** ( ) | **Demultiplex Reads** ( )

This opens a dialog where you can specify the sequences to process (figure 23.15).

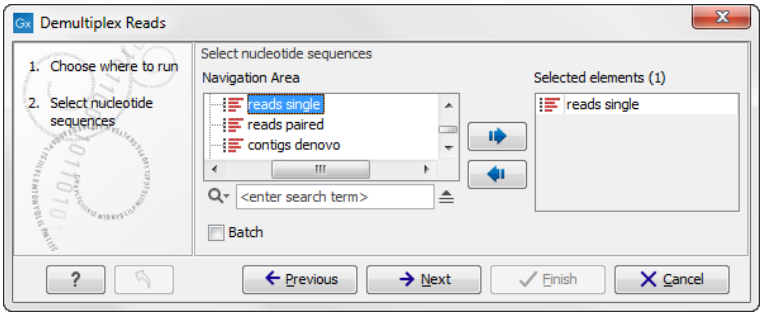



Figure 23.15: *Specify the sequences to demultiplex.*

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 23.16.

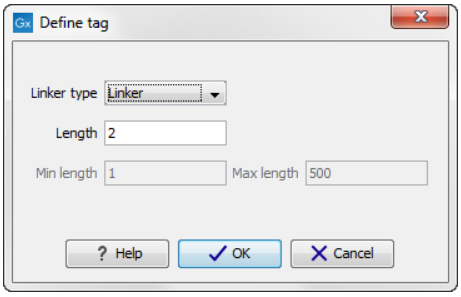



Figure 23.16: *Defining an element of the barcode system.*

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 23.14, 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 23.14, the dialog should include a linker, a barcode, and a sequence as shown in figure 23.17.

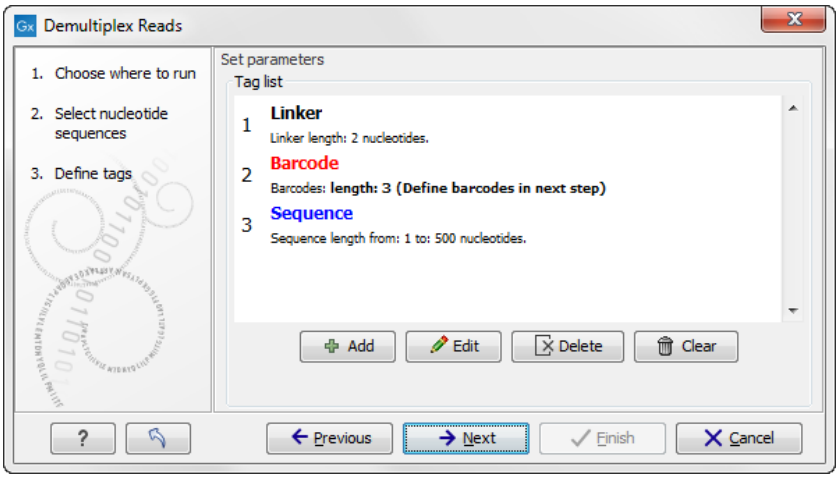



Figure 23.17: *Processing the tags as shown in the example of figure 23.14.*

Click on the button labeled **Next** to go to 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 "Barcode:" and is followed by the specified barcode sequence), 2) The discarded reads that did not have a barcode (this file is called "Not grouped"), and 3) a "Demultiplex Reads report", which shows the fraction of reads with and without a barcode (see figure 23.18).

If you have paired data the procedure is exactly the same, except for one thing: the dialog shown in figure 23.17 will be displayed twice - one for each part of the pair.

Figure 23.19 shows an example that could illustrate paired end reads. In this example we have paired end reads from two different samples mixed together.

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

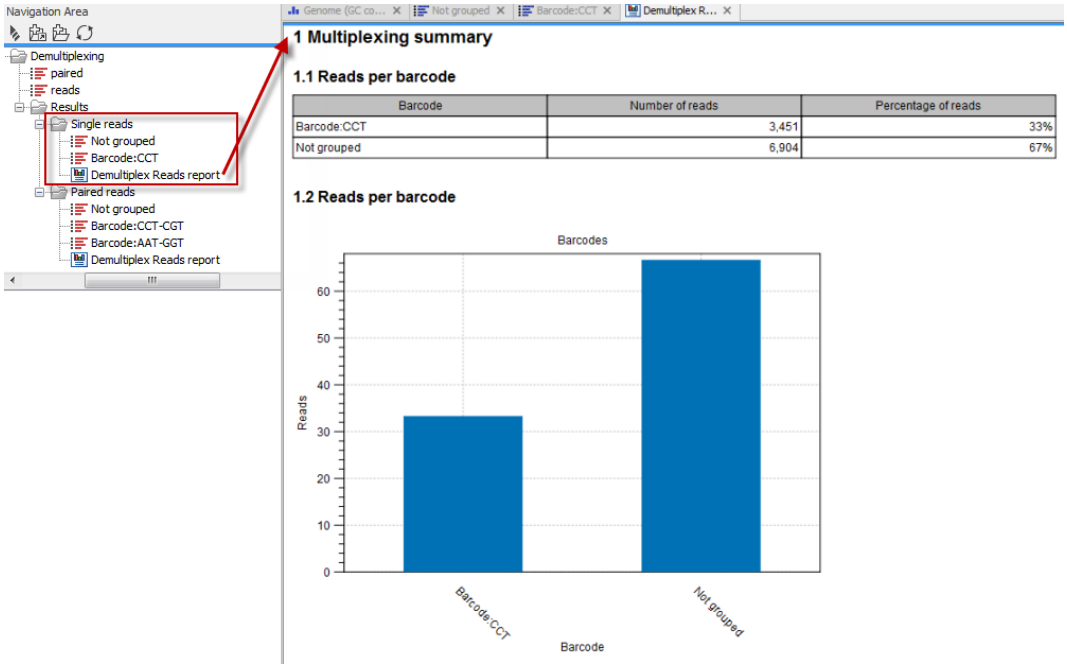



Figure 23.18: *Three different outputs are generated when analyzing single reads with only one sample using the default output settings. If several samples had been mixed together there would be a sequence list for each sample (each specified barcode). The Demultiplex Reads report is shown in the right-hand side of the figure.*

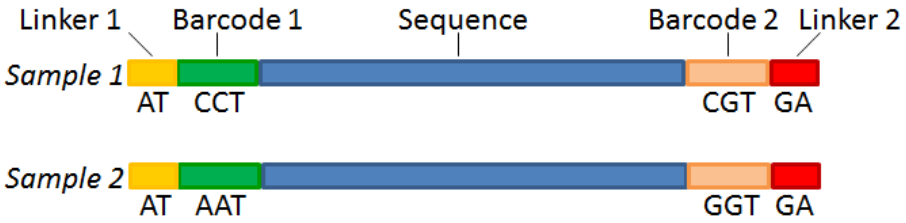



Figure 23.19: *Paired end reads with linkers and barcodes. Two different samples are mixed together in this example.*

reads. Select the paired end read sequences that should be demultiplexed (figure 23.20).

Click on the button labeled **Next** to define the tags and sequence for the forward reads (figure 23.21).

Barcodes can be entered manually or imported from a properly formatted CSV or Excel file:

**Manually** The barcodes can be entered manually by clicking the **Add** ( ) button. In the example shown in figure 23.19 the barcodes should be defined as shown in figure 23.22.

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 from CSV or Excel** To import a file of barcodes, click on the **Import** ( ) button. 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 contain columns of information

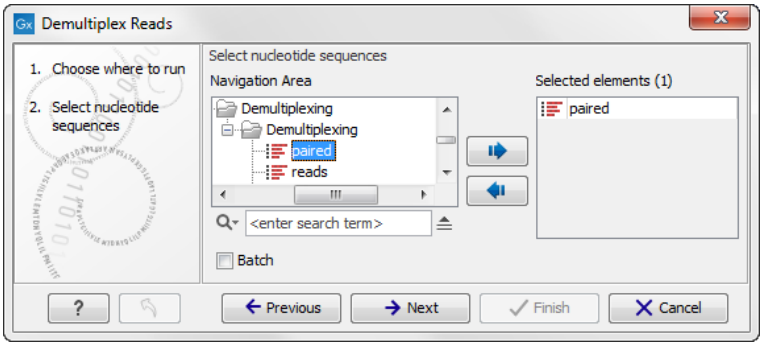



Figure 23.20: *Specifying the paired end reads to demultiplex.*

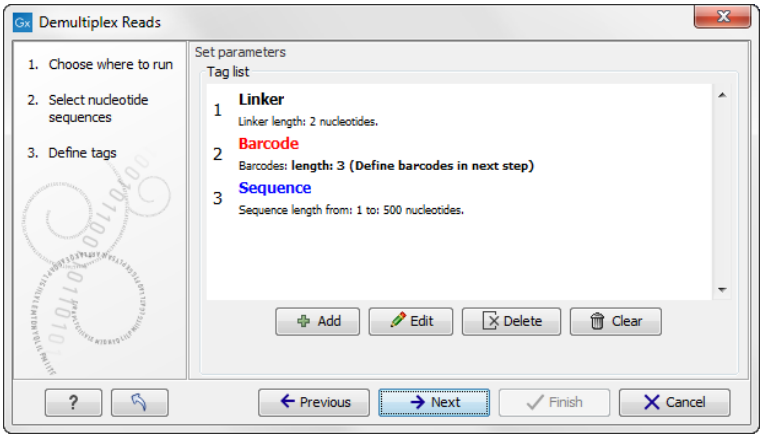



Figure 23.21: *Specifying the setup of the forward read (tags and sequence).*

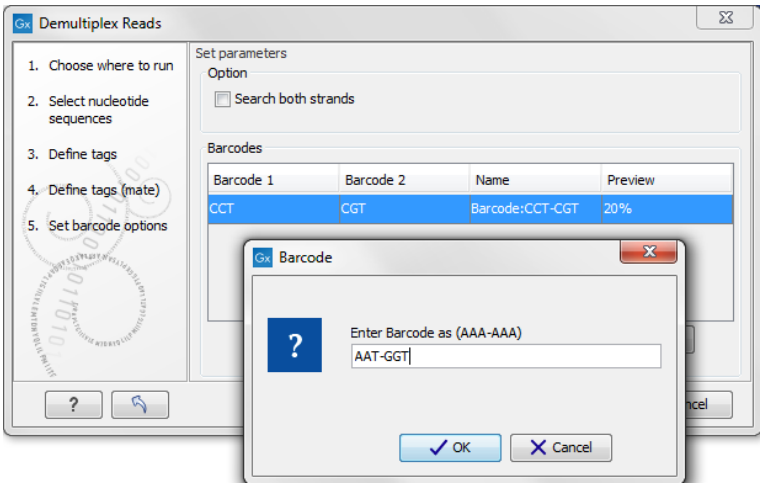



Figure 23.22: *The barcodes for the set of paired end reads for sample 1 have already been defined and the barcodes for sample 2 is being entered in the format AAA-AAA, which corresponds to Barcode1-Barcode2 for sample 2 in the example shown in figure 23.19.*

that looks like:

```
 "AAAAAA","Sample1"
"GGGGGG","Sample2"
 "CCCCCC","Sample3"
```

The **Preview** column will show a preview of the results by running through the first 10,000 reads (figure 23.23).

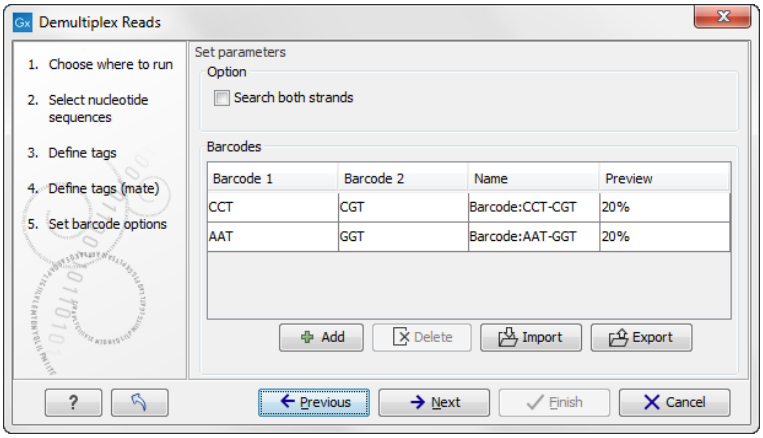



Figure 23.23: *A preview of the results.*

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).

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 23.24.

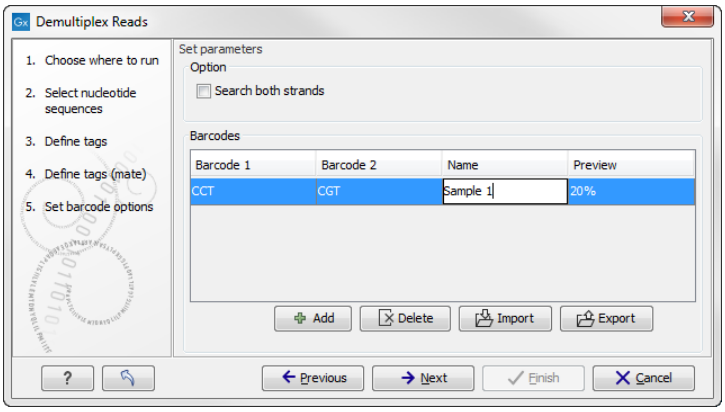



Figure 23.24: *The name of the sequence can be renamed by double-clicking on the existing name.*

Click on the button labeled **Next** to define the tags and sequence for the reverse reads (figure 23.25).

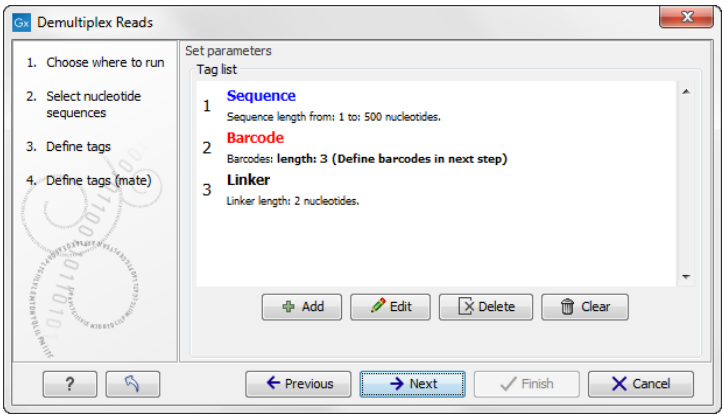



Figure 23.25: *Specifying the setup of the reverse read (tags and sequence).*

Click **Next** to specify the output options. First, you can choose to create a list of the reads that

could not be grouped. Second, you can create a summary report showing how many reads were found for each barcode. Click **Finish** to perform the demultiplexing.

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 8.1).

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 23.26.

## 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

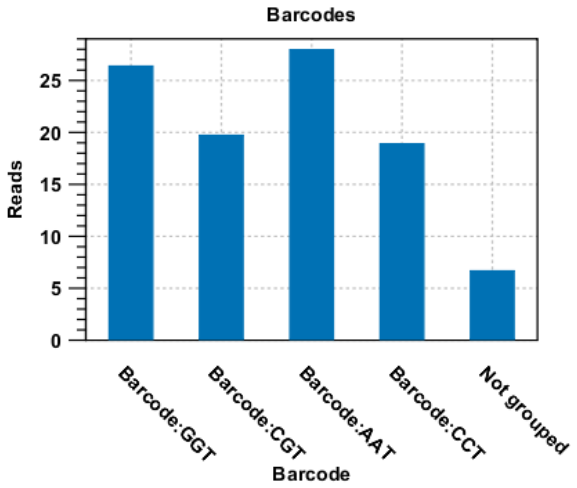



Figure 23.26: *An example of a report showing the number of reads in each group. In this example four different barcodes were used to separate four different samples.*

### An example using Illumina barcoded sequences

The data set in this example can be found at the Short Read Archive at NCBI: http://www.ncbi.nlm.nih.gov/sra/SRX014012. It can be downloaded directly in fastq format via the URL http://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?cmd=dload&run_list=SRR030730&format=fastq. The file you download can be imported directly into the Workbench.

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] at http://nar.oxfordjournals.org/cgi/data/gkn502/DC1/1).

The settings in the dialog should thus be as shown in figure 23.27.

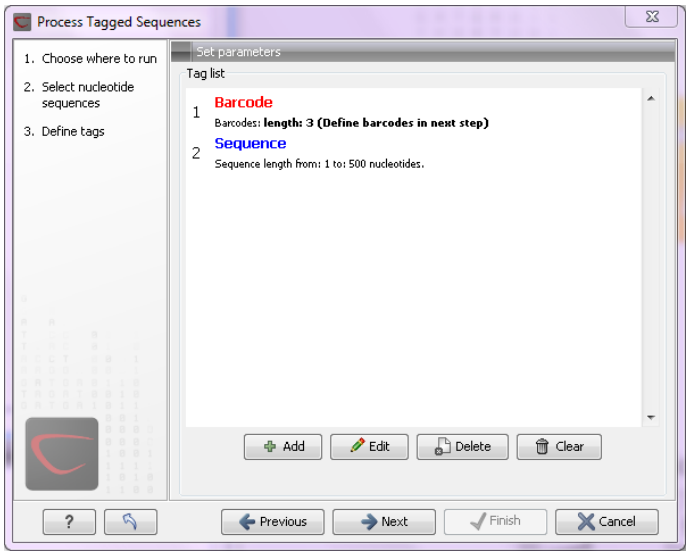



Figure 23.27: *Setting the barcode length at three*

Click **Next** to specify the bar codes as shown in figure 23.28 (use the **Add** button).

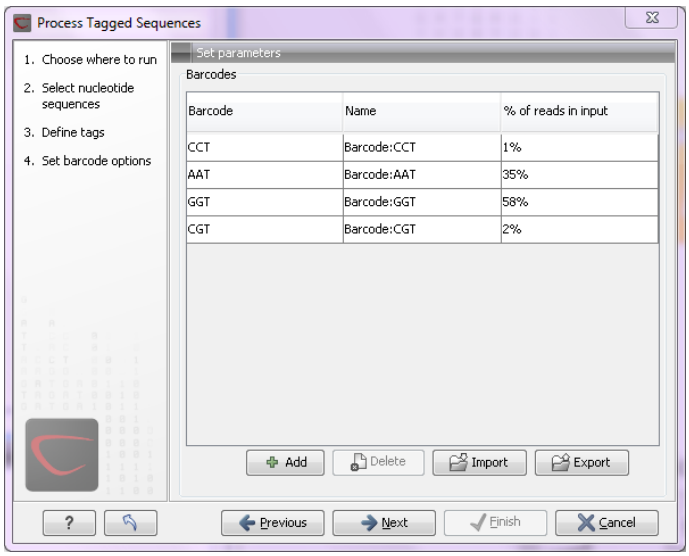



Figure 23.28: *A preview of the result*

With this data set we got the four groups as expected (shown in figure 23.29). The **Not grouped** list contains 445,560 reads that will have to be discarded since they do not have any of the barcodes.

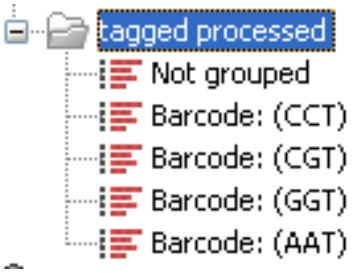



Figure 23.29: *The result is one sequence list per barcode and a list with the remainders*

## 23.3 Sequencing data quality control

Quality assurance as well as concern 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, however, introduce additional error-sources. Resulting error-signatures can neither be easily removed from resulting sequencing data nor even recognized, which is mainly due to the massive amount of data. Altogether 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-up, ultimately threatening the accuracy of biological conclusions.

Unfortunately, most of the problems and evolving questions raised above can't be solved and answered entirely. However, the sequencing data quality control tool of the *CLC Genomics Workbench* provides various generic tools to assist in the quality control process of the samples by assessing and visualizing statistics on:

- Sequence-read lengths and base-coverages
- Nucleotide-contributions and base-ambiguities
- Quality scores as emitted by the base-caller
- Over-represented sequences and hints suggesting contamination events

This tool aims at assessing above quality-indicators and investigates proper and improper result presentation. The inspiration comes from the FastQC-project (http://www.bioinformatics.babraham.ac.uk/projects/fastqc/).

### 23.3.1 Report contents

The sections below describe the contents of the report. Note that the two terms "per-sequence" and "per-base" are used frequently in the following descriptions. The generated report is divided into per-sequence and per-base sections. In per-sequence assessments 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.

The report comes in two different flavors: a supplementary report consisting of tables representing all the values that are calculated, and a main summary reports where the tables are visualized in plots (see an example in figure 23.30). Both reports can be exported as pdf files or Excel spread sheets.

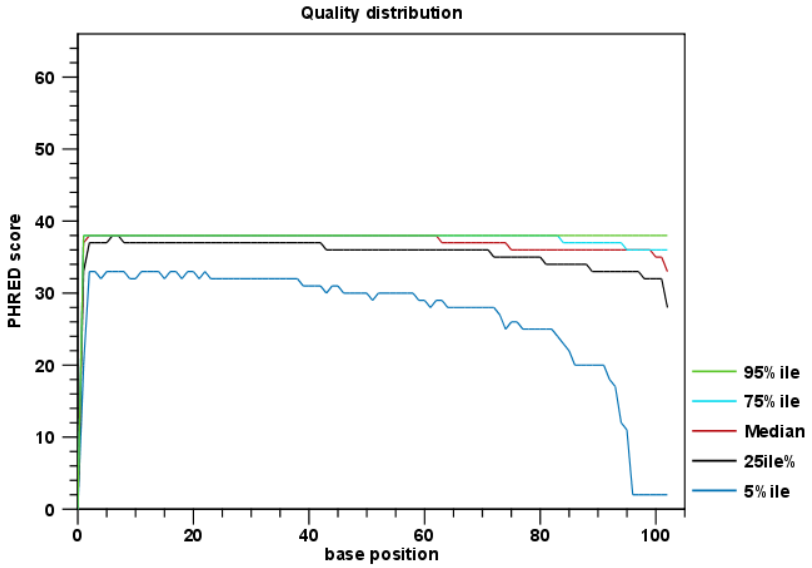



Figure 23.30: *An example of a plot from the graphical report, showing the quality values per base position.*

### Basic analysis

The basic analysis section assesses the most simple characteristics that are supported by all sequencing technologies.

The Summary table provides information regarding the creation date, the author, the software used, the number of data sets the report is based upon, as well as data name and content in terms of read number and total number of nucleotides.

**Sequence length 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.

**Base coverage distribution** Calculates absolute coverages for individual base positions. The resulting table correlates base-positions with the number of sequences that supported (covered) that position.

**Sequence-wise %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-nulceotides by the length of that sequence.

**Sequence-wise %N-content distribution** 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.

**Base-wise nucleotide distributions** Calculates absolute coverages for the four DNA nucleotides (A, C, G or T) for each base position in the sequences.

**Base-wise GC-distribution** Calculates absolute coverages of C's + G's for each base position in the sequences.

**Base-wise N-distribution** 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 analysis

The quality analysis examines 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. The following quality distributions are reported:

**per-sequence quality distribution** Calculates amounts 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.

**per-base quality distribution** Calculates amounts 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.

#### Over-representation analysis

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.

**Individual 5-mer distribution** 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). This analysis will reveal if there is a pattern of bias at different positions over the read length. Such a bias might origin from non-trimmed adapter sequences, poly-A tails or other sources.

#### Duplicated sequences analysis

The duplicated sequences analysis identifies sequences that have been sequenced multiple times. In order to achieve reasonable performance, not all input sequences are analyzed. Instead a sequence-dictionary is used, whose entries are sampled evenly from input sequences. Please note that if you select multiple sequence lists as an input, they will all be considered one data set for this analysis (batching can be used to generate separate reports for an individual sequence list). As soon as a sequence makes it into the dictionary (which is a random process), it is

tracked for duplicates until all sequences have been examined. The dictionary size is 250 000 sequences.

Because all current sequencing techniques tend to report fading quality scores for the 3' ends of sequences, there is a risk that duplicates are NOT detected, just because of sequencing errors near their 3' ends. Therefore, the identity of two sequences is calculated using only the first 50nt from the 5' end.

**Sequence duplication levels** This results in a table correlating duplication counts with the number of sequences that featured that duplicate-count. For example, if the dictionary contains 10 sequences and each sequence was seen exactly once, then the table will contain only one row displaying: 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, such that a significantly elevated bar-height at x=100 is a normal observation. 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.

### 23.3.2 Adapters

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.

### 23.3.3 Running the quality control tool

The tool is found in the Toolbox:

**Toolbox** | **NGS Core Tools** (  ) | **Create Sequencing QC Report** (  )

Select one or more sequence lists with sequencing reads as input. If sequence lists in the Navigation Area were already selected, these will be shown in the Selected Elements window. When multiple lists are selected as an input, they are all analyzed in one pool. If you need separate reports for each data set, you can run it in a batch. Clicking **Next** allows you to set parameters as displayed in figure 23.31.

The following parameters can be set:

- **Quality analysis** as described in section 23.3.1.
- **Over-representation analysis** as described in section 23.3.1.

Click **Next** to adjust output options which allow you to select the graphical and supplementary report.

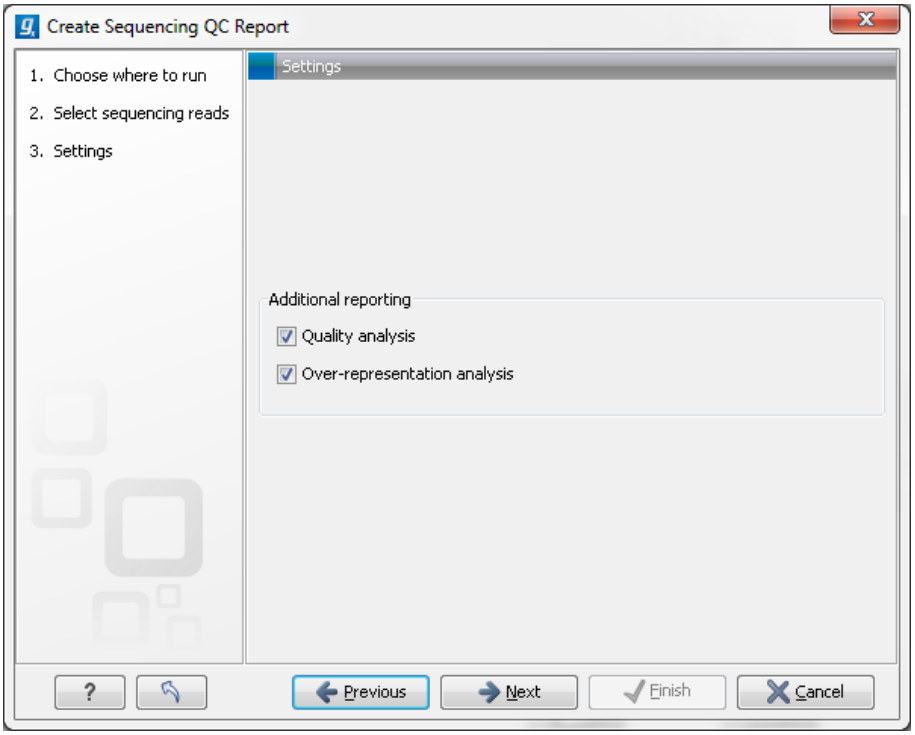



Figure 23.31: *Setting parameters for quality control.*

## 23.4 Merge overlapping pairs

Some paired end library preparation methods using relatively short fragment size will generate data with overlapping pairs. This type of data can be handled as standard paired-end data in the *CLC Genomics Workbench*, and it will work perfectly fine (see details for variant detection in section 27.12).

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** | **NGS Core 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 23.32.

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

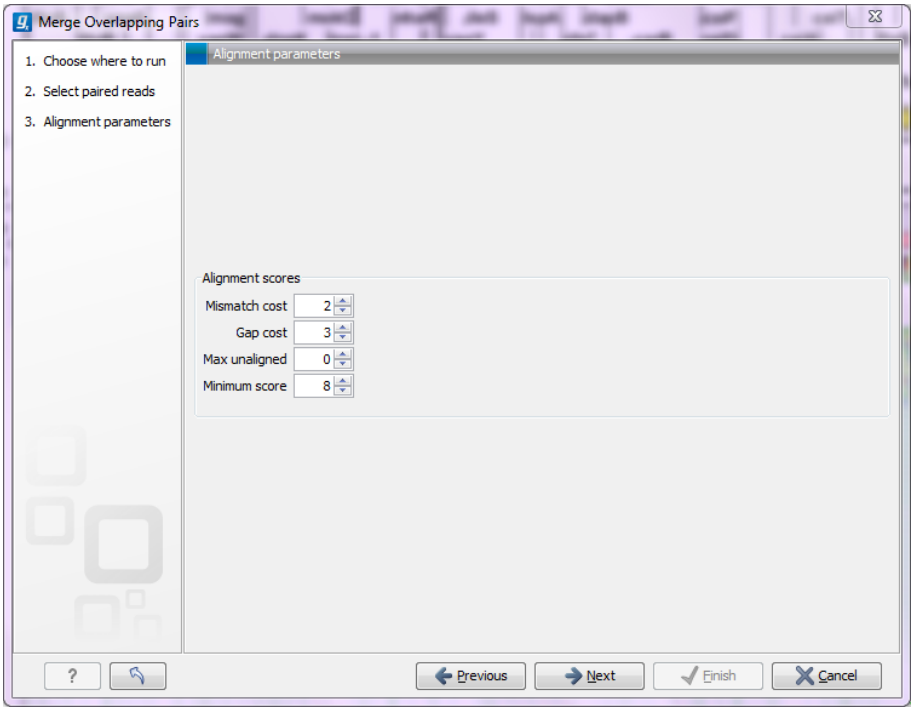



Figure 23.32: *Setting parameters for merging overlapping pairs.*

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.

### 23.4.1 Using quality scores when merging

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 I) 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:

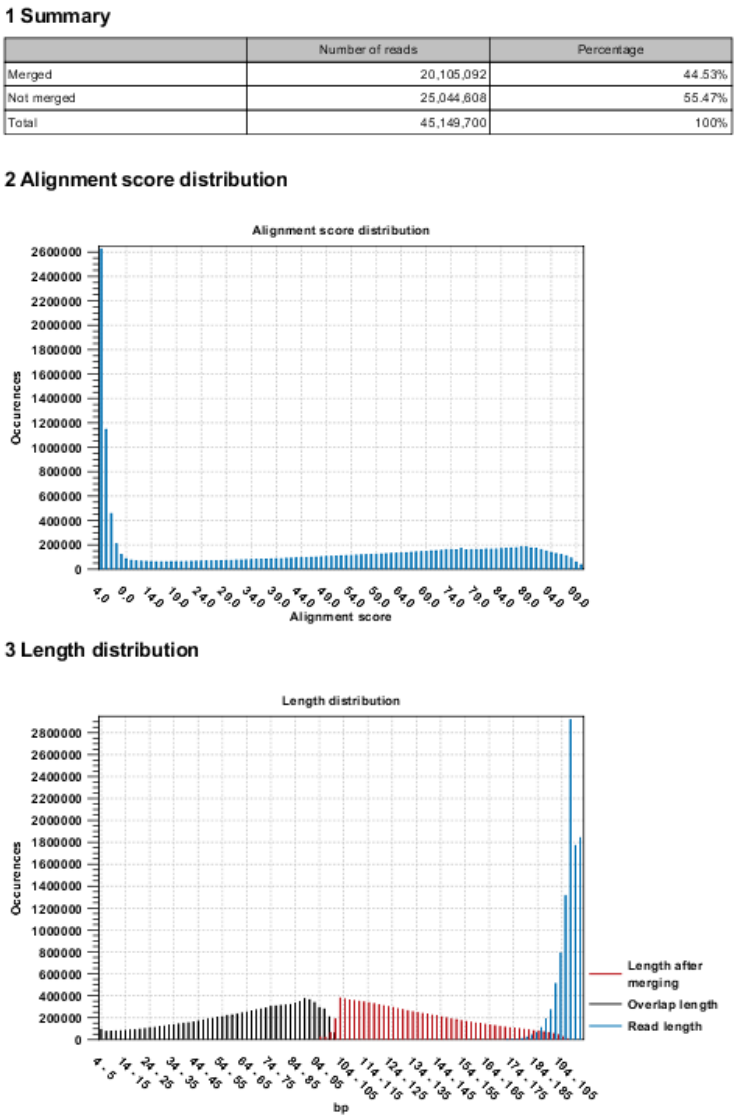
- When the two reads agree at a position, the two quality scores are summed to form the quality score of the base in the new read. The score is capped at the maximum value on the quality score scale which is 64. Phred scores are log scores, so their sums represent the multiplication of the original error 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.

### 23.4.2 Report of merged pairs

Figure 23.33 shows an example of the report generated when merging overlapping pairs.



**1 Summary**

| | Number of reads | Percentage |
|---|---|---|
| Merged | 20,105,092 | 44.53% |
| Not merged | 25,044,608 | 55.47% |
| Total | 45,149,700 | 100% |

**2 Alignment score distribution**



**3 Length distribution**



Figure 23.33: *Report of 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 23.4.
- A plot of read lengths. This shows the distribution of read lengths for the pairs that have been merged:
  - The length of the overlap.
  - The length of the merged read.

- The combined length of the two reads in the pair before merging.

# Chapter 24

# Tracks

## Contents


**24.1 Track lists . . . 544**
24.1.1 Zooming and navigating track views . . . 545
24.1.2 Adding, removing and reordering tracks . . . 549
24.1.3 Showing a track in a table . . . 550
24.1.4 Open track from a track list in table view . . . 551
24.1.5 Finding annotations on the genome . . . 551
24.1.6 Extract sequences from tracks . . . 553
24.1.7 Creating track lists in workflows . . . 555
**24.2 Retrieving reference data tracks . . . 555**
**24.3 Merging tracks . . . 556**
**24.4 Converting data to tracks and back . . . 557**
24.4.1 Convert to tracks . . . 557
24.4.2 Convert from tracks . . . 558
**24.5 Annotate and filter tracks . . . 559**
24.5.1 Annotate with overlap information . . . 560
24.5.2 Extract reads based on overlap . . . 560
24.5.3 Filter annotations on name . . . 563
24.5.4 Filter Based on Overlap . . . 564
**24.6 Creating graph tracks . . . 564**


This chapter explains how to visualize tracks, how to retrieve reference data and finally how to perform generic comparisons between tracks.

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).[1]

[1]The track concept was first introduced with the Genomics Gateway plugin in 2011 and made an integral part of the CLC Genomics Workbench 5.5 release.

In tracks, all information is tied to genomic positions. A central coordinate-system is provided by a reference genome, which allows that different types of data or results for different samples can be seen and analyzed together.

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.

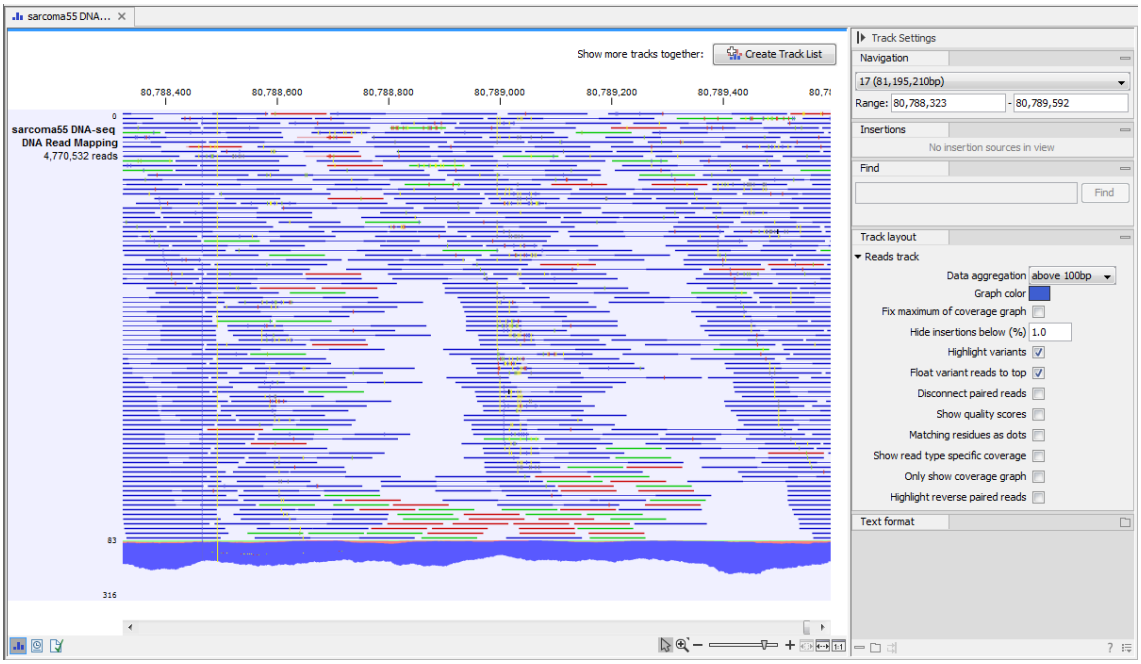



Figure 24.1: *A paired-end mapping read-track opened, displaying reads and coverage. On the top right, the button for creating a Track List is visible. On the right is the SidePanel.*

The different track types in the *CLC Genomics Workbench* are:

**A sequence ( )** This is the track type that is used for holding the reference genome. The sequence track contains the single reference sequences of the genome (e.g. the chromosomes or the consensus sequences of de novo assembled contigs).

**A reads track ( )** This is the track type that is used for holding a read mapping e.g. as produced by the Map Reads to Reference (see section 25.1), Local Realignment (see section 25.6) or RNA-Seq Analysis (see section 28.1) 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.

**A variant track ( )** A variant track is a particular kind of track that is used to store 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 analysis (e.g using the Probabilistic or

Quality-based variant callers or by importing a variant format file (such as a "vcf" or a "gvf" file) or downloading it from a database (e.g. COSMIC or dbSNP).

The tool InDels and Structural Variants (see section 27.10) 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).

**An annotation track ( )** Each annotation track contains a certain type of annotations. Examples are gene or mRNA tracks, which contain gene, respectively mRNA, annotations, UTR tracks, conservation score tracks and target region tracks. They may be obtained either by importing (see section 6.2 or downloading them into the Workbench (e.g from a .bed, .gtf or .gff file or a database, such as ENSEMBL). Also, many of the tools in the *CLC Genomics Workbench* will output annotation tracks. Examples are the Indels and Structural Variants tool, which will put the detected structural variants (that do not fulfill the requirements for being of type "variant") in an annotation track, or the ChIP-seq detection tool which will put the detected "peaks" into a "peak" annotation track. A description of how to annotate and filter tracks is found in section 24.5.

**A coverage graph ( )** The coverage graph track is calculated from a reds track and contains a graphical display of the coverage at each position in the reference.

**An expression track ( )** The RNA-seq algorithm produces expression tracks; one for genes and one for transcripts. These are tracks that have an annotation for each gene, respectively transcript, *and* an expression value associated to that annotation.

An example of the different types of tracks is given in figure 24.2.

For comparison tools specific to resequencing and variants, please see chapter 27.

## 24.1 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 24.3 shows an example of a track list including a track with mapped reads at the top, followed by a variant detection track, and in the lower part of the figure, the reference sequence with CDS annotations.

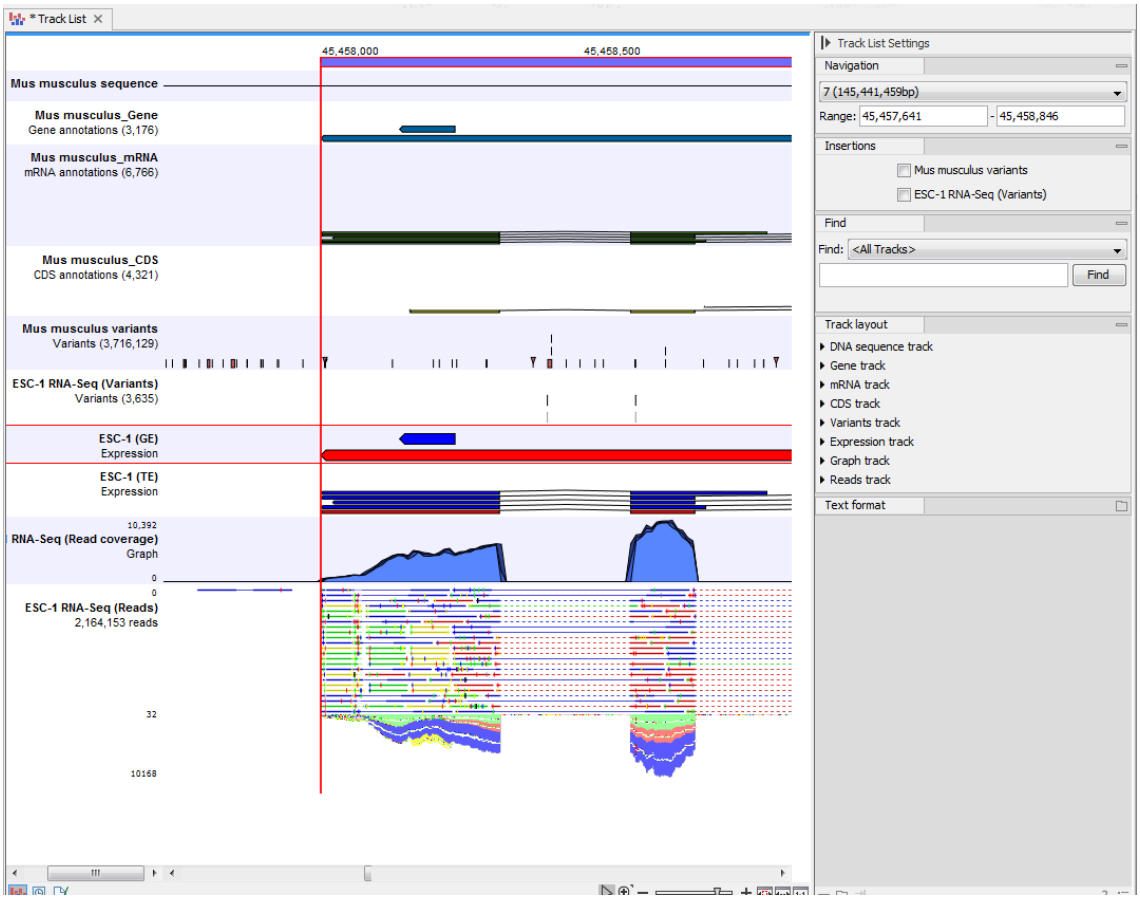



Figure 24.2: *A tracklist containing different types of tracks. From the top: a sequence track, three annotation tracks with gene, mRNA and CDS annotations respectively, two variant tracks, a gene-level (GE) and a transcript level (TE) expression track, a coverage track and a reads track.*

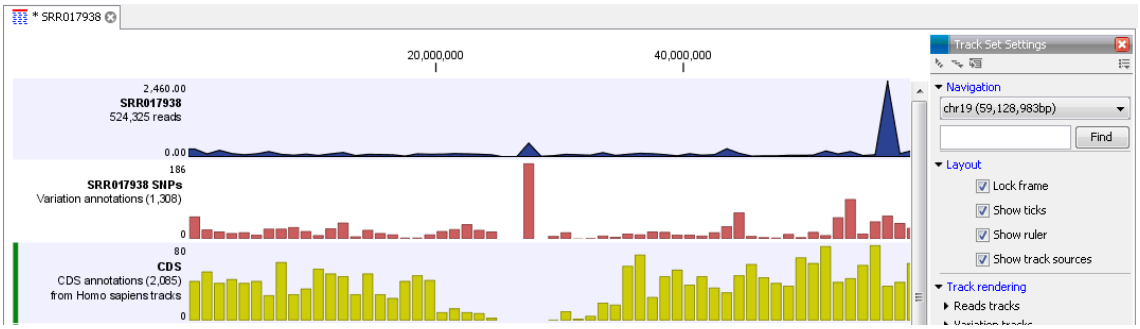



Figure 24.3: *Three tracks shown in the track list view*

### 24.1.1 Zooming and navigating track views

It is possible to zoom in and out with the zoom tools in the lower right-hand corner of the View Area, or by using a mouse scroll wheel while pressing the Ctrl (⌘ on Mac) key.

When zooming in and out you will see that, when zoomed out, the data is visualized in an aggregated format using a density bar plot or a graph. An aggregated Reads Track is shown in figure 24.4. The three blue shades shown in the Reads Track represent the aggregated mapped reads. This is displayed when the amount of data in view reaches the data aggregation setting.

Within the specified aggregation bucket, from top to bottom the blue colors mean:

- The maximum coverage value (read count).
- The average coverage value.
- The minimum coverage value.

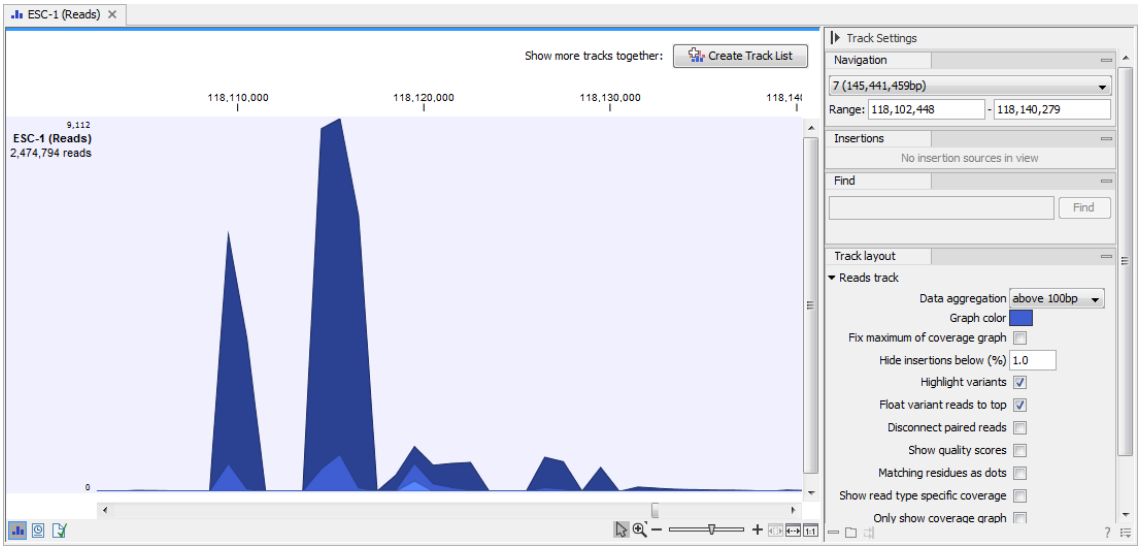



Figure 24.4: *In the aggregated Reads Track the three blue shades represent the aggregated mapped reads.*

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 and when used in combination with other tracks, this view can be used to get an overview of e.g. how many SNPs are located in a certain region.

In figure 24.5 we have zoomed in on a specific region of the view shown in 24.3, with a read track at the top showing the individual reads and with CDS and SNP annotations shown below.

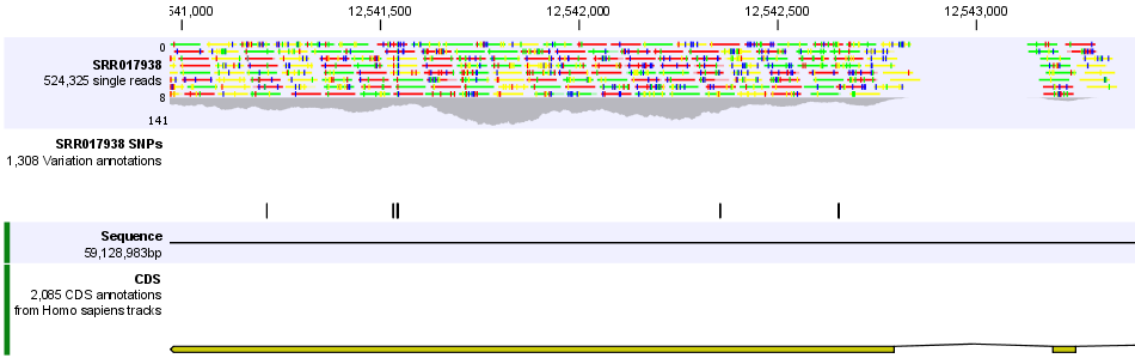



Figure 24.5: *Zooming in on the tracks reveals details.*

If you zoom in further the alignment of the reads and the reference sequence can be viewed at single nucleotide level (see figure 24.6).

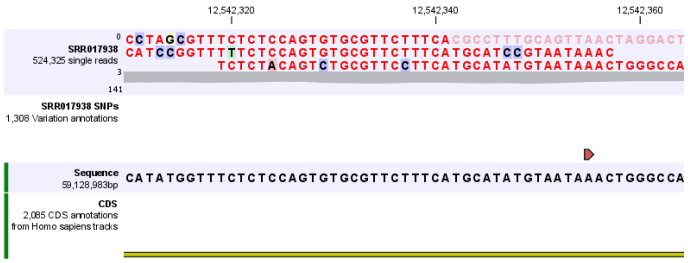



Figure 24.6: *Zoom in to see the bases of the reads and the reference sequence.*

In this case only three 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 (Figure 24.7).

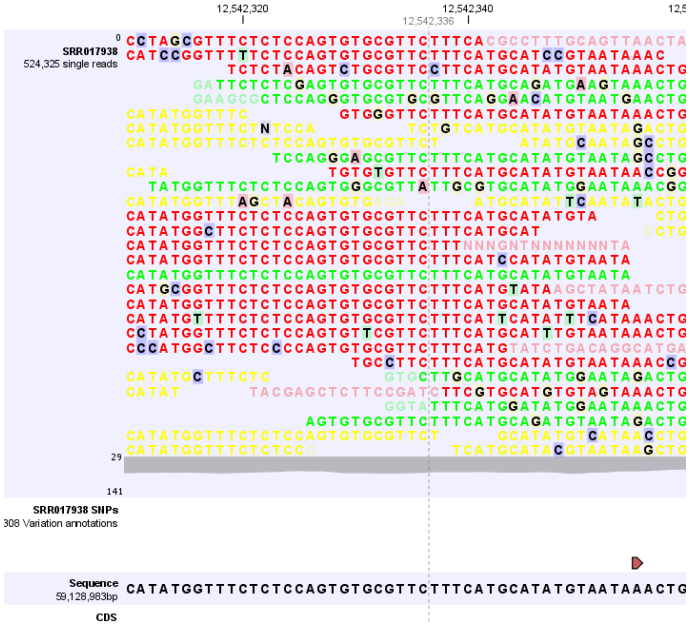



Figure 24.7: *Adjusting the height of the track.*

The options for the **Side Panel** vary depending on which track is shown in the View Area. In figure 24.8 an example is shown for a read mapping:

**Navigation.** Gives information about which chromosome is currently shown. Below this, you can see the start and end positions of the shown region of the chromosome. The drop-down list can be used to jump to a different chromosome. It is also possible to jump to a new position. This can be done by typing in the start and end positions in the text fields. Thousands separators are supported. The selected region will automatically appear in the viewing area.

**Insertions.** Only relevant for variant tracks.

**Find.** Not relevant for reads tracks.

**Track layout.** The options for the Track layout varies depending on which track type is shown. The options for a read 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 24.8 shows the options for a read track and an annotation track. The data aggregation settings can be adjusted for each displayed track type.

- **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 read tracks directly comparable with each other. Applies across all of the read mapping tracks.
- **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
- **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.
- **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.
- **Matching residues as dots.** Replaces matching residues with dots, only variants are shown in letters.
- **Show read type specific coverage.** When enabled, the coverage graph that summarizes those reads that could not be explicitly shown is now replaced by one coverage graph for each read type found in the Reads track. This could for instance be used for easy and visual comparison of the strand specific coverage.
- **Only show coverage graph.** When enabled, only the coverage graph is shown and no reads are shown.
- **Highlight reverse paired reads.** When enabled, read pairs with reverse orientation are highlighted with a light blue color.

When working with other track types such as gene tracks, other options are available:

*Labels.* Controls where in relation to the annotation features the labels will be shown, i.e. **Flag** places a label at the beginning of the gene, and above the feature graphics as shown in figure 24.9.

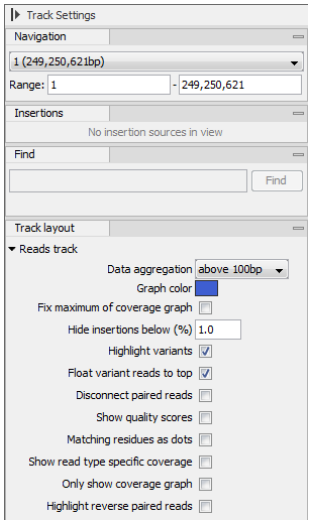



Figure 24.8: *The Side Panel for reads tracks.*

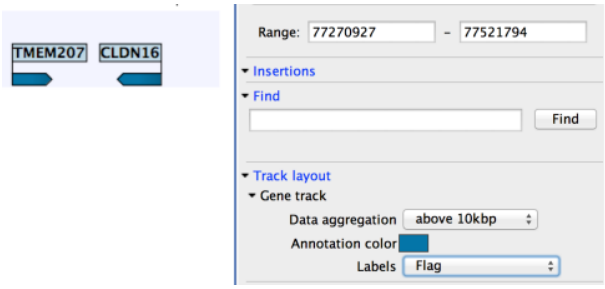



Figure 24.9: *The Side Panel for annotation tracks.*

### 24.1.2 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 24.10). 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.

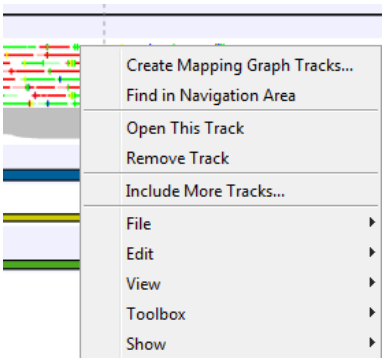



Figure 24.10: *Options to handle and organize tracks.*

**Create Mapping Graph Tracks** This will allow you to create a new track from a mapping track (learn more in section 24.6).

**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 annotation and variant tracks, a table view is opened as described in section 24.1.3. 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.

### 24.1.3 Showing a track in a table

All tracks containing annotations (including variants) can be opened in a table. From the track list (see section 24.1) this is done either by double-clicking the label of the track or by right-clicking the track and choosing **Open This Track**. Alternatively, you can open the track from the **Navigation Area** and switch to the table view ( ) at the bottom.

The table will have one row for each annotation, and the columns will reflect its information content. Figure 24.11 shows an example of a variant database track that is presented in a table.

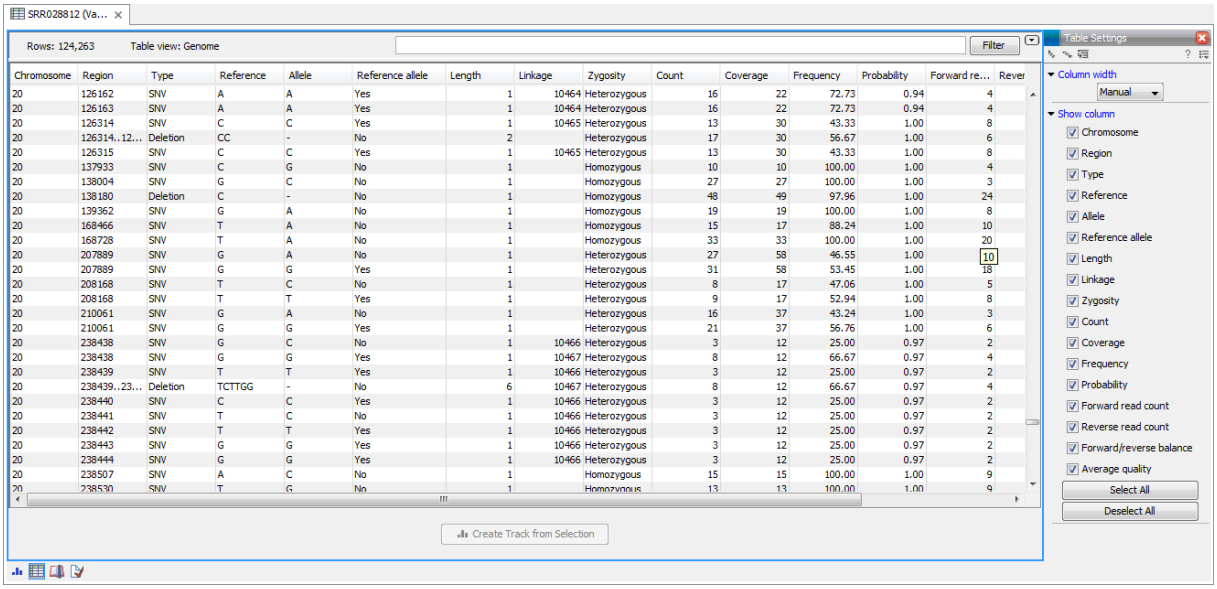



Figure 24.11: *Showing a variant track in a table view.*

You can use the table to sort, filter and select annotations (see Appendix 8.3). 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 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.

### 24.1.4 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 24.12) 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.

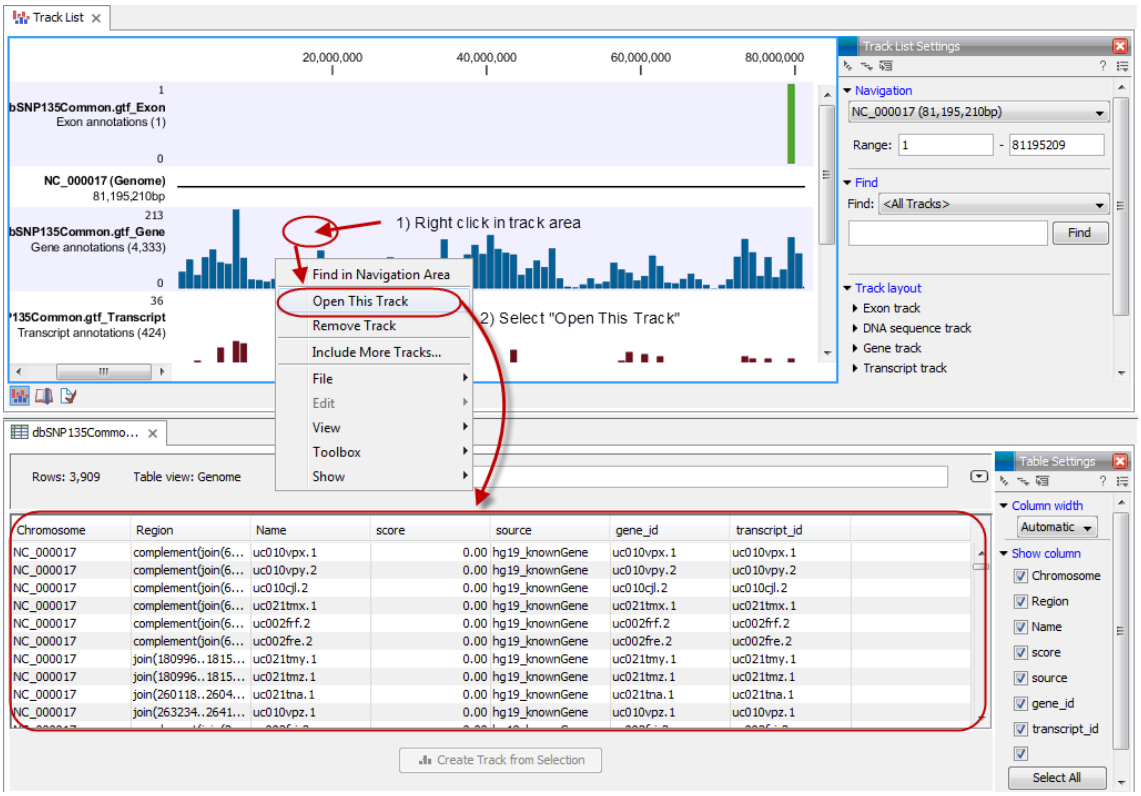



Figure 24.12: *One way to open a table view of a track that is part of a track list is to right click on the track of interest and select "Open This Table".*

### 24.1.5 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 24.13.

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

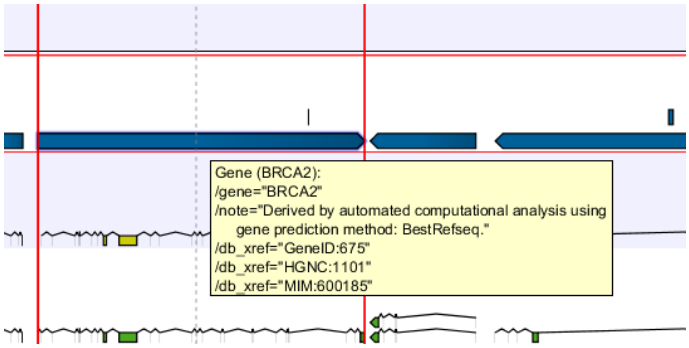



Figure 24.13: *The BRCA2 gene.*

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 24.13 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 tool tip with more info (see 24.14).

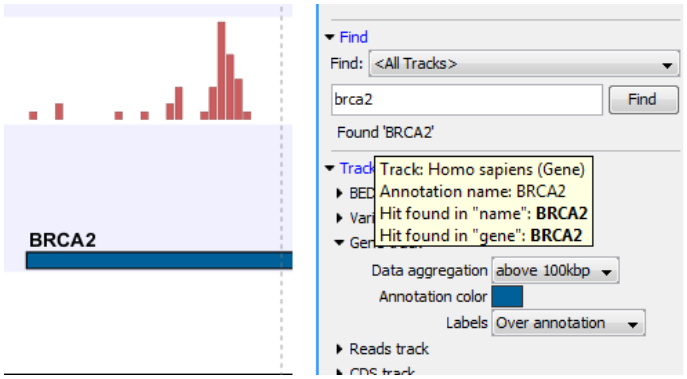



Figure 24.14: *The BRCA2 gene found.*

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 (see section 24.1.3).

### 24.1.6 Extract sequences from tracks

Like for all other sequence lists (see section 14.2), it is possible to extract sequences from tracks. The sequence of interest can be selected by dragging the mouse over the region of interest followed by a right click on the reads and a click on **Extract sequences** (figure 24.15).

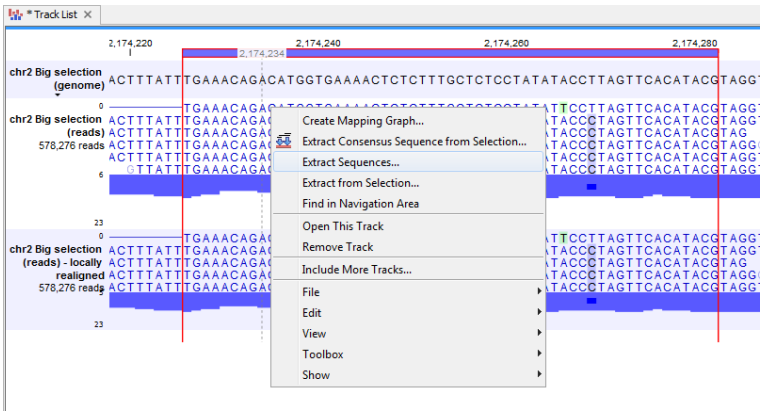



Figure 24.15: *Extract sequences from tracks.*

This opens up the dialog shown in figure 24.16 that allows specification of whether the selected sequences should be extracted as single sequences or as a list of sequences.

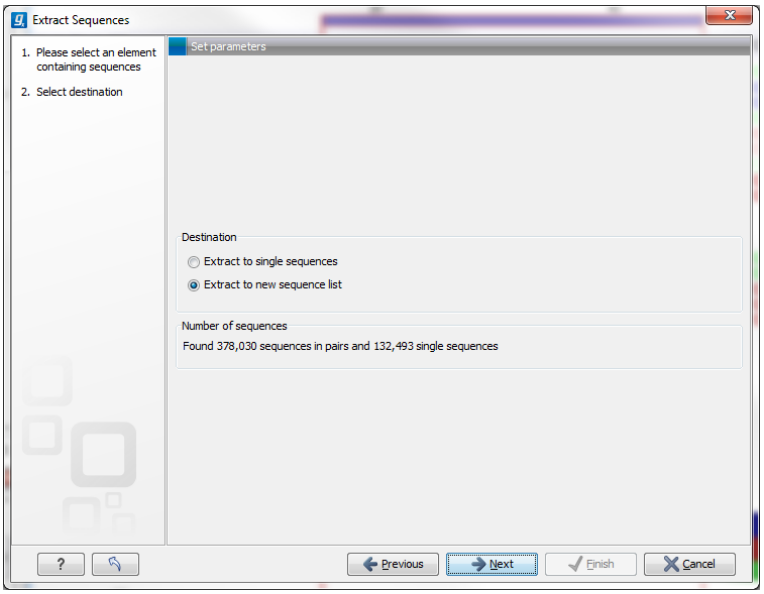



Figure 24.16: *Select destination for extracted sequences.*

Right clicking on the reads also enable the option **Extract from selection**, a function that corresponds to the **Extract from selection** described in section 18.7.6 although with small differences. Common for both versions of the **Extract from selection** function is that when extracting reads in an interval, only reads that are completely covered by the selection will be part of the extracted sequence, which in turn means that the tool can be used to extract only a subset of reads.

Clicking **Extract from selection** opens up the dialog shown in figure 24.17.

The purpose of this dialog is to let you specify which kinds of reads you wish to include. Per default all reads are included.

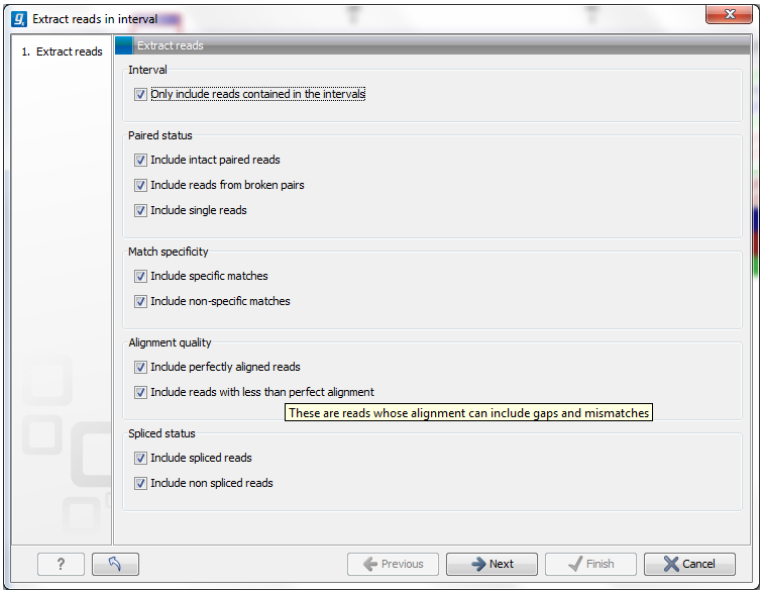



Figure 24.17: *Select the reads to include.*

The options are:

### Interval

**Only include reads contained within the intervals** Only reads that are included within the selection will be extracted. Reads that continue outside the selected area are not included.

### 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.

### 24.1.7 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 24.18 shows an example where two tracks are fed into the **Create Track List** element.

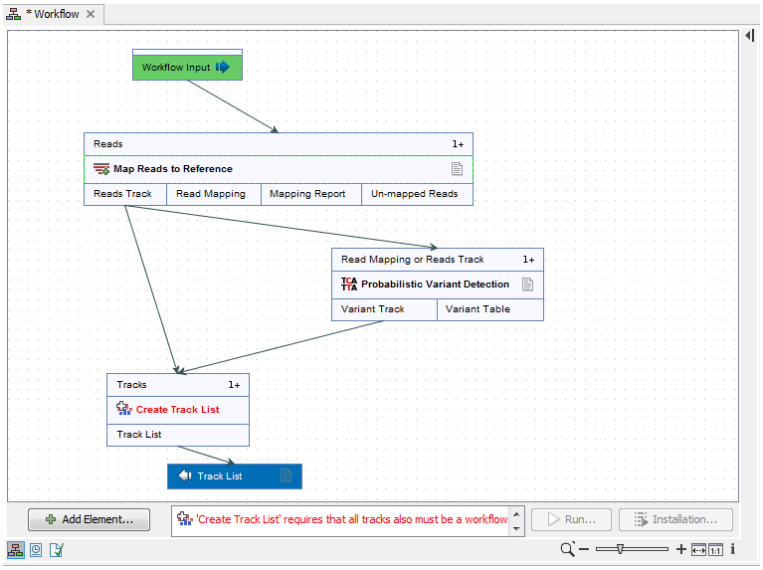



Figure 24.18: *This workflow does not work because the two tracks need to be marked as output.*

In this 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. In figure 24.19, this has been corrected by selecting the tracks as output, and the workflow can now be executed.

## 24.2 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 11.4).

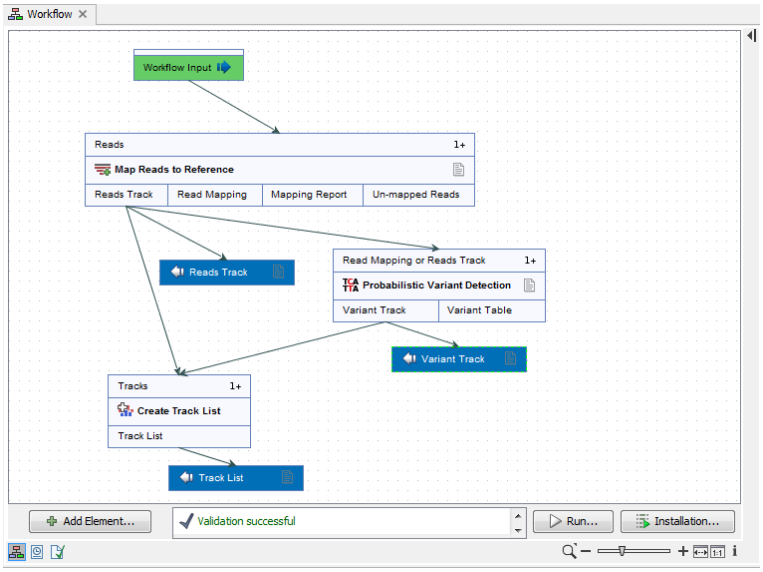



Figure 24.19: *The tracks have been selected as output, and the workflow can now be executed.*

2. Import tracks from files (learn more in section 6.2).

3. Convert sequences with annotations to tracks (learn more in section 24.4). 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 11.1).

   **Contigs created from de novo assembly.** Contig sequences from de novo assembly (see section 29.1) can be considered a reference genome for e.g. subsequent resequencing analysis applications.

4. Use the special plugins that integrate with Biobase's Genome Trax (learn more at `http://www.clcbio.com/clc-plugin/biobase-genome-trax-download/`).

Please note that tracks are not yet supported with the transcriptomics tools of *CLC Genomics Workbench*. This means you have to provide standard sequences (downloading from NCBI or importing files).

## 24.3 Merging 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 Merging 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 27.14 instead).

## 24.4 Converting data to tracks and back

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.

### 24.4.1 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** ( ) | **Convert to Tracks**

The following kinds of data can be converted to tracks: nucleotide sequences ( ), sequence lists ( ), read mappings ( )/ ( ), and the mapping and annotation information from RNA-Seq results ( ). Select the input and click **Next** to specify which tracks should be created (see figure 24.20).

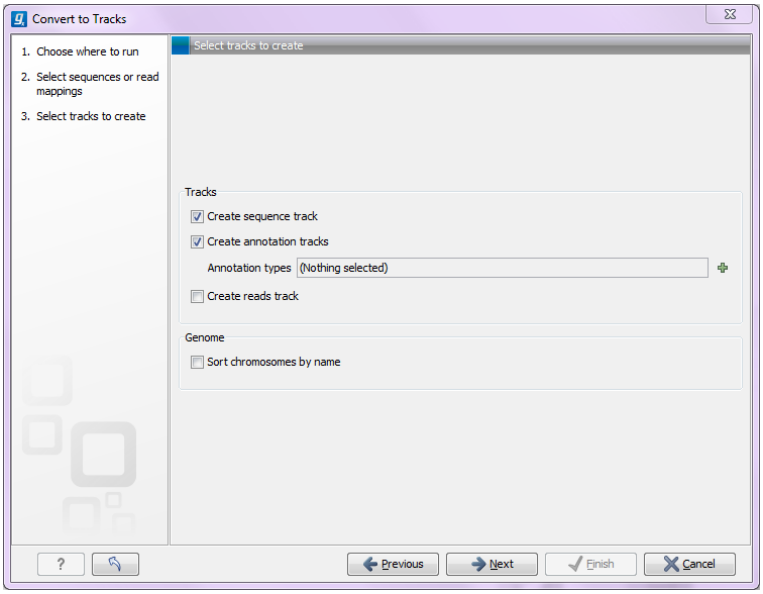



Figure 24.20: *Converting data to tracks.*

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.

### 24.4.2 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** ( ) | **Convert from Tracks**

One or more tracks can be used as input. In the example given in figure 24.21 a reads track and two annotation tracks are converted simultaneously to an annotated read mapping (figure 24.22).

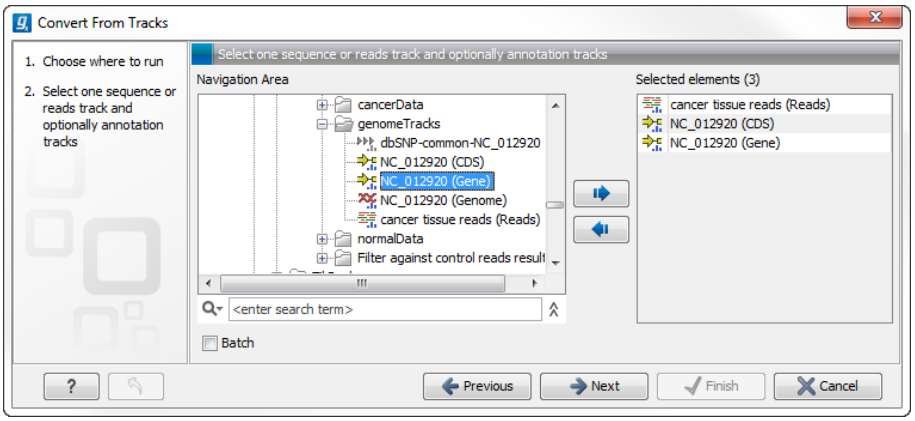



Figure 24.21: *A reads track and two annotation tracks are converted from track format to stand alone format.*

Likewise it is possible to create an annotated, stand-alone reference from a reference track and the desired number of annotation tracks. This is shown in figure 24.23 where one reference and two annotation tracks are used as input.

The output is shown in figure 24.24. The reference sequence has been transformed to stand alone format with the two annotations "CDS" and "Gene".

Depending on the input provided, the tool will create one of the following types of output:

**Sequence ( )** Will be created when a sequence track ( ) with a genome with only one sequence (one chromosome) is provided as input

**Sequence list ( )** Will be created when a sequence track ( ) with a genome with several sequences (several chromosomes) is provided as input

**Mapping ( )** Will be created when a reads track ( ) with a genome with only one sequence (one chromosome) is provided as input.

**Mapping table ( )** Will be created when a reads track ( ) with a genome with several sequences (several chromosomes) is provided as input.




Figure 24.22: *The upper part of the figure shows the three input tracks. For simplicity the three individual tracks are shown in a track list. The lower part of the figure shows the resulting stand alone annotated read mapping.*

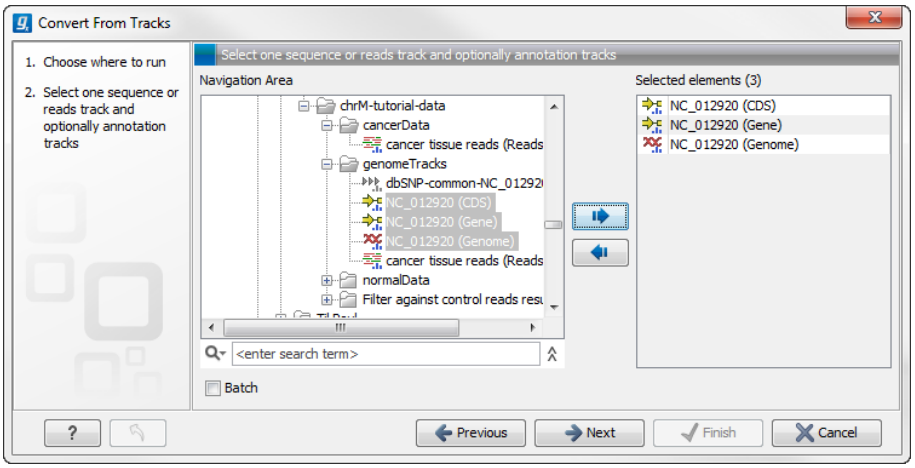



Figure 24.23: *A reference track and two annotation tracks are converted from track format to stand alone format.*

In all cases, any number of annotation tracks ( )/ ( ) can be provided, and the annotations will be added to the sequences (reference sequence for mappings) as shown in figure 24.22.

## 24.5 Annotate and filter tracks

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 27).

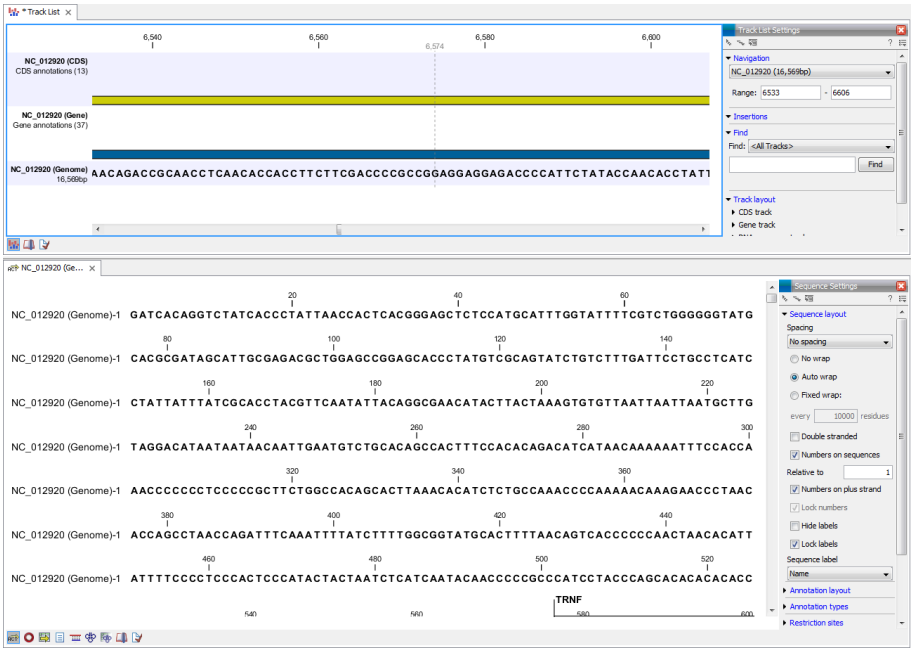


Figure 24.24: *The upper part of the figure shows the three input tracks. For simplicity the three individual tracks are shown in a track list. The lower part of the figure shows the resulting stand alone annotated reference sequence.*

### 24.5.1 Annotate with overlap information

This will create a copy of the track used as input and add information from overlapping annotations or variants:

**Toolbox** | **Track Tools** ( ) | **Annotate and Filter** | **Annotate with Overlap Information**

First, select the track you wish to annotate and click **Next**. You can choose any kind of variant or annotation track as input. Next, select the track for overlap comparison, you can choose any variant, annotation, or RNA-seq expression track.

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. The requirement for being registered as an overlap is that parts of the annotations are overlapping, regardless of the strandedness of the annotations (note that this makes it unsuitable for comparing e.g. two gene tracks but great for annotating variants with overlapping genes or regulatory regions).

When running the "Annotate with overlap information" tool with a gene track as input and a variant track as parameter track, a new column describing the specific variant is added to the Track Table. The variant description also appears in the track tooltips when mousing over the individual variants.

### 24.5.2 Extract reads based on overlap

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** | **Track Tools** ( ) | **Annotate and Filter** | **Extract Reads Based on Overlap** ( )

Read mapping tracks can be used as input as shown in the dialog in figure 24.25.

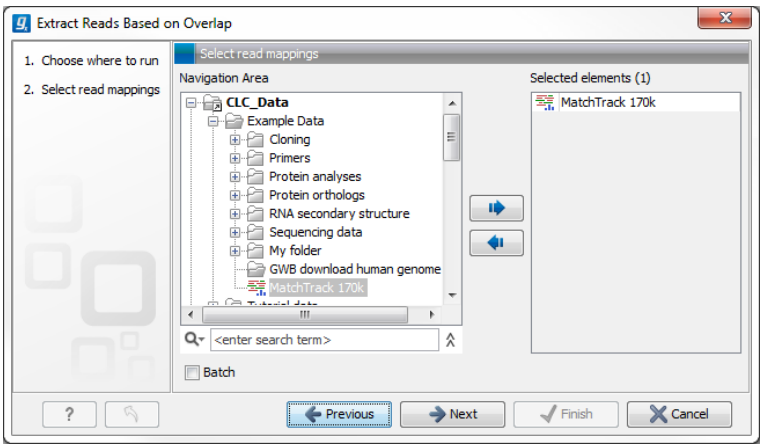



Figure 24.25: *Select a read mapping. Only one read mapping can be selected at the time.*

The next step is to select the annotated track(s) to be used for pulling out reads and specify which reads to include (figure 24.26).

The options in this wizard are:

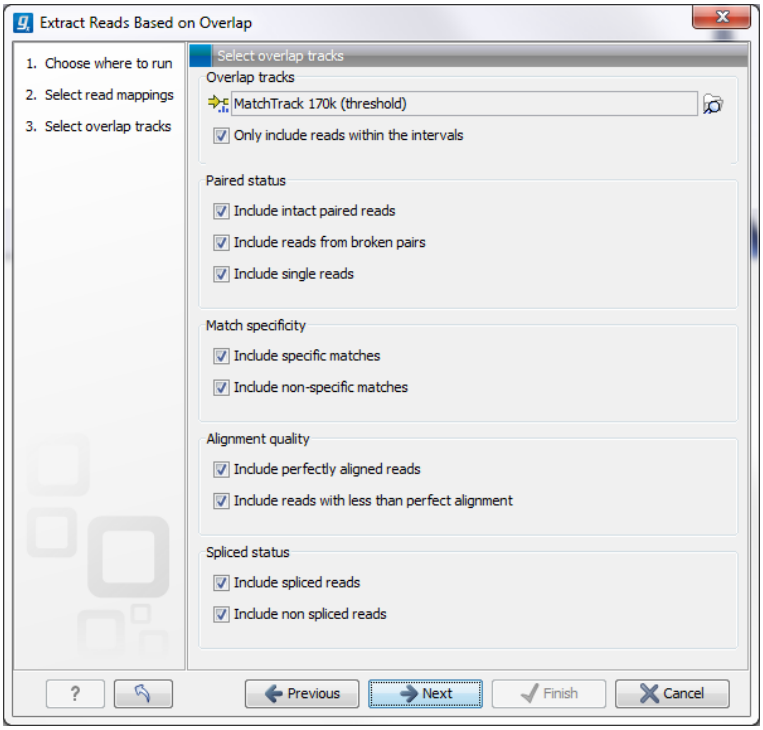



Figure 24.26: *Select the track(s) containing the annotation(s) of interest. Multiple tracks can be selected at the same time.*

### Overlap tracks

Select the annotated track

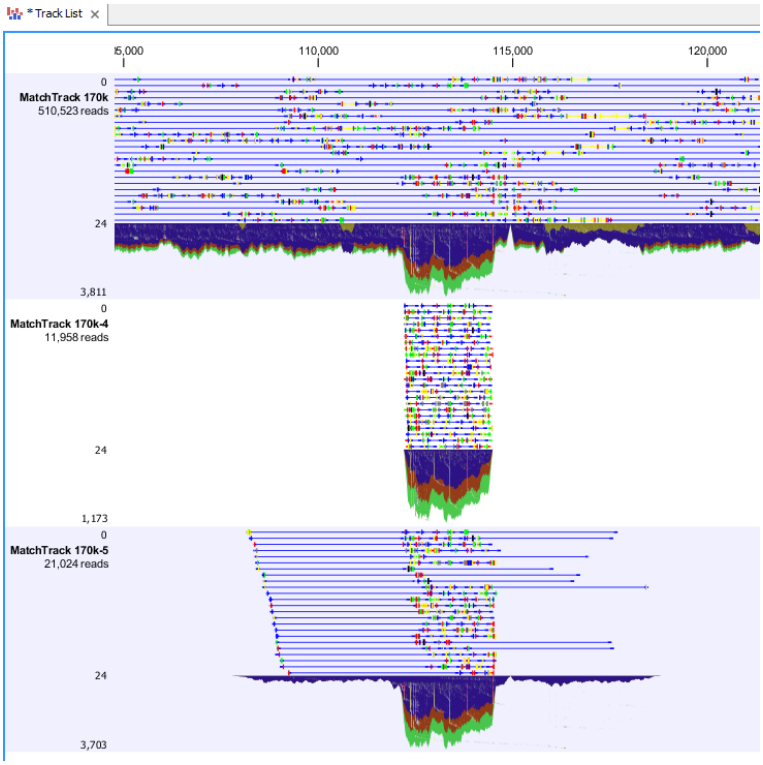



Figure 24.27: *Output from Extract reads based on overlap. The overlap track used as input was generated using the "Identify Graph Threshold Areas". Top: The read mapping used as input, middle: Output when "Only include reads within intervals" has been ticked, bottom: Output when "Only include reads within intervals" has been deselected.*

**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 24.27.

### 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.

### 24.5.3 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 24.5.4) 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**.

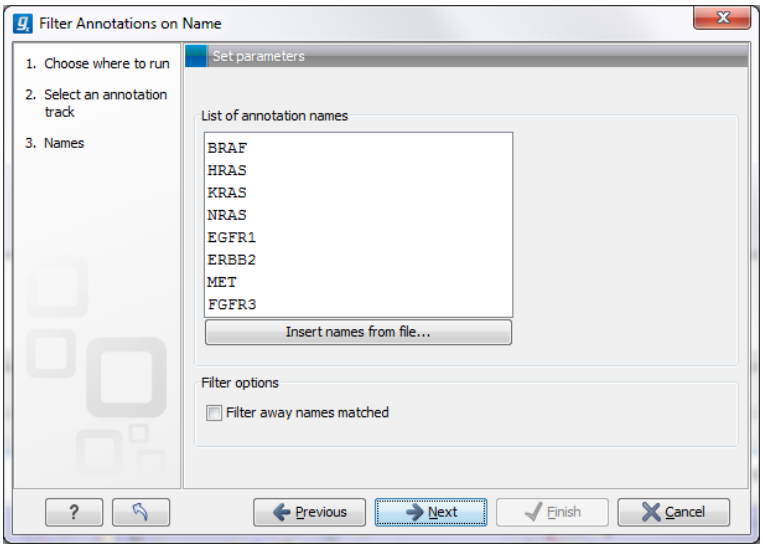



Figure 24.28: *Specify names for filtering.*

As shown in figure 24.28, 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 to reduce the risk of coincidental findings when analyzing patient samples).

### 24.5.4 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 24.5.3. Please note that for comparing variant tracks, more specific filters should be used (see section 27.13.1).

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 24.1.3).

**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 24.29).

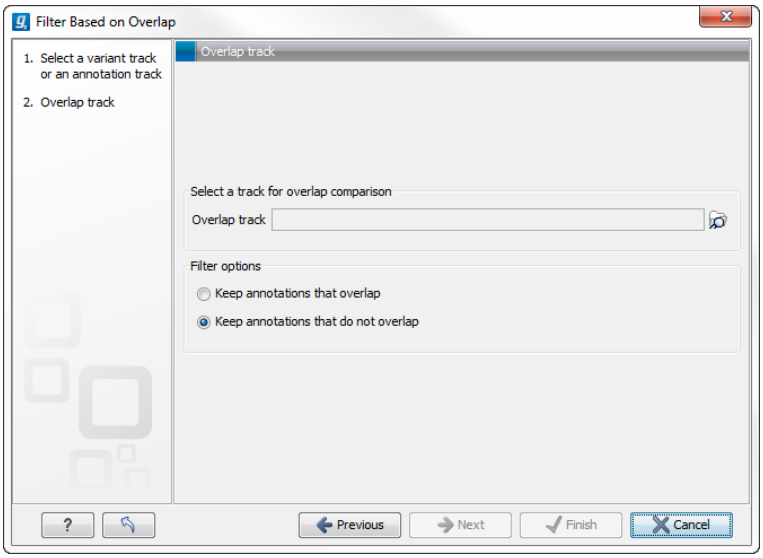



Figure 24.29: *Select overlapping annotations track.*

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.

## 24.6 Creating graph tracks

Graphs can be a good way to quickly get an overview of certain types of information. This is the case for e.g. the GC content in a sequence or the read coverage. The *CLC Genomics Workbench*

offers two different tools that can create graph tracks from either a sequence or a read mapping. The two available tools are:

- **Create GC Content Graph**
- **Create Mapping Graph**

Both tools are found in the toolbox:

**Toolbox** | **Track Tools** ( ) | **Graphs**

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.

**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 24.30).

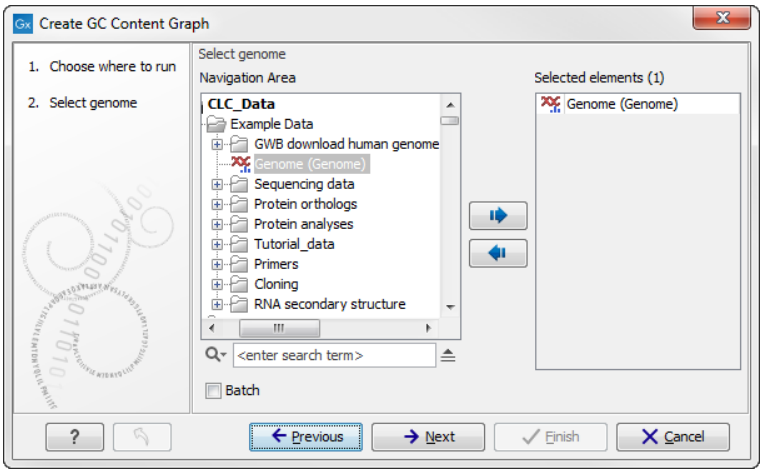



Figure 24.30: *Select the sequence track that should be used as input.*

Click on the button labeled **Next** to go to the next wizard step (see figure 24.31). In this wizard step you can specify the window size, which is 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. E.g. 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.

Click on the button labeled **Next**, choose to save your results, and click on the button labeled **Finish**. The output can be seen in figure 24.32. 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

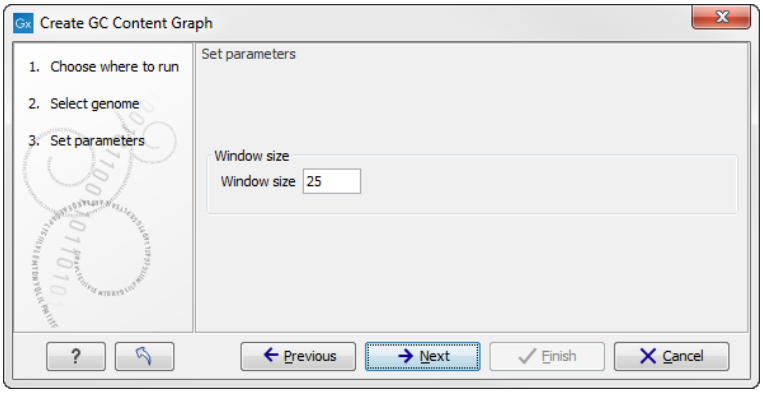



Figure 24.31: *Specify the window size. The window size is the region around each individual base that should be used to calculate the GC content in a given region.*

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.

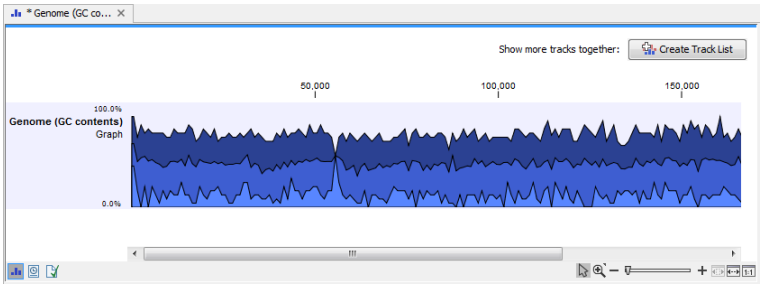



Figure 24.32: *The output from "Create GC Content Graph" is a graph track. The graph track shows one value for each base with one graph being available for each chromosome.*

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. In stead, 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 24.33 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 24.1) 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).

**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 24.34 and click on the button labeled **Next**.

Select the graph tracks that you would like to create. The following options exist:

- Read coverage. For each position this graph shows the number of reads contributing to the

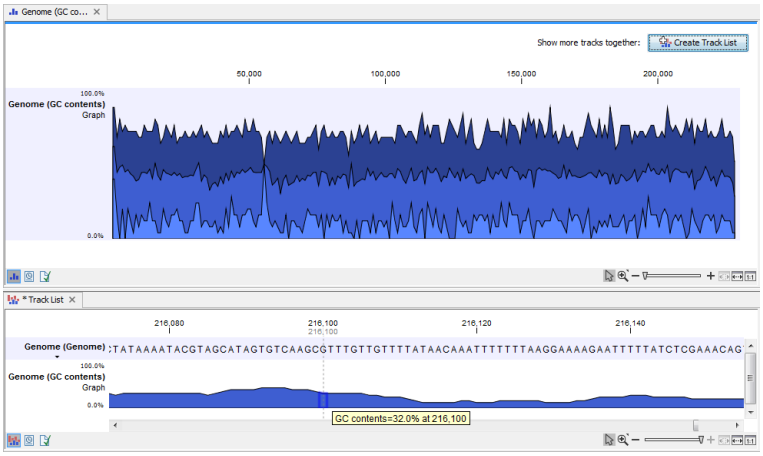



Figure 24.33: *The top part of the figure shows the graph track when zoomed all the way out. The bottom part of the window shows a graph track together with the input genomic sequence at single nucleotide resolution. By mousing over one nucleotide, you can see the GC content for this position. In our 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.*

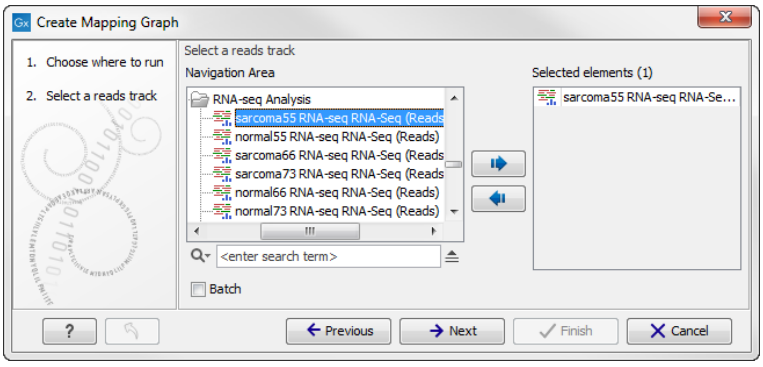



Figure 24.34: *Creating graph track from mappings.*

alignment (see a more elaborate definition in section 25.2).

- Non-specific read coverage. Non-specific reads are reads that would fit equally well other places in the reference genome.
- Unaligned ends coverage. Un-aligned ends arise when a read has been locally aligned to a reference sequence, and then end of the read is left unaligned because there are mismatches or gaps relative to the reference sequence. This part of the read does not contribute to the read coverage above. The unaligned ends coverage graph shows how many reads that have unaligned ends at each position.
- Non-perfect read coverage. Non-perfect reads are reads with one or more mismatches or gaps relative to the reference sequence.
- Paired read coverage. This lists the coverage of intact pairs. If there are no single reads and no pairs are broken, it will be the same as the standard read coverage above.
- Broken pair coverage. A pair is broken either because only one read in the pair matches, or because the distance or relative orientation between the reads is wrong.

- Paired end distance. Displays the average distance between the forward and the reverse read in a pair. A pair contributes to this graph from the beginning of the first read to the end of the second read.

One graph track output will be created for each of the graph tracks you have chosen by checking the boxes shown in figure 24.35.

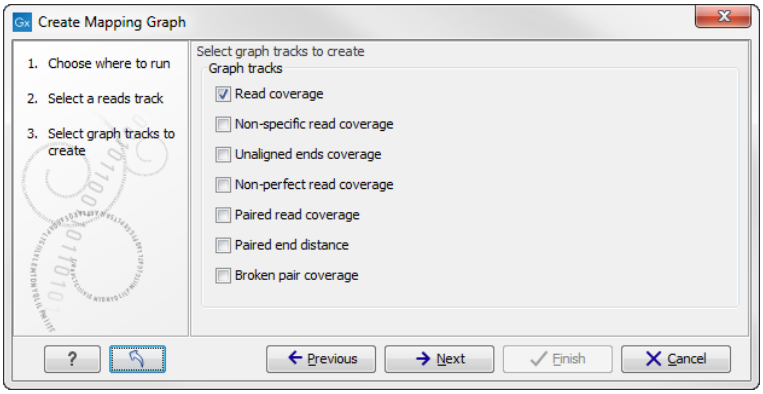



Figure 24.35: *Choose the types of graph tracks you would like to generate.*

Click on the button labeled **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 24.36. Two of the views have been dragged and dropped to other areas of the **View Area** to be able to see them in the same window. If you would like to learn more about how to do this, please refer to section 2.1.6.

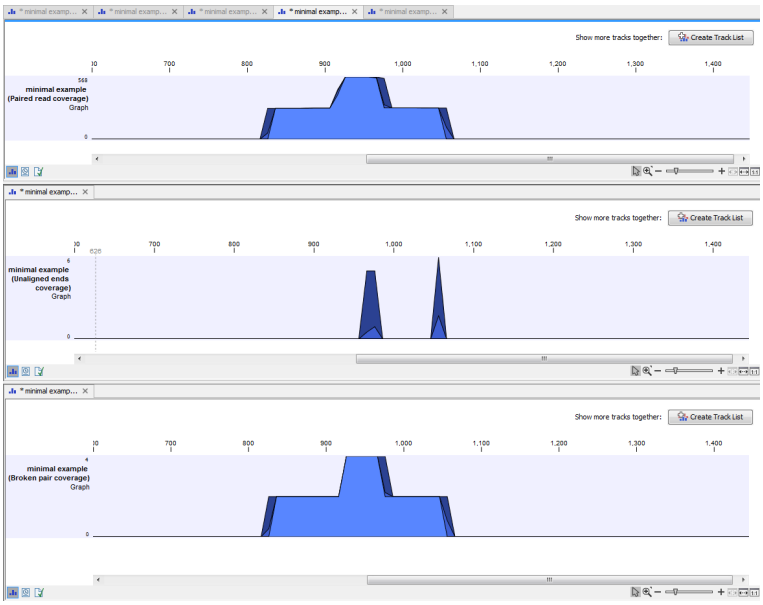



Figure 24.36: *Three types of graph tracks are shown.*

In this example we generated all possible outputs and chose to open them without saving. You can see that the names of the tabs are marked with an asterisk, which indicates that the graph shown in the view area has not been saved or that changes have been made that must be saved if you want to keep them. Three of the generated outputs have been opened. If you would like to see the outputs in the same view, you can do this by creating a track list. Click on the button

labeled **Create Track List** in the upper right corner of each of the graph tracks shown in the **View Area**. Combining graph tracks in a track list links the individual tracks together, which makes it much easier to compare the different graph tracks.

Another thing that is worth noting is the options found in the **Side Panel**. In particular the option "Fix graph bounds" found under **Track layout** is useful to know if you would like to manually adjust the numbers on the y-axis.

**Identify Graph Threshold Areas**

The **Identify Graph Threshold Areas** tool uses graph tracks as input to identify graph regions that fall within certain limits (thresholds). 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 24.37). Consequently, in order to identify only those parts of the track that exceed a certain minimum, one would choose the minimum threshold and set the upper limit to a value well above the maximum occurring in the track (and vice versa for finding ranges that are below a maximum threshold). Obviously, 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, hence resulting in the same "non-smoothing" behavior as previous versions of the workbench without this parameter. 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. An example output is shown in figure 24.38 where the coverage graph has a couple of local minima near zero. 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.

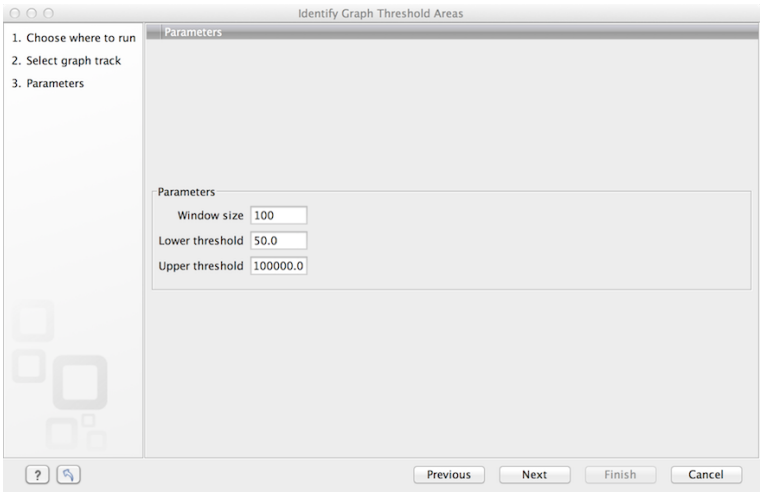



Figure 24.37: *Specification of lower and upper thresholds.*

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 24.38). When zoomed in all the way

down to base resolution only one curve will be shown reflecting the exact observation at each individual position.

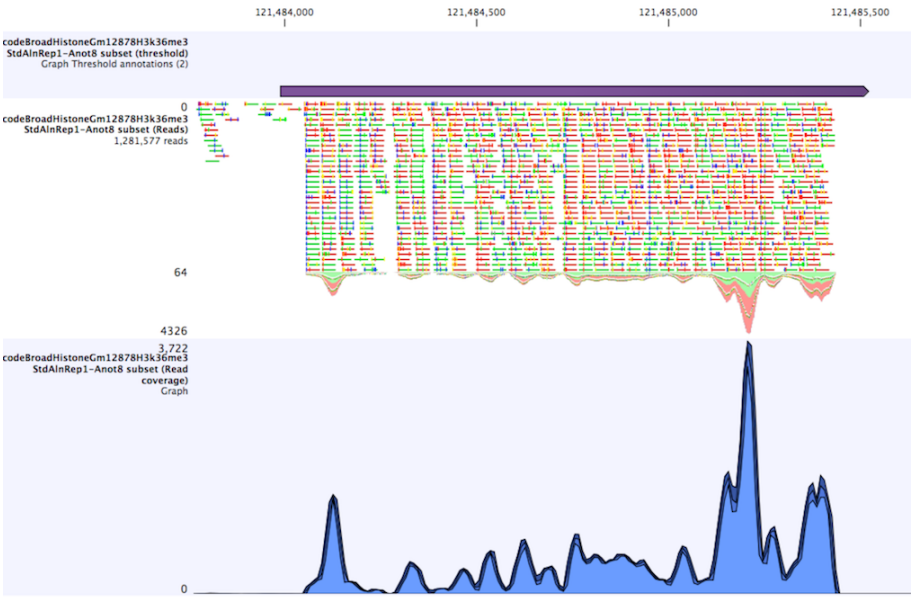



Figure 24.38: *Track list including a region identified by the parameters set above on a dataset of H3K36 methylation from ENCODE. The top track shows the resulting region. Below is the track containing the reads. The graph track at the bottom shows the coverage with the minimum, mean, and maximum observed values.*

# Chapter 25

# Read mapping

## Contents


**25.1 Map Reads to Reference . . . 572**
25.1.1 Selecting reads and reference . . . 572
25.1.2 Including or excluding regions (masking) . . . 573
25.1.3 Mapping parameters . . . 573
25.1.4 Gap placement . . . 579
25.1.5 Computational requirements . . . 580
25.1.6 Reference Caching . . . 581
**25.2 Mapping output options . . . 581**
**25.3 Mapping reports . . . 582**
25.3.1 Detailed mapping report . . . 582
25.3.2 Summary mapping report . . . 589
**25.4 Color space . . . 590**
25.4.1 Sequencing . . . 590
25.4.2 Error modes . . . 590
25.4.3 Mapping in color space . . . 591
25.4.4 Viewing color space information . . . 593
**25.5 Mapping result . . . 594**
25.5.1 Mapping table . . . 595
25.5.2 View settings in the Side Panel . . . 597
25.5.3 Overlapping paired reads . . . 599
25.5.4 Find broken pair mates . . . 600
**25.6 Local realignment . . . 601**
25.6.1 Method . . . 603
25.6.2 Realignment of unaligned ends . . . 604
25.6.3 Guided Realignment . . . 604
25.6.4 Multi-pass local realignment . . . 606
25.6.5 Known Limitations . . . 606
25.6.6 Computational Requirements . . . 607
25.6.7 How to run the Local Realignment tool . . . 607
**25.7 Merge mapping results . . . 609**

**25.8 Extract consensus sequence . . . . . 610**
**25.9 Coverage analysis . . . . . 613**
25.9.1 Running the Coverage analysis tool . . . . . 613

## 25.1 Map Reads to Reference

Read mapping is a very fundamental step in most applications of high-throughput sequencing data. The *CLC Genomics Workbench* includes read mapping in several other tools (e.g. in the 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.

There are two different versions of the core mapper: one for color space data, and one for base space data. At `http://www.clcbio.com/white-paper` you can find white papers with detailed benchmarks and descriptions of both algorithms.

The following description focuses on the parameters that can be directly influenced by the user.

### 25.1.1 Selecting reads and reference

To start the read mapping:

**Toolbox** | **NGS Core Tools** ( ) | **Map Reads to Reference** ( )

In this dialog, select the sequences or sequence lists containing the sequencing data. Note that the reference sequences should be selected in the next step. Please note that reads longer than 100,000 bases are not supported.

When the sequences are selected, click **Next**, and you will see the dialog shown in figure 25.1.

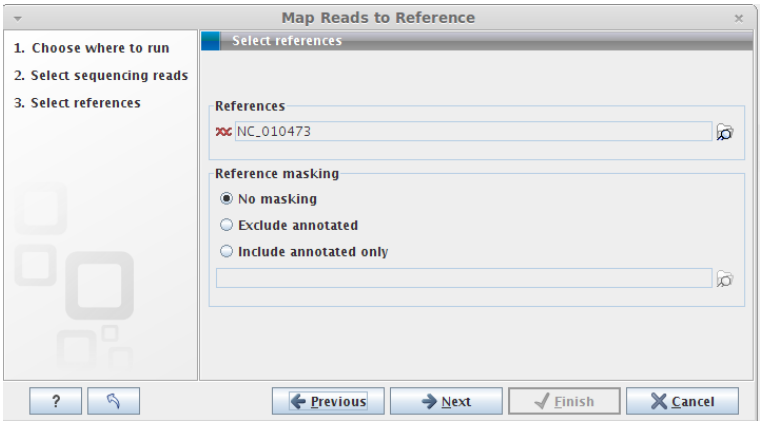



Figure 25.1: *Specifying the reference sequences and masking.*

At the top you select one or more reference sequences by clicking the **Browse and select element** ( ) button. You can select either single sequences, a list of sequences or a sequence track as reference. Note that single reference sequences longer than 2gb ($2 \cdot 10^9$ bases) are not supported.

### 25.1.2 Including or excluding regions (masking)

The next part of the dialog shown in figure 25.1 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. If you have annotations on a sequence instead of a track, you can convert the annotation type to a track (see section 24.4).

### 25.1.3 Mapping parameters

Clicking **Next** leads to the parameters for the read mapping (see figure 25.2).

The first parameter allows the mismatch cost to be adjusted:

**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:

**Linear gap cost** The cost of a gap is computed directly from the lenght of the gap and the insertion or deletion cost. This model often favors small, fragmented gaps over long contiguous gaps.

**Affine gap cost** An extra cost associated with opening a gap is introduced such that long contiguous gaps are favored over short gaps.

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. If you choose linear gap cost, you must set the insertion cost and the deletion cost:

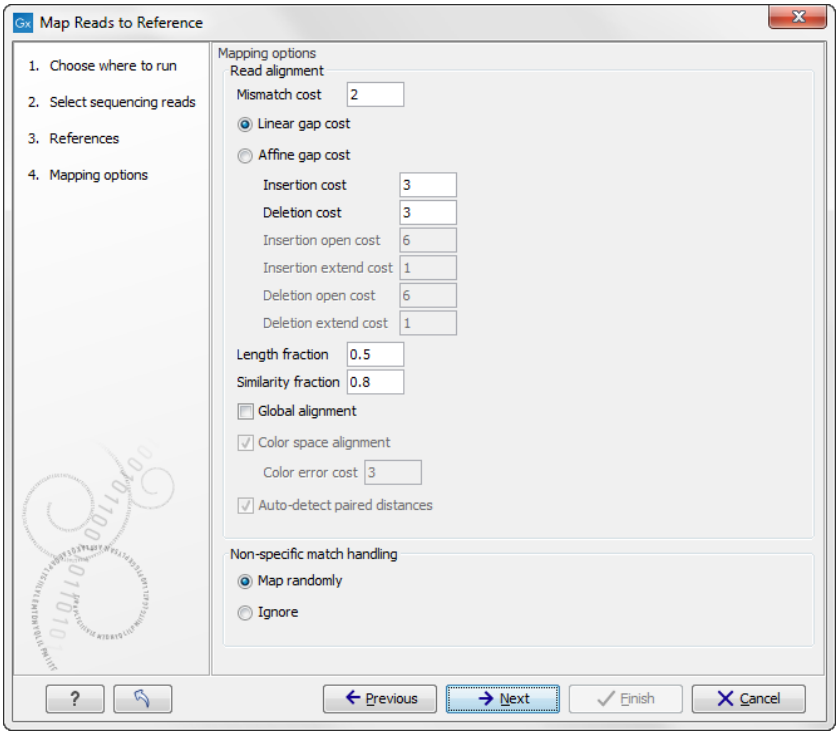



Figure 25.2: *Setting parameters for the mapping.*

**Insertion cost** The cost of an insertion in the read (a gap in the reference sequence).

**Deletion cost** The cost of a deletion in the read (gap in the read sequence).

In this case, the cost of an insertion of length $\ell$ will be

$$\ell \cdot \text{Insertion cost}, \tag{25.1}$$

and, similarly, the cost of a deletion of length $\ell$ will be

$$\ell \cdot \text{Deletion cost}. \tag{25.2}$$

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 $\ell$ is penalized by

$$\text{Insertion open cost} + \ell \cdot \text{Insertion extend cost}, \tag{25.3}$$

and, similarly, a deletion of length $\ell$ is penalized by

$$\text{Deletion open cost} + \ell \cdot \text{Deletion extend cost.} \tag{25.4}$$

In this way long consequtive 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 always 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.

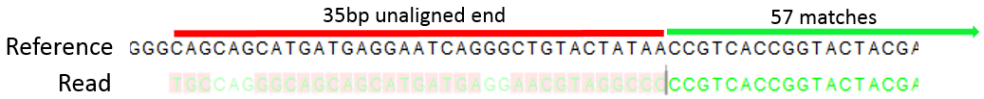



Figure 25.3: *An alignment of a read where a region of 35bp at the start of the read is unaligned while the remaining 57 nucleotides matches the reference.*

Figure 25.3 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 25.4 using the default scoring scheme, the score would become:

$$(26 + 1 + 3 + 57) * 1 - 5 * 2 - 8 * 3 = 53 \tag{25.5}$$

In this case, the alignment shown in Figure 25.3 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 25.4 would become 61 and 58, respectively, and thus make it optimal.

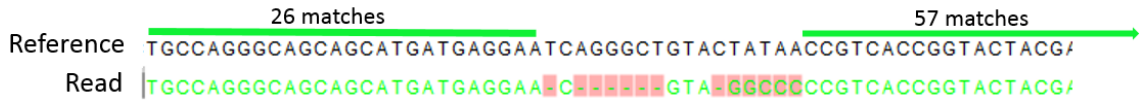



Figure 25.4: *An alignment of a read containing a region with several mismatches and deletions. By reducing the default cost of either mismatches or deletions the read mapper can make an alignment that spans the full length of the read.*

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 25.5).

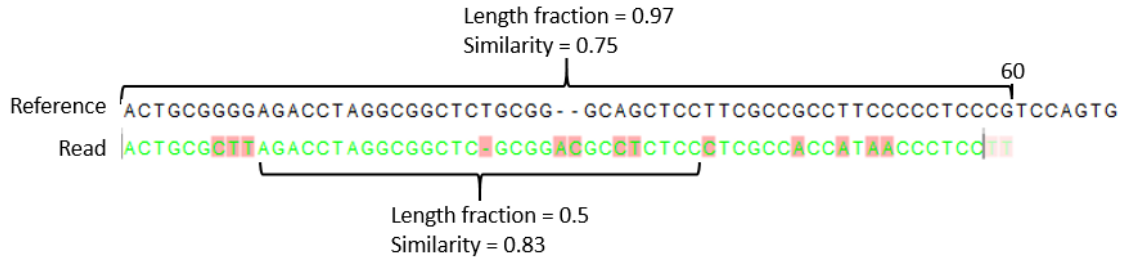



Figure 25.5: *A read containing 59 nucleotides where the total alignment length is 60. The part of the alignment that gave rise to the optimal score has length 58 which excludes 2 bases at the left end of the read. The length fraction of the matching region in this example is therefore 58/60 = 0.97. 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.*

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 25.2.

When mapping data in color space (data from SOLiD systems), the **color space** checkbox is enabled, and a corresponding cost for color errors can be set. If you do not have color space data, these will be disabled and are not relevant. For more details about this, please see section 25.4 which explains how color space mapping is performed in greater detail.

### Mapping paired reads

At the bottom of the dialog shown in figure 25.2 you can specify how **Paired reads** should be handled. You can read more about how paired data is imported and handled in section 6.3.8. 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 100,000 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, and an appropriate distance interval is selected:

- If less than 10,000 reads map, a simple calculation is used where the minimum distance is one standard deviation below the average distance, and the maximum distance is one standard deviation above the average distance.
- If more than 10,000 reads map, a more sophisticated method is used which 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 25.6 shows an example of the distribution of intervals with and without automatic pair distance interval estimation.

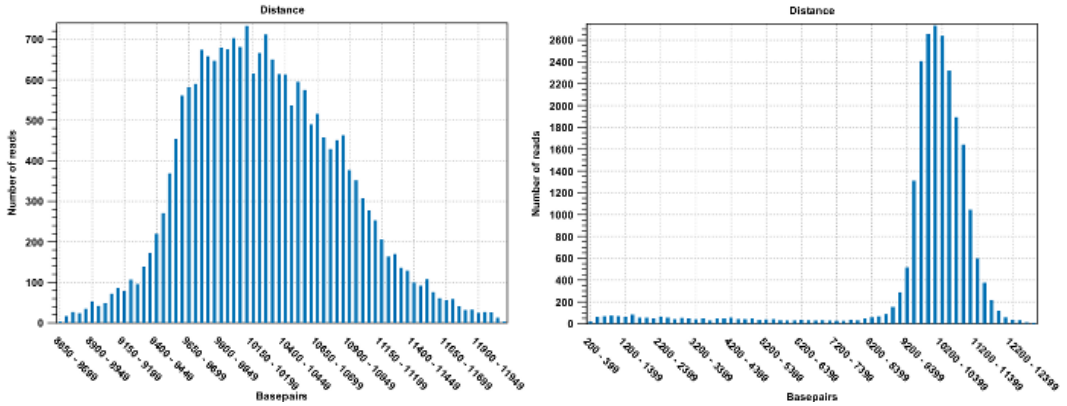



Figure 25.6: *To the left: mapping with a narrower distance interval estimated by the workbench. To the right: mapping with a large paired distance interval (note the large right tail of the distribution).*

Sometimes the automatic estimation of the distance between the pairs may return a warning "multiple intervals detected". This may happen if e.g. the reads derive from multiple libraries or from certain types of amplicon sequencing protocols. In this case, the estimates may still be correct, but, if in doubt, the user 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 6.3.8).

The 'automatic detection of paired distance' option when mapping should be used with caution. It is possible that the estimated distance setting is too narrow and consequently many read pairs will be flagged broken. Sometimes, a second peak in the Paired Distance distribution graph is not picked up on by the estimation tool.

If a large portion of pairs are flagged 'Broken' we recommend the following:

1. Inspect the detailed mapping report (see section 25.3.1) 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 25.3.1 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 25.1.3 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 25.1.3).

When looking at the mapping, the default color for non-specific matches is yellow.

### 25.1.4 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 25.7.




Figure 25.7: *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 25.8.




Figure 25.8: *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.

Many insertions and deletions in homopolymeric or repetitive regions reported in the public databases dbSNP and 1000Genomes have been identified based on mappings done with tools like BWA and Bowtie, that place insertions or deletions at the left side of a homopolymeric tract. Thus, to help facilitate the comparison of variant results with such public resources, the CLC bio **Map Reads to Reference** tool, as of version 6.5 of the *CLC Genomics Workbench*, will place insertions or deletions in homopolymeric tracts at the left hand side.

This is a change to earlier versions of the *CLC Genomics Workbench* (version 6.0.5 and earlier) where the CLC bio read mapper placed insertions and deletions in homopolymeric tracts at the right hand side of the homopolymer, as viewed in the Workbench.

This has the implication that insertion and deletion variants called in homopolymeric regions will be in different positions relative to the reference when based on mappings run in version 6.0.5 and earlier, compared to variant calls based on mappings run in version 6.5 and later. Thus, if comparisons between sample variant tracks will be done in the *CLC Genomics Workbench*, we

recommend re-running mappings so all samples are mapped using the mapping tool in version 6.5 of the *CLC Genomics Workbench* or higher, or all samples to be compared have been mapped using version 6.0.5 and lower.

**For users of the COSMIC database or other clinical databases following the recommendations from the Human Genome Variation Society (HGVS)**

The Human Genome Variation Society (HGVS) recommendations, which pertain to variants within genes, state that for insertions and deletions in homopolymeric or repetitive regions, the most 3' position (corresponding to the strand of the gene) possible should be arbitrarily assigned as the site of change (see http://www.hgvs.org/mutnomen/recs-DNA.html#del). Resources such as COSMIC adhere to these recommendations. In this case, placement to the farthest possible left hand position, as viewed in the *CLC Genomics Workbench*, of insertions or deletions in repetitive or homopolymeric tracts, has a different effect, depending on whether the gene involved is on the positive or negative strand of the reference. Such variants located within genes on the negative strand can be compared with the COSMIC database, while those within genes lying on the positive strand cannot be, as the positions relative to the reference will be different in this case. The opposite situation is true when variant calls are based on mappings run in version 6.0.5 of the *CLC Genomics Workbench* or earlier. That is, if comparing to a resource following HGVS recommendations, like COSMIC, insertions and deletions in homopolymeric or repetitive regions called within genes that lie on the positive strand will be comparable based on position relative to the reference, while those within genes on the negative strand will not be.

### 25.1.5 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. The color space mapper is able to scale down its memory consumption, such that even large references can be represented using small amounts of memory. However, when the memory consumption is scaled down it causes the read mapping to become slower.

When mapping short high quality reads, such as Illumina reads, the added memory consumption per CPU core is small. 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. 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.

### 25.1.6 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 8 GB which can be changed by going to

**Edit** | **Preferences** | **Advanced** | **Read Mapper**

On the server and for webstart the cache size can be controlled by creating a settings file "readmapper.properties" with an entry going like this "referencecachesize = 8589934592" where the size is in bytes. On grid setups, the "readmapper.properties" file will have to be added manually to each grid worker directory.

## 25.2 Mapping output options

Click **Next** lets you choose how the output of the mapping should be reported (see figure 25.9).

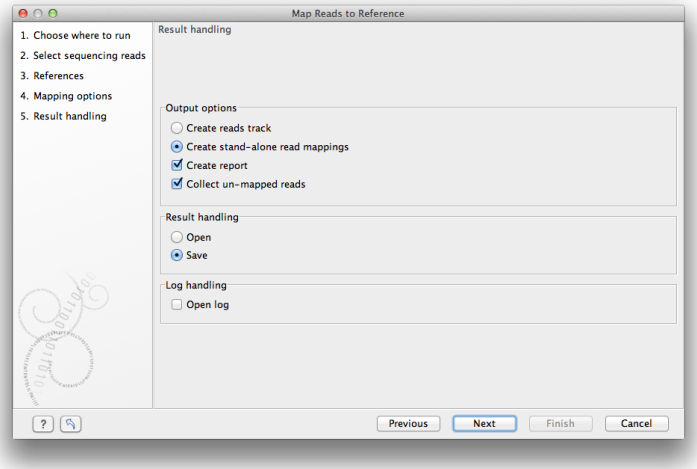



Figure 25.9: *Mapping output options.*

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 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. Details about viewing and editing of reads-tracks are described in section 24, and the resequencing is detailed in section 27. 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. In contrast, the stand-alone read mapping output has a couple of specialized functions which are not directly available for single reads-tracks but requires the context of the overall tracks-framework. However, unless any specific functionality of the stand-alone read-mapping is required, we recommend to use the tracks output for the additional flexibility in further analysis. Later it is possible to convert to and from tracks (see section 24.4). The side-panel functionality for viewing a read track and an annotation track is shown in Figure 24.8.

**Stand-alone read mapping** This output is more elaborate than the reads track and includes the full reference sequence (including annotations) and 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 18.7). The down-side of a stand-alone read mapping is that it copies all the information from the reference sequence which can take up a lot of disk space, and second that it does not lend itself to comparative analyses. If you wish to compare e.g. SNPs from one sample to another sample, or against a database of variants, this calls for using a reads-track instead. Note that if multiple reference sequences are used as input, a read mapping table is created (see section 25.5.1).

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 25.3.2.
- **Collect un-mapped 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, and if you wish to see a log of the process (see section 8.2).

Clicking **Finish** will start the mapping.

## 25.3 Mapping reports

You can create two kinds of reports regarding read mappings and de novo assemblies: *First*, you can choose to generate a summary report about the mapping process itself (see section 25.2). *Second*, you can generate a detailed statistics report after the mapping or assembly has finished. This report is useful if you want to generate statistics across results made in different processes, and it generates more detailed statistics than the summary mapping report. Both reports are described below.

See section section 29.1.12 for more information about de novo assembly reports.

### 25.3.1 Detailed mapping report

To create a detailed mapping report:

**Toolbox** | **NGS Core Tools (**  **)** | **Create Detailed Mapping Report (**  **)**

This opens a dialog where you can select mapping results ( )/ ( )/ ( ) or RNA-Seq analysis results ( ).

Clicking **Next** will display the dialog shown in figure 25.10

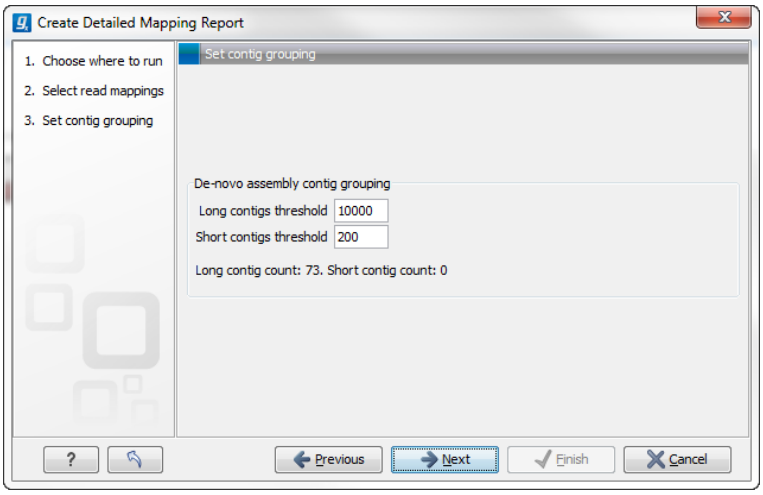



Figure 25.10: *Parameters for mapping reports.*

The first option is to set thresholds for grouping long and short contigs. The grouping is used to show statistics like number of contigs, mean length etc. for the contigs in each group. Thresholds can only be specified for de novo assemblies that does not have a consensus sequence. Whenever a consensus sequence is present the "De novo assembly contig grouping" options are disabled. 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).

Click **Next** to select output options as shown in figure 25.11

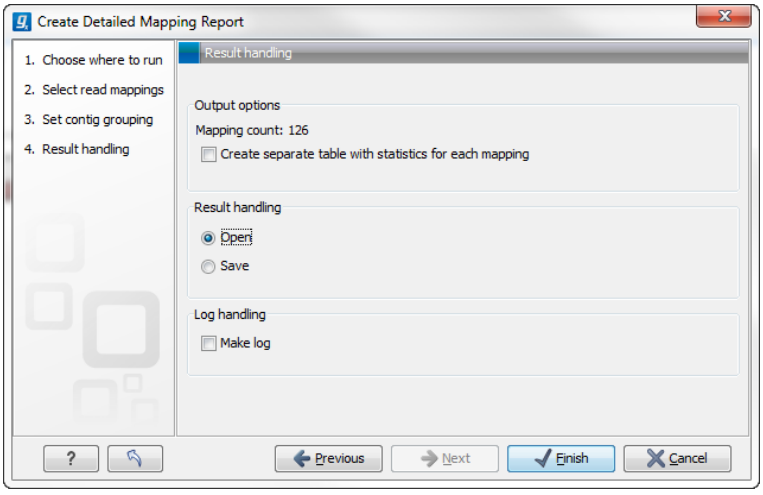



Figure 25.11: *Optionally create a table with detailed statistics per reference.*

Per default, an overall report will be created as described below. In addition, by checking **Create table with statistics for each mapping**, you can create a table showing detailed statistics for each reference sequence (for de novo results the contigs act as reference sequences, so it will be one row per contig). The following sections describe the information produced.

### Reference sequence statistics

For reports on results of read mapping, section two concerns the reference sequences. The reference identity part includes the following information:

**Reference name** The name of the reference sequence.

**Reference Latin name** The reference sequence's Latin name.

**Reference description** Description of the reference.

If you want to inspect and edit this information, right-click the reference sequence in the contig and choose **Open Sequence** and switch to the **Element info** ( ) tab (learn more in section 10.4). Note that you need to create a new report if you want the information in the report to be updated. If you update the information for the reference sequence within the contig, you should know that it doesn't affect the original reference sequence saved in the **Navigation Area**.

The next part of the report reports coverage statistics including GC content of the reference sequence. Note that 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).

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. In the example shown in figure 25.12, 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".

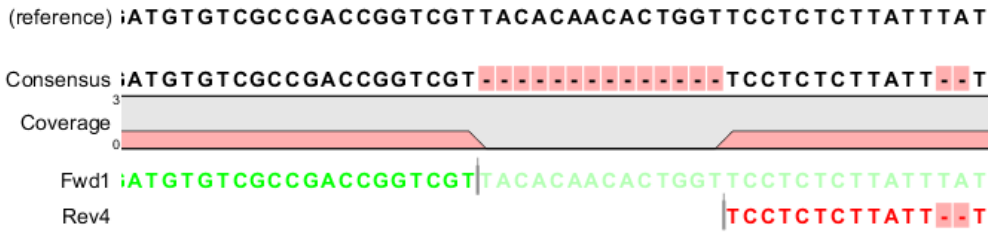



Figure 25.12: *A region of zero coverage in the middle and one time coverage on each side. Note that the gaps to the very right are within the same read which means that these two positions on the reference sequence are still counted as "covered".*

The identity section is followed by some statistics on the zero-coverage regions; the number, minimum and maximum length, mean length, standard deviation, total length and a list of the regions. If there are too many regions, they will not all be listed in the report (if there are more than 20, only the first 10 are reported).

Next follow two bar plots showing the distribution of coverage with coverage level on the x-axis and number of positions with that coverage on the y-axis. An example is shown in figure 25.15.

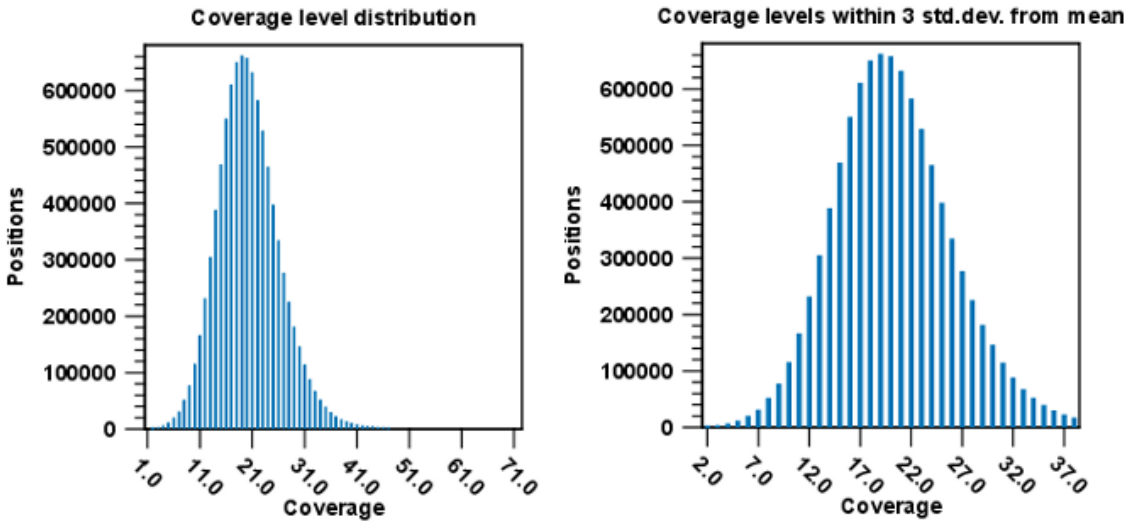



Figure 25.13: *Distribution of coverage - to the left for all the coverage levels, and to the right for coverage levels within 3 standard deviations from the mean.*

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. Note that zero-coverage regions are not shown in the graph but reported in text below (this information is also in the zero-coverage section). Below the second coverage graph there are some statistics on the data that is outside the 3 standard deviations.

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 a graph plotting coverage against GC content (see figure 25.14). Note that you can see the GC content for each reference sequence in the table above.

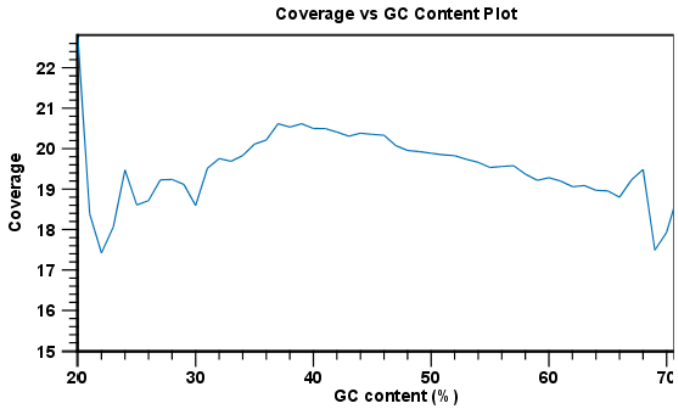



Figure 25.14: *The plot displays, for each GC content level (0-100 %), the mean read coverage of 100bp reference segments with that GC content.*

The plot displays, for each GC content level (0-100 %), the mean read coverage of 100bp reference segments with that GC content.

At the end follows statistics about the reads which are the same for both reference and de novo assembly (see section 25.2 below).

#### Contig statistics for de novo assembly

After the summary there is a section about the contig lengths. For each set of contigs, you can see the number of contigs, minimum, maximum and mean lengths, standard deviation and total contig length (sum of the lengths of all contigs in the set). The contig sets are:

**N25 contigs** 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.

**N50** This measure is similar to N25 - just with 50 % instead of 25 %. This is probably the most well-known measure of de novo assembly quality - it is a more informative way of measuring the lengths of contigs.

**N75** Similar to the ones above, just with 75 %.

**All contigs** All contigs that were selected.

**Long contigs** This contig set is based on the threshold set in the dialog in figure 25.10.

**Short contigs** This contig set is based on the threshold set in the dialog in figure 25.10. Note that the de novo assembly in the *CLC Genomics Workbench* per default only reports contigs longer than 200 bp.

Next follow two bar plots showing the distribution of coverage with coverage level on the x-axis and number of positions with that coverage on the y-axis. An example is shown in figure 25.15.

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. At the end follows statistics about the reads which are the same for both reference and de novo assembly (see section 25.2 below).

#### 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 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:

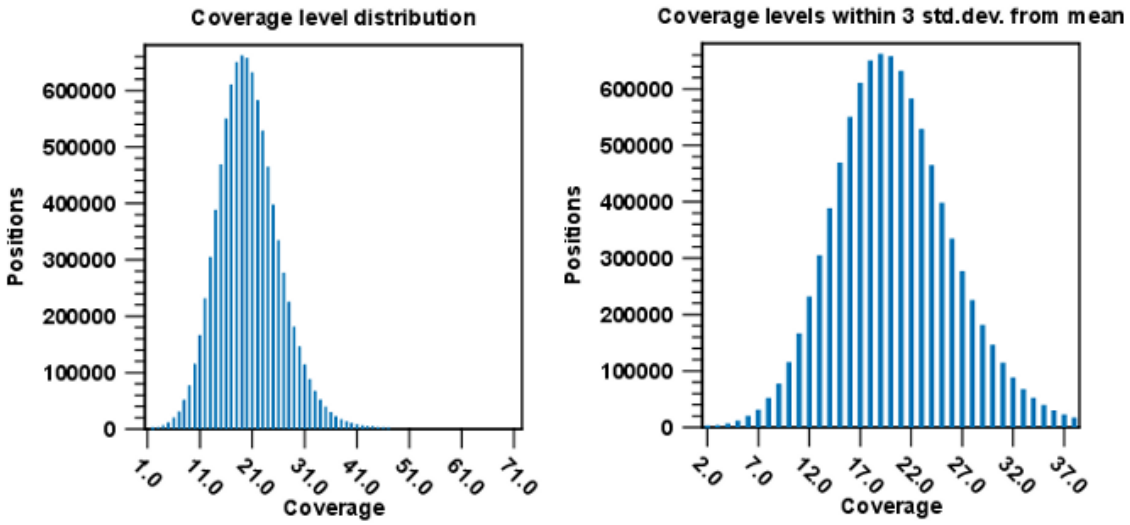



Figure 25.15: *Distribution of coverage - to the left for all the coverage levels, and to the right for coverage levels within 3 standard deviations from the mean.*

**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.

Below these tables follow two graphs showing distribution of paired distances (see figure 25.16) and distribution of read lengths. Note that the distance includes both the read sequence and the insert between them as explained in section 6.3.8.

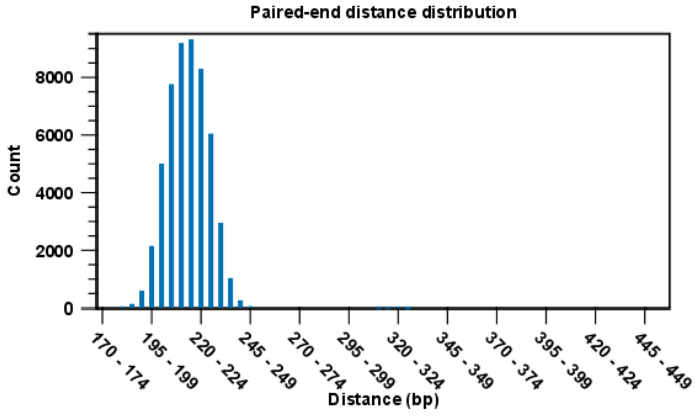



Figure 25.16: *A bar plot showing the distribution of distances between intact pairs.*

Two plots of the distribution of insertion and deletion lengths can bee seen in figure 25.17 and figure 25.18.

### Quality and mismatches

Next follows a detailed description of 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 25.17.

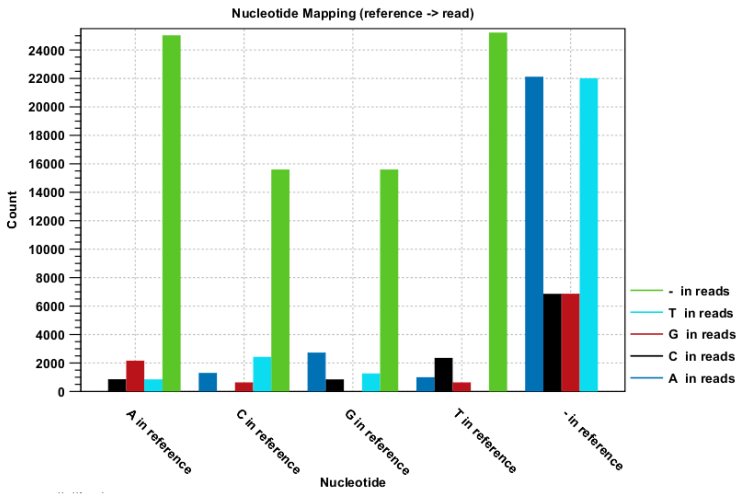



Figure 25.17: *The As and Ts are more often substituted with a gap in the sequencing reads than C and G.*

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 mismatches, respectively. 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 25.18 shows.




Figure 25.18: *There are mismatches towards the end of the reads.*

The last plots shows the unaligned read lengths.

### 25.3.2 Summary mapping report

If you choose to create a report as part of the read mapping (see section 25.2), this report will summarize the results of the mapping process. An example of a report is shown in figure 25.19

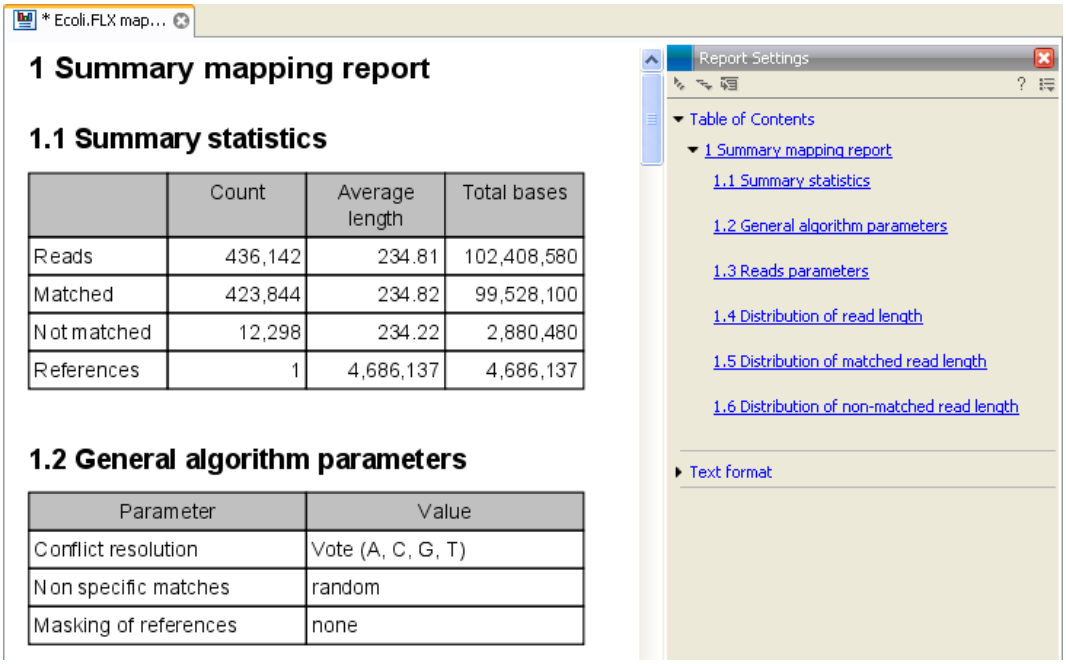



Figure 25.19: *The summary mapping report.*

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.
- **Parameters**. The settings used are reported for the process as a whole and for each sequence list used as input.
- **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 you have 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.

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.

## 25.4 Color space

### 25.4.1 Sequencing

The SOLiD sequencing technology from Applied Biosystems is different from other sequencing technologies since it does not sequence one base at a time. Instead, two bases are sequenced at a time in an overlapping pattern. There are 16 different dinucleotides, but in the SOLiD technology, the dinucleotides are grouped in four carefully chosen sets, each containing four dinucleotides. The colors are as follows:

| Base 1 | Base 2 | | | |
|---|---|---|---|---|
| | A | C | G | T |
| A | ● | ● | ● | ● |
| C | ● | ● | ● | ● |
| G | ● | ● | ● | ● |
| T | ● | ● | ● | ● |

Notice how a base and a color uniquely defines the following base. This approach can be used to deduce a whole sequence from the initial nucleotide and a series of colors. Here is a sequence and the corresponding colors.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Sequence** | T | A | C | T | C | C | A | T | G | C | A |
| **Colors** | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | |

The colors do not uniquely define the sequence. Here is another sequence with the same list of colors:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Sequence** | A | T | G | A | G | G | T | A | C | G | T |
| **Colors** | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | |

But if the first nucleotide is known, the colors do uniquely define the remaining sequence. This is exactly the strategy used in SOLiD sequencing: The first nucleotide is known from the primer used, and the remaining nucleotides are deduced from the colors.

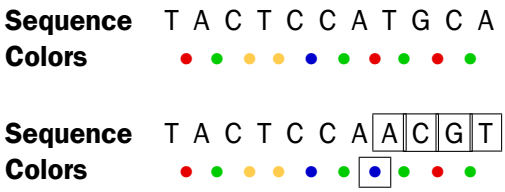
### 25.4.2 Error modes

As with other sequencing technologies, errors do occur with the SOLiD technology. If a single nucleotide is changed, two colors are affected since a single nucleotide is contained in two overlapping dinucleotides:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Sequence** | T | A | C | T | C | C | A | T | G | C | A |
| **Colors** | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Sequence** | T | A | C | T | C | C | A | A | G | C | A |
| **Colors** | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | |

Sometimes, a wrong color is determined at a given position. Due to the dependence between dinucleotides and colors, this affects the remaining sequence from the point of the error:




Thus, when the instrument makes an error while determining a color, the error mode is very different from when a single nucleotide is changed. This ability to differentiate different types of errors and differences is a very powerful aspect of SOLiD sequencing. With other technologies sequencing errors always appear as nucleotide differences.

### 25.4.3 Mapping in color space

Reads from a SOLiD sequencing run may exhibit all the same differences to a reference sequence as reads from other technologies: mismatches, insertions and deletions. On top if this, SOLiD reads may exhibit color errors, where a color is read wrongly and the rest of the read is affected. If such an error is detected, it can be corrected and the rest of the read can be converted to what it would have been without the error.

Consider this SOLiD read:

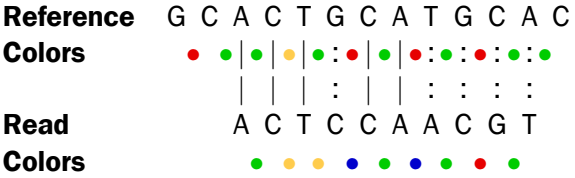



The first nucleotide (T) is from the primer, so this is ignored in the following analysis. Now, assume that a reference sequence is this:




Here, the colors are just inferred since they are not the result of a sequencing experiment.

Looking at the colors, a possible alignment presents itself:




In the beginning of the read, the nucleotides match (ACT), then there is a mismatch (G in reference and C in read), then two more matches (CA), and finally the rest of the read does not match. But, the colors match at the end of the read. So a possible interpretation of the alignment is that there is a nucleotide change in position four of the read and a color space error between positions six and seven in the read. Such an interpretation can be represented as:



Here, the * represents a color error. The remaining part of the displayed read sequence has been adjusted according to the inferred error. So this alignment scores nine times the match score minus the mismatch cost and a color error cost. This color error cost is a new parameter that is introduced when performing read mapping in color space.

Note that a color error may be inferred before the first nucleotide of a read. This is the very first color after the known primer nucleotide that is wrong, changing the whole read.

Here is an example from a set of real SOLiD data that was reference assembled by taking color space into account using ungapped global alignments.

```
444_1840_767_F3 has 1 match with a score of 35:

    1046535 GATACTCAATGCCGCCAAAGATGGAAGCCGGGCCA 1046569    reference
            |||||||||||||||||||||||||||||||||||
            GATACTCAATGCCGCCAAAGATGGAAGCCGGGCCA            reverse read

444_1840_803_F3 has 0 matches

444_1840_980_F3 has 1 match with a score of 29:

    2620828 GCACGAAAACGCCGCGTGGCTGGATGGT*CAAC*GTC 2620862    reference
            |||||||||||||||||||||||||||*||||*|||
            GCACGAAAACGCCGCGTGGCTGGATGGT*CAAC*GTC            read

444_1840_1046_F3 has 1 match with a score of 32:

    3673206 TT*GGTCAGGGTCTGGGCTTAGGCGGTGAATGGGGC 3673240    reference
            ||*|||||||||||||||||||||||||||||||||
            TT*GGTCAGGGTCTGGGCTTAGGCGGTGAATGGGGC            reverse read

444_1841_22_F3 has 0 matches

444_1841_213_F3 has 1 match with a score of 29:

    1593797 CTTTG*AGCGCATTGGTCAGCGTGTAATCTCCTGCA 1593831    reference
            |||||*|||||||| ||||||||||||||||||||
            CTTTG*AGCGCATTAGTCAGCGTGTAATCTCCTGCA            reverse read
```

The first alignment is a perfect match and scores 35 since the reads are all of length 35. The next alignment has two inferred color errors that each count is -3 (marked by * between residues), so the score is *35 - 2 x 3 = 29*. Notice that the read is reported as the inferred sequence taking the color errors into account. The last alignment has one color error and one mismatch giving a score of *34 - 3 - 2 = 29*, since the mismatch cost is 2.

Running the same reference assembly without allowing for color errors, the result is:

```
444_1840_767_F3 has 1 match with a score of 35:

    1046535 GATACTCAATGCCGCCAAAGATGGAAGCCGGGCCA 1046569    reference
            |||||||||||||||||||||||||||||||||||
            GATACTCAATGCCGCCAAAGATGGAAGCCGGGCCA            reverse read
```

```
444_1840_803_F3 has 0 matches

444_1840_980_F3 has 0 matches

444_1840_1046_F3 has 1 match with a score of 29:

    3673206 TTGGTCAGGGTCTGGGCTTAGGCGGTGAATGGGGC 3673240    reference
              |||||||||||||||||||||||||||||||||
            AAGGTCAGGGTCTGGGCTTAGGCGGTGAATGGGGC            reverse read

444_1841_22_F3 has 0 matches

444_1841_213_F3 has 0 matches
```

The first alignment is still a perfect match, whereas two of the other alignment now do not match since they have more than two errors. The last alignment now only scores 29 instead of 32, because two mismatches replaced the one color error above. This shows the power of including the possibility of color errors when aligning: many more matches are found.

The reference assembly program in *CLC Genomics Workbench* does not directly support alignment in color space only, but if such an alignment was carried out, sequence 444_1841_213_F3 would have three errors, since a nucleotide mismatch leads to two color space differences. The alignment would look like this:

```
444_1841_213_F3 has 1 match with a score of 26:

    1593797 CTTTG*AGCGCATT*G*GTCAGCGTGTAATCTCCTGCA 1593831    reference
            |||||*|||||||*|*|||||||||||||||||||||
            CTTTG*AGCGCATT*G*GTCAGCGTGTAATCTCCTGCA           reverse read
```

So, the optimal solution is to both allow nucleotide mismatches and color errors in the same program when dealing with color space data. This is the approach taken by the assembly program in *CLC Genomics Workbench*.

**Note!** If you set the color error cost as low as 1 while keeping the mismatch cost at 2 or above, a mismatch will instead be represented as two adjacent color errors.

### 25.4.4 Viewing color space information

Importing data from SOLiD systems (see section 6.3.3) will from *CLC Genomics Workbench* version 3.1 be imported as color space. This means that if you open the imported data, it will look like figure 25.20

In the **Side Panel** under **Nucleotide info**, you find the **Color space encoding** group which lets you define a few settings for how the colors should appear. These settings are also found in the side panel of mapping results and single sequences.

**Infer encoding** This is used if you want to display the colors for non-color space sequence (e.g. a reference sequence). The colors are then simply inferred from the sequence.

**Show corrections** This is only relevant for mapping results - it will show where the mapping process has detected color errors. An example of a color error is shown in figure 25.21.

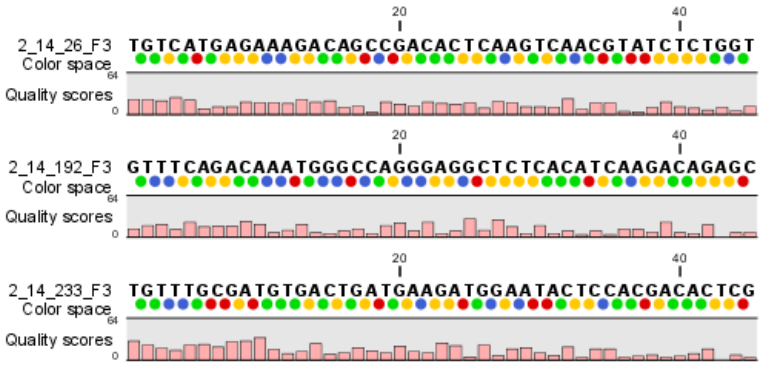



Figure 25.20: *Color space sequence list.*

**Hide unaligned ends** This option determines whether color for the unaligned ends of reads should be displayed. It also controls whether colors should be shown for gaps. The idea behind this is that these color dots will interfere with the color alignment, so it is possible to turn them off.

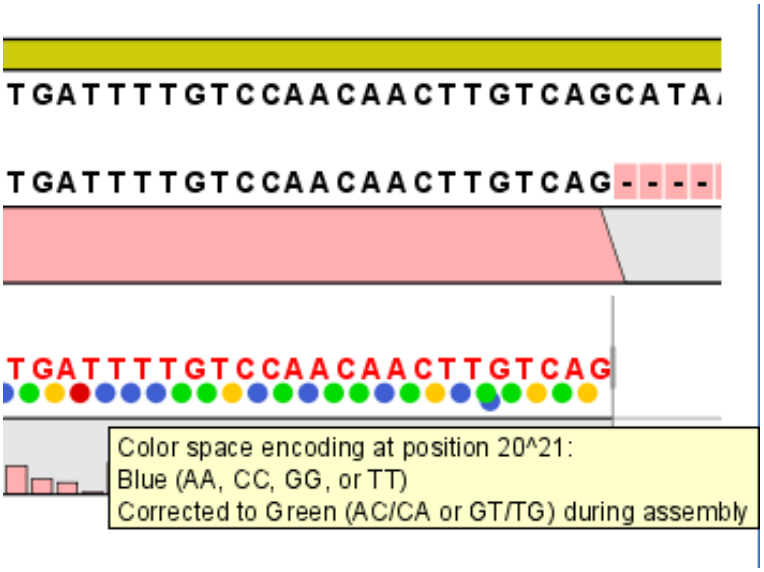



Figure 25.21: *One of the dots have both a blue and a green color. This is because this color has been corrected during mapping. Putting the mouse on the dot displays the small explanatory message.*

## 25.5 Mapping result

Reads can be mapped to linear and circular chromosomes. Read mappings to circular genomes are visualized linearly as shown in figure 25.22.

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.

Mapping results can either be tracks ( ) (see chapter 24) or mapping tables ( ) or single mappings ( ). This section explains more about the latter two.

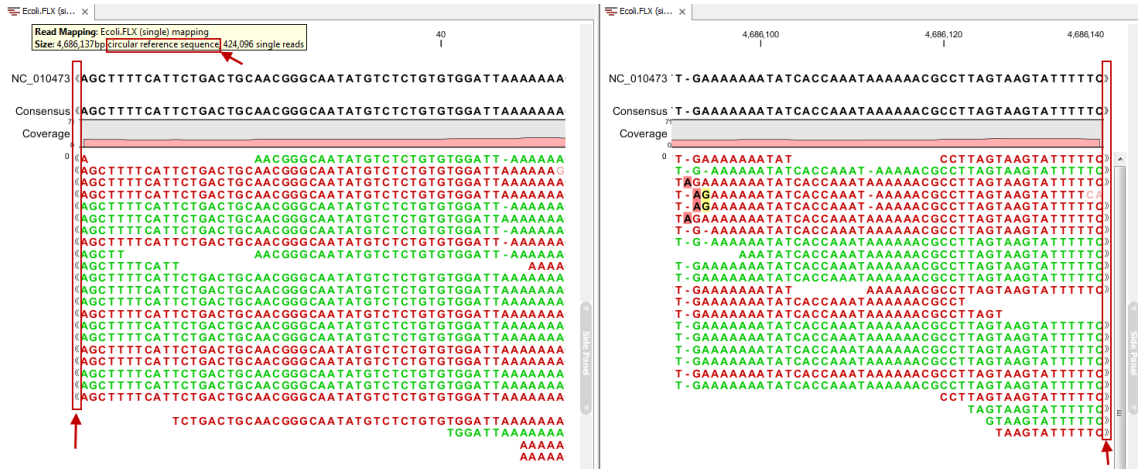


Figure 25.22: *Mapping reads to a circular chromosome. Reads that are marked with double arrows at the ends are reads that map across the starting point of the sequence. The arrows indicate that the alignment continues at the other end of the reference sequence.*

**Note:** 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.

### 25.5.1 Mapping table

When several reference sequences are used or you are performing de novo assembly with the reads mapped back to the contig sequences, all your mapping data will be accessible from a table ( ). It means that all the individual mappings are treated as one single file to be saved in the **Navigation Area** as a table.

An example of a mapping table for a *de novo* assembly is shown in figure 25.23.

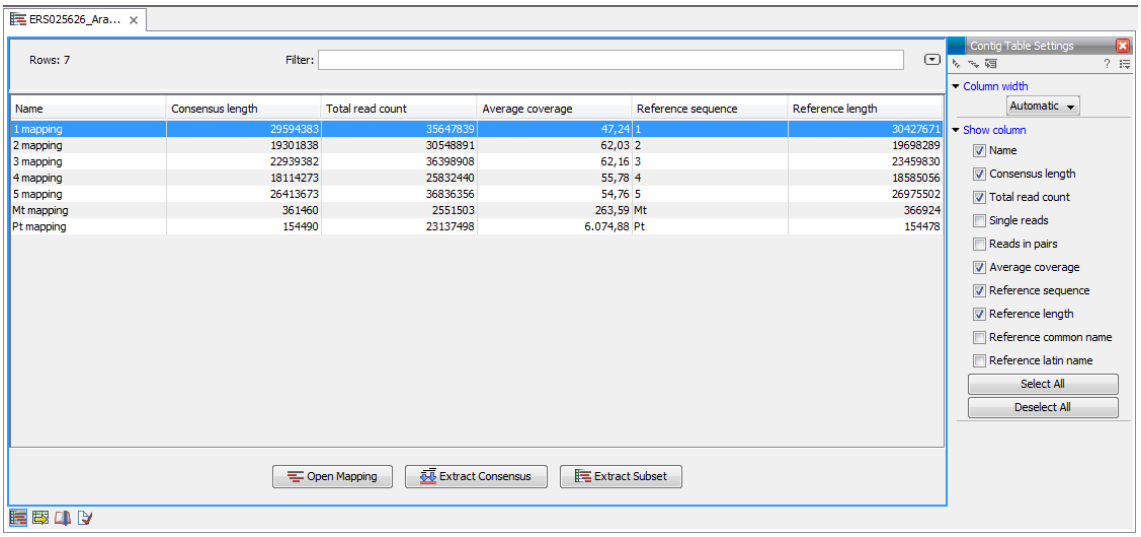



Figure 25.23: *The mapping table.*

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**
- **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.

An example of a contig table produced by mapping reads to a reference is shown in figure 25.24. The read mappings use information from the reference sequences that were used as input.

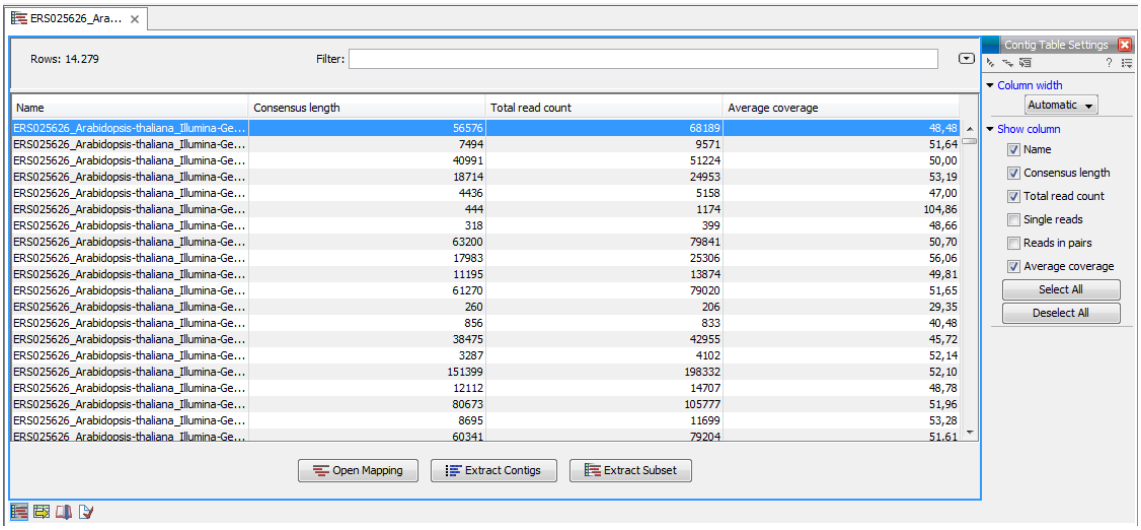



Figure 25.24: *The contig table.*

In addition to the information found in the *de novo* table, the mapping table also provides information about 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 (press Ctrl + A - ⌘ +A on Mac - to select all) 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 25.8).

- **Extract Subset**. Creates a new mapping table with the mappings that you have selected.

You can copy the textual information from the table by selecting in the table and click **Copy** (). This can then be pasted into e.g. Excel. You can also export the table in Excel format.

### 25.5.2 View settings in the Side Panel

When you open a single mapping, the following settings are 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 section 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 18.1.2 and section 10.1.
    - **Low.** Hides trace data, quality scores and puts the reads' annotations on the sequence.
    - **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 25.25.
  - **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.

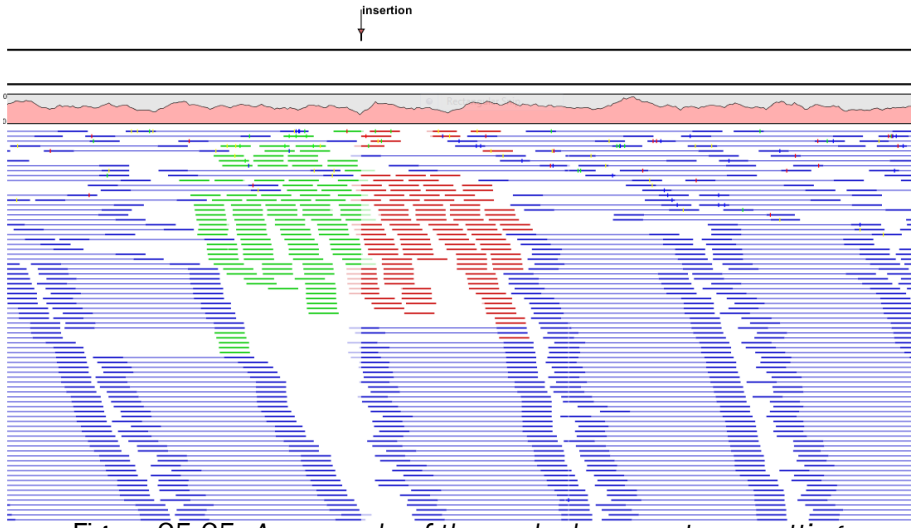



Figure 25.25: *An example of the packed compactness setting.*

- **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 horizontal lines in. 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). E.g. 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.

- **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 6.6).
            - **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.** At the top of the **Side Panel**:

    - **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 mapping, which can be adjusted in the View settings. Those covered here were the key ones relevant for standard review of mapping results.

### 25.5.3 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 callers 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 25.26.

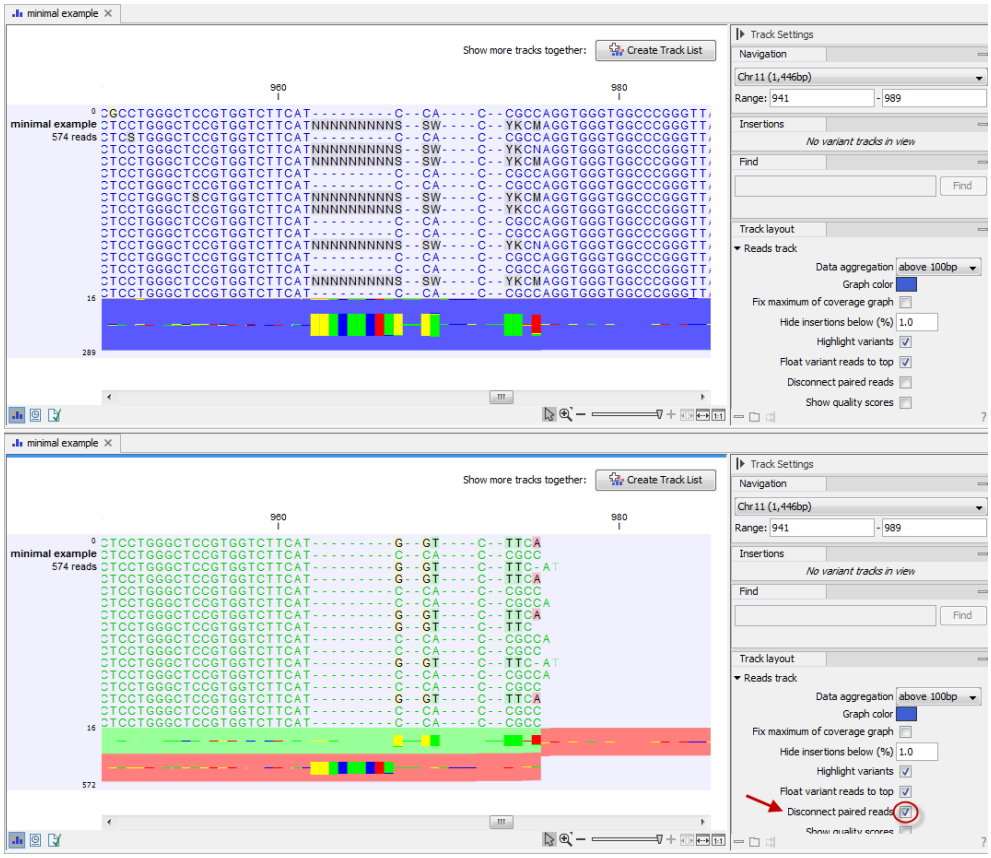



Figure 25.26: *Overlapping paired reads containing discrepancies in the overlapping region. The divergent area in the overlapping read pair is marked with N's and IUPAC codes in the top part of the figure. Disconnecting the paired reads by ticking the box "Disconnect Paired Reads" in the Side Panel allows you to see the individual reads.*

### 25.5.4 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 25.27 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 6.3.8), or if one of the reads do not map at all.

In some situations it is useful to investigate where the mate of the broken pairs map. This would indicate genomic rearrangements, mis-assemblies of de novo assembly etc. In order to see this, select the region in question on the reference sequence, right-click and choose **Find Broken Pair Mates**.

This will open the dialog shown in figure 25.28.

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.

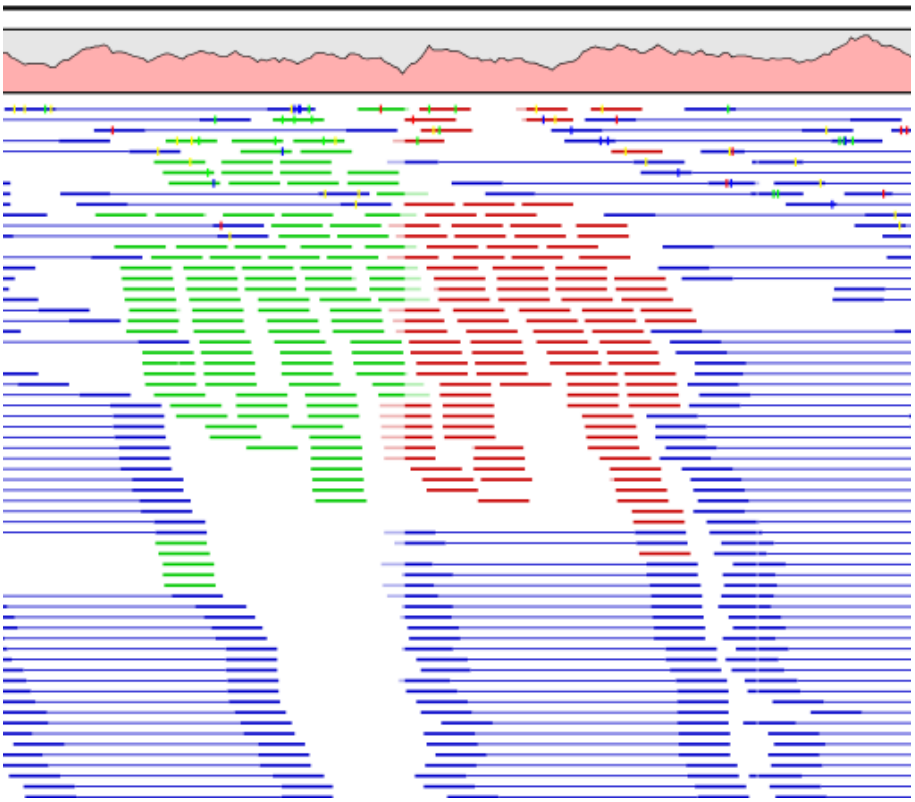


Figure 25.27: *Broken pairs.*

Click **Next** and **Finish**, and you will see an overview table as shown in figure 25.29.

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 25.28.

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.

## 25.6 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.

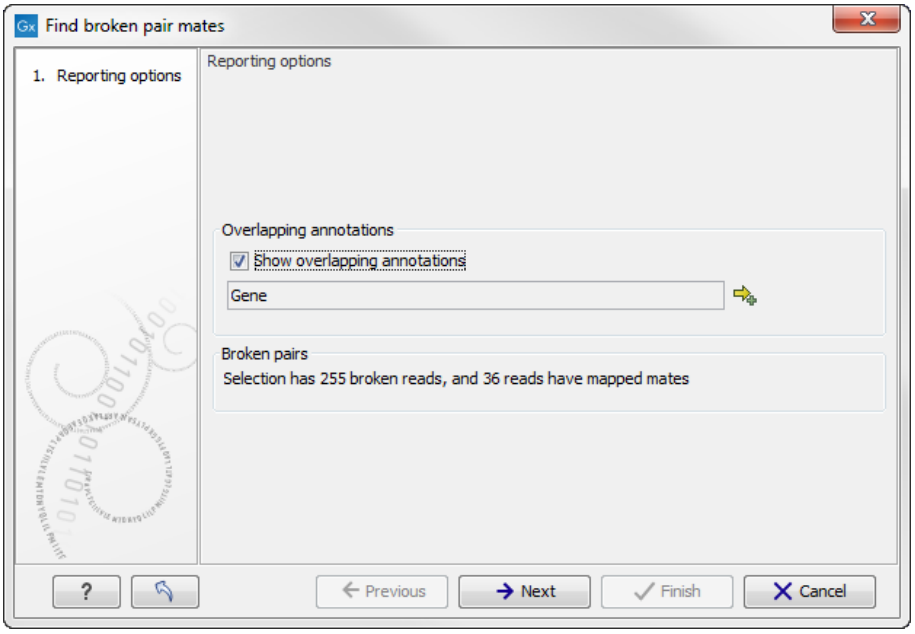



Figure 25.28: *Finding the mates of broken pairs.*

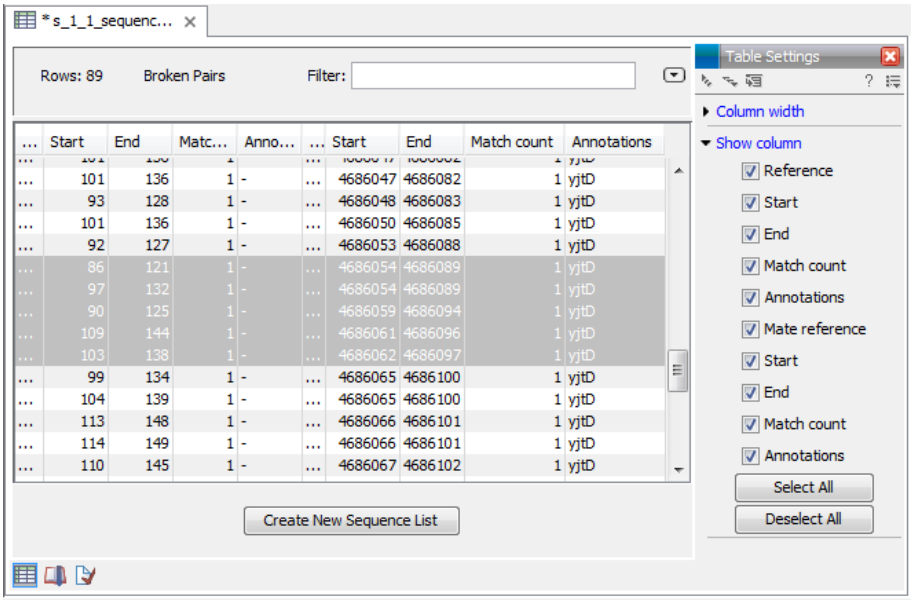



Figure 25.29: *An overview of the broken pairs.*

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 the local alignment algorithm is shown in figure 25.30. An example in the case of a mapping made using the global alignment algorithm is shown in figure 25.31.




Figure 25.30: *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 caller 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.*

### 25.6.1 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 25.32 depicts a partial realignment graph for the read

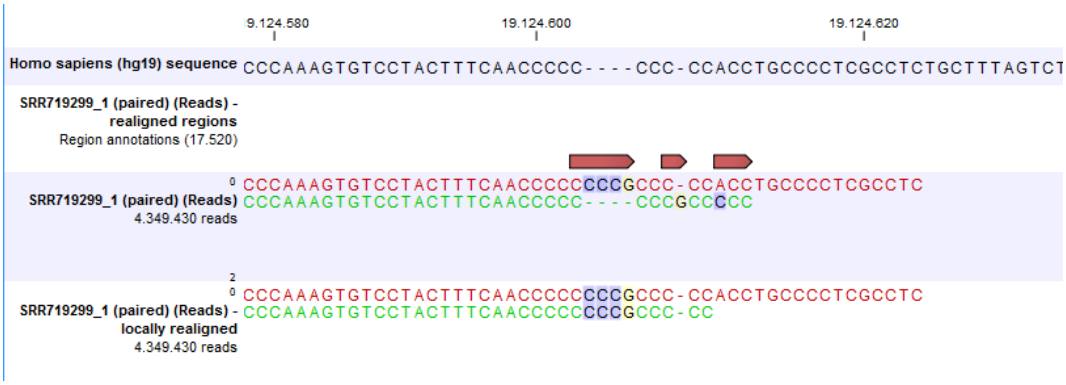



Figure 25.31: *Local realignment of a read mapoping produced with the 'global' option. Before realignment the green read was mapped with two mismatches. After realignment it is mapped with the inserted 'CCCG' sequence (seen in the alignment of the red read) and no mismatches.*

mapping shown in figure 25.30.

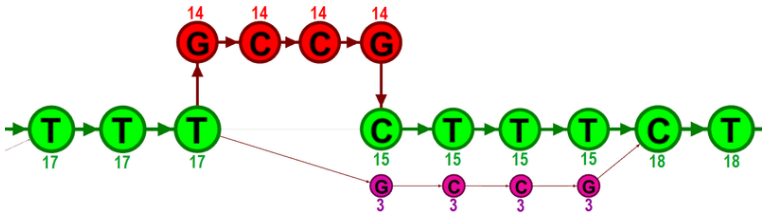



Figure 25.32: *The green nodes represent nucleotides of the reference sequence. The four red nodes represent the four-nucleotide insertion observed in fourteen mapped reads. The four violet nodes represent the four mismatches to the reference sequence observed in three mapped reads. During realignment 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.*

### 25.6.2 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 25.33 depicts such an example.

### 25.6.3 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'

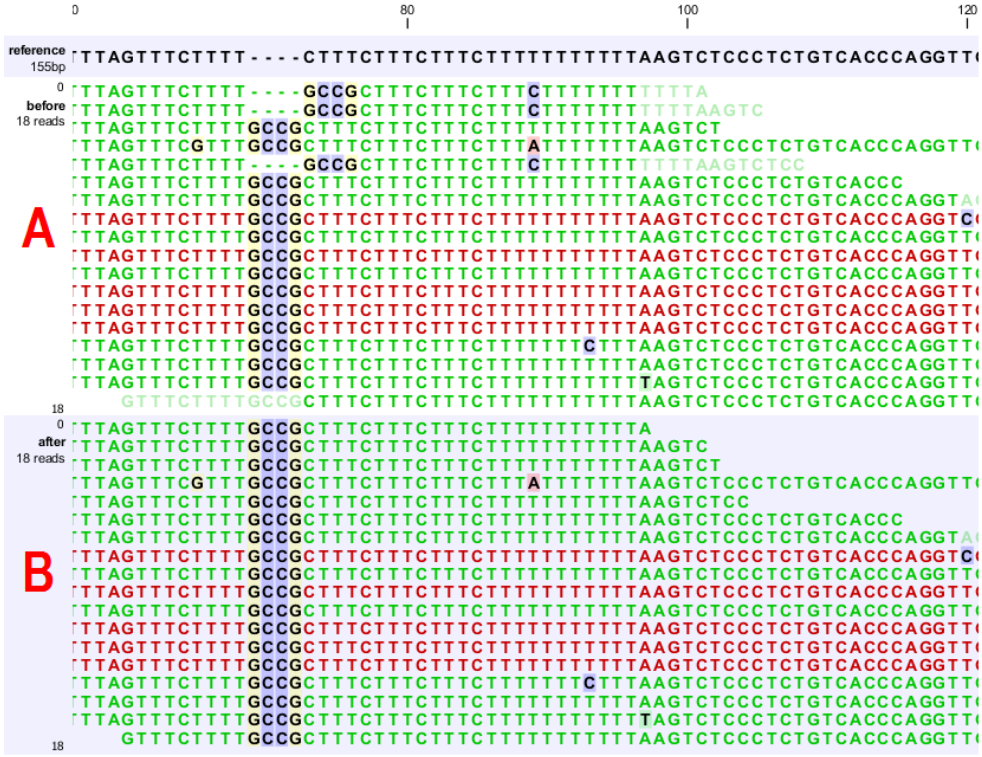



Figure 25.33: *[A] The alignments of the first, second, and fifth read in this read mapping do not support the four-nucleotide insertion supported by the remaining reads. Additionally, the first, second, fifth and the last reads have unaligned ends. [B] After applying local realignment the first, second 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 25.30).*

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 prior evidence that they exist in the data (e.g., the 'InDel' track from the Structural Variants' tool (see Section 27.10) produced on the read mapping that is being aligned).

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 25.34 and figure 25.35 show examples that can be improved by guiding the local realignment algorithm.




Figure 25.34: *[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 variants track shown in (E), and this will result in realignment of all three reads. [E] The guiding variants track contains, amongst others, the four nucleotide insertion.*

### 25.6.4 Multi-pass local realignment

As described in section 25.6.1 the algorithm initially builds the realignment graph using the input read mapping. After the graph has been built the algorithm realigns individual reads based on information inferred from the realignment graph structure and its associated metadata. In some cases repetitive realignment iterations yield even more improvements, because with each realignment iteration the structure of the realignment graph changes slightly, potentially permitting further improvements. Local realignment therefore supports to perform multiple 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.

### 25.6.5 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:




Figure 25.35: *[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).*

- Lengths are longer than 50,000 base pairs.
- The alignment is longer than 50,000 base pairs.
- Crossing the boundaries of circular chromosomes.

Guiding variants are ignored, if:

- They are of type "Replacement".
- They are longer than 100 bp.
- If they are inter-chromosomal structural variations.
- If they contain ambiguous nucleotides.

### 25.6.6 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.

### 25.6.7 How to run the Local Realignment tool

The tool is found in the Toolbox:

**Toolbox | NGS Core Tools ( ) | 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 25.36.

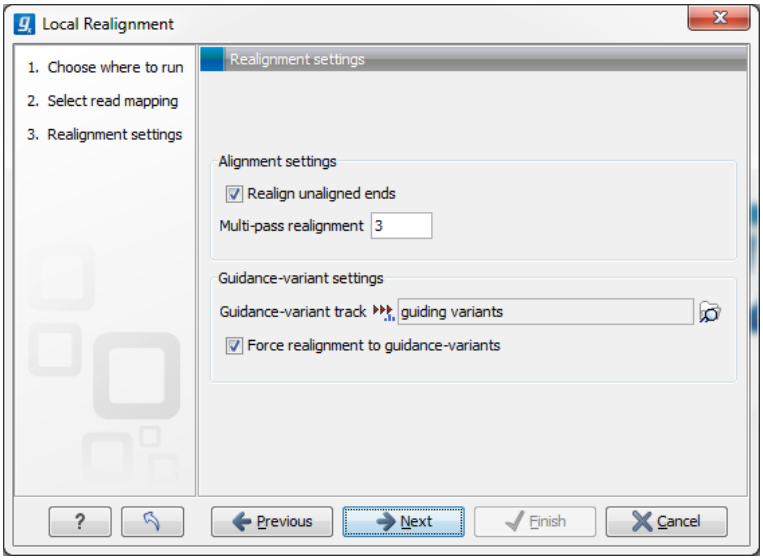



Figure 25.36: *Set the realignment options.*

**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 shall 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 either of the variant callers, The Structural Variant tool or by importing variants from external data sources, such as COSMIC, dbSNP, etc.

  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 27.10), 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 25.37).

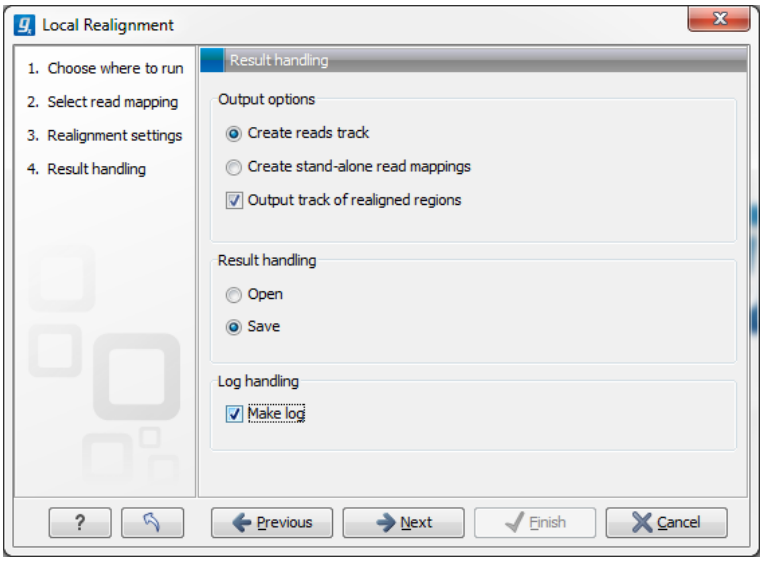



Figure 25.37: *An output track of realigned regions can be created.*

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.

## 25.7 Merge mapping results

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** | **NGS Core Tools** ( ) | **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 25.1).

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**. . 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.

## 25.8 Extract consensus sequence

For all kinds of read mappings, including those generated from *de novo* assembly or RNA-seq analyses, a consensus sequence can be extracted. In addition, you can extract a consensus sequence from a BLAST result as well. The consensus sequence extraction tool can be run in batch and as part of workflows.

To start the tool:

**Toolbox** | **NGS Core Tools** ( ) | **Extract Consensus Sequence** ( )

This opens a dialog where you can select mappings, either in the form of tracks or read mappings, or BLAST results. Click **Next** to specify how the consensus sequence should be created (see figure 25.38).

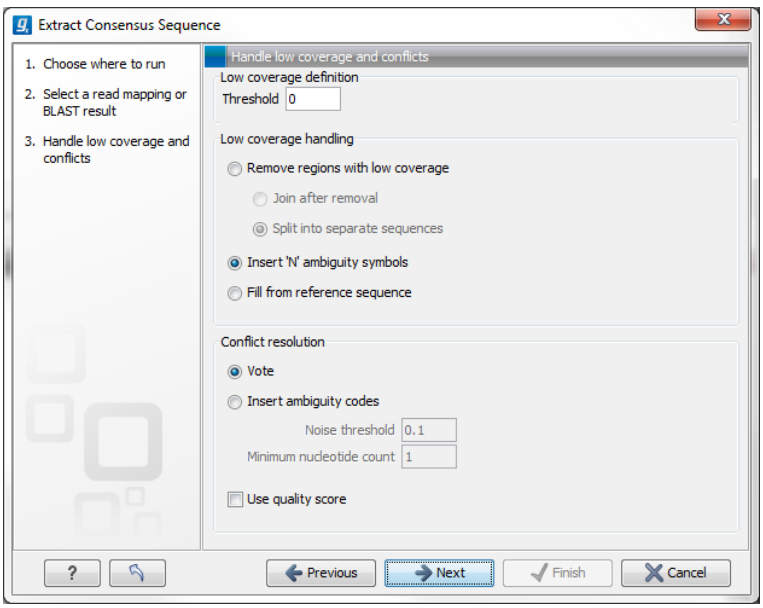



Figure 25.38: *Specifying how the consensus sequence should be extracted.*

It is also possible to extract a consensus sequence from a mapping view by right-clicking the name of the consensus or reference sequence or a selection on the reference sequence and select **Extract Consensus Sequence (  )**.

When extracting a consensus sequence, you can decide how to handle regions with low coverage (a definition of coverage can be found in section 25.2). The first step is to define a **threshold for when coverage is considered low**. The default value is 0, which means that low coverage is defined as no coverage (i.e. no reads align to the reference at this position). That means if you have one read covering a given position, it will only be that read that determines the consensus sequence. If you need more confidence that the consensus sequence is correct, we advise raising this value. Setting a higher low coverage threshold will require more mapped reads to construct the consensus sequence.

A consensus based on mapped reads cannot be generated in regions that meet or are below the value set for the low coverage threshold, there are several options for handling these low coverage regions:

- **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 sequence 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 will simply add 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 will use 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).

In addition to deciding how to handle low coverage regions, you can also decide how to handle conflicts or disagreement between the reads when building a consensus sequence in regions above the low coverage threshold:

- **Vote**. Whenever the reads disagree on the base at a given position, the vote resolution will let the majority of the reads decide which base is correct. In addition, you can specify to let the voting use the base calling **quality scores** from the reads. This is done be simply adding all quality scores for each base and let the sum determine which one is correct. The base with the highest total quality scores will be chosen. If there are two bases that end up summing to the same total quality score for all reads at that location, A is preferred before C, C before G, and G before T. An annotation with the complete information that was used to resolve the conflict will be added.
- **Insert ambiguity codes**. When this options is selected, read conflicts are addressed by using an ambiguity code representing all read bases represented at the reference location. The problem with the voting option is that it will not be able to represent true biological heterozygous variation in the data. For 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. With the option to insert ambiguity codes, this can be solved. (The IUPAC ambiguity codes used can be found in Appendix I and H.) However, if an ambiguity code would always be inserted if just one read had a different base, there would be an ambiguity code whenever there was a sequencing error. In high-coverage NGS data that would be a big problem, because sequencing errors would be abundant. To solve this problem, you can specify a **Noise threshold**. The default value for this is 0.1 which means that for a base to contribute to the ambiguity code, it must be in at least 10 % of the reads at a given position. The **Minimum nucleotide count** specifies the minimum number of reads that are required before a nucleotide is included. Nucleotides below this limit are considered noise.

- **Use quality score**. The "Use quality score" checkbox option is available for conflicts regardless of whether "Vote" or "Insert ambiguity codes" has been selected. The "Use quality score" checkbox option allows you to use the base calling **quality scores** from the reads. This is done by simply adding all the quality scores for each base and let the sum determine which bases to consider. In other words, if quality scores are used, we will sum the quality score (instead of amount of reads) for each base on each position before applying the noise filters and finally call the consensus symbol.

Click **Next** to set the output option as shown in figure 25.39).

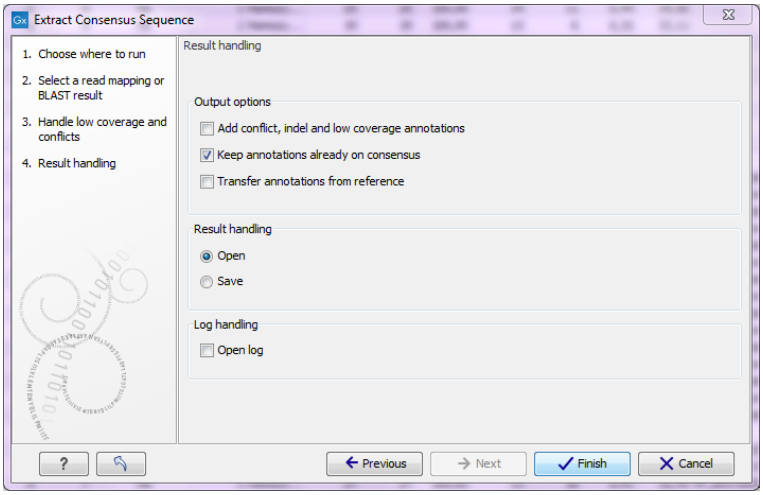



Figure 25.39: *Choose to add annotations to the consensus sequence.*

The annotations that can be added to the consensus sequence produced by this tool, show both conflicts that have been resolved and low coverage regions (unless you have chosen to split the consensus sequence). Please note that for large data sets, this can amount to a very high number of annotations, which will cause the tool to take longer to complete, and the result will take up much more disk space.

It is also possible to transfer existing annotations to the consensus sequence produced. Please note that since the consensus sequence produced may be broken up, the annotations will also be broken up, and you cannot expect them to 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.

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.

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

$$q = \begin{cases} 64 & \text{if } Q > 64 \\ 0 & \text{if } Q < 0 \\ Q & \text{otherwise} \end{cases}$$

## 25.9 Coverage analysis

The coverage analysis tool is designed to identify regions in read mappings with unexpectedly low or high coverage. Such regions may e.g. be indicative of a deletion or an amplification in the sample relative to the reference. The algorithm fits a Poisson distribution to the observed coverages 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 coverages 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.

The coverage analysis tool may produce either an annotation track or a table, depending on the users choice, and, optionally, a report. The annotation track (or 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.

### 25.9.1 Running the Coverage analysis tool

To run the Coverage analysis tool:

**Toolbox** | **Resequencing Analysis** ( ) | **Coverage Analysis** ( )

This opens the dialog shown in figure 25.40.

Select a reads track or read mapping and click **Next**. This opens the dialog shown in figure 25.41.

Set the p-value and minimum length cutoff.

[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 are used instead, all reads contribute to the called consensus and thus $Y = Z$.

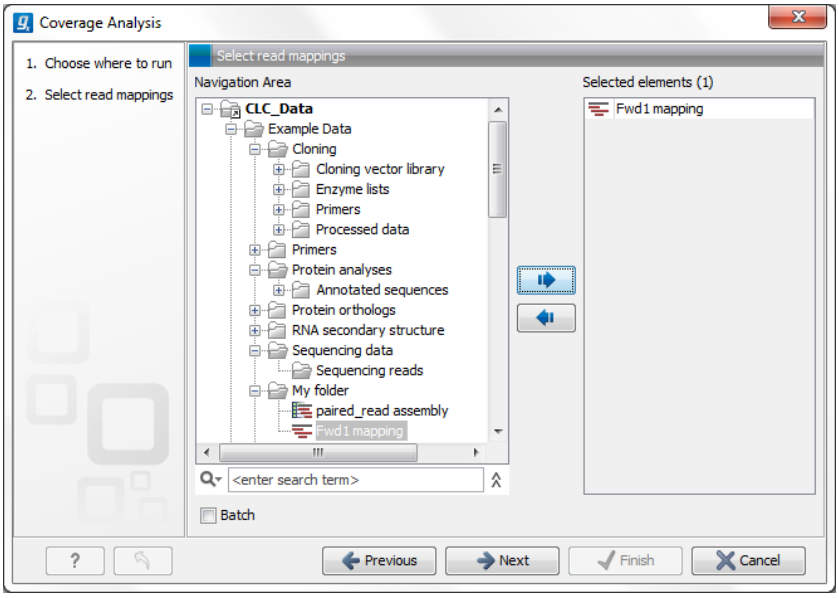



Figure 25.40: *Select read mapping results.*

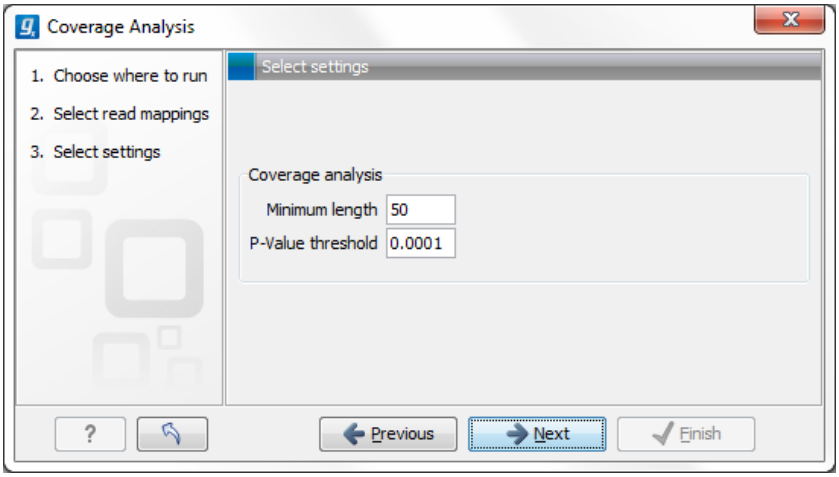



Figure 25.41: *Specify the p-value cutoff.*

Click **Next** and specify the result handling (figure 25.42).

Open or save and click **Finish**.

An example of a track output of the Coverage analysis tool is shown in figure 25.43.

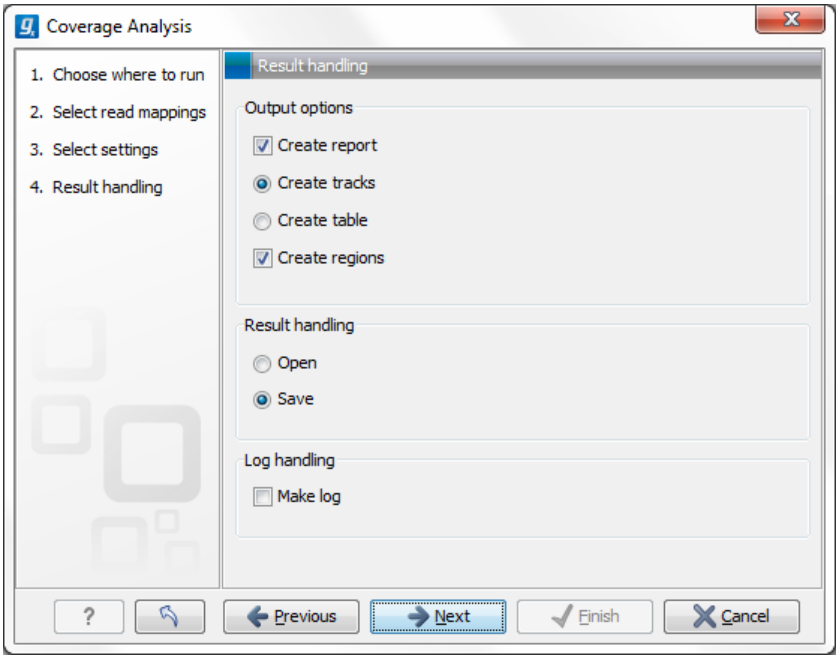



Figure 25.42: *Specify the output.*

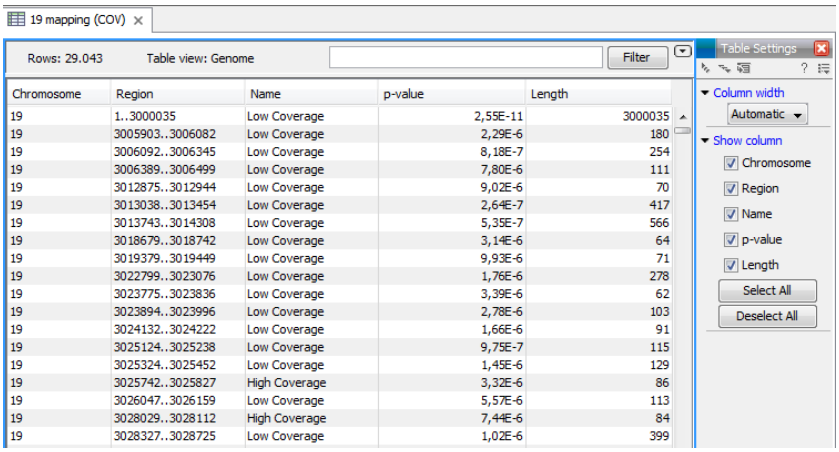



Figure 25.43: *An example of a track output of the Coverage analysis tool.*

# Chapter 26

# Sample Reads

## 26.1 What is Sample Reads?

The **Sample Reads** tool allows a user defined reduction of the number of reads. This can be relevant when editing contigs with a high coverage as an excessive amount of reads can be overly resource demanding and cause the edit options in the read mapping editor to disallow performing actions. The **Sample Reads** tool is very useful for reducing the size of data sets. The sampled reads can sequentially be mapped to contigs in the Contig Aligner. In this way the speed of many edit operations in the contig aligner can be increased significantly without any loss of important information. The preservation of a high coverage is only necessary for de novo assembly. After de novo assembly a large number of reads does not add extra to the analysis.

## 26.2 How to run Sample Reads

**Toolbox** | **NGS Core Tools (**  **)** | **Sample reads (**  **)**

This opens the dialog shown in figure 26.1.

Select the relevant assembled reads and click **Next**. This leads to the **Sample size parameters** step shown in figure 26.2.

In the **Sample size parameters** it is possible to select either an absolute number of reads to sample or a sample percentage, with sample percentage defining the percentage of the sub-sample paired reads (for single end reads) and pairs of reads (for paired reads).

After the **Result handling** step, click **Finish**.

This generates a sampled version of the reads that can be used in a new read mapping or used to replace existing reads in the Contig Aligner.

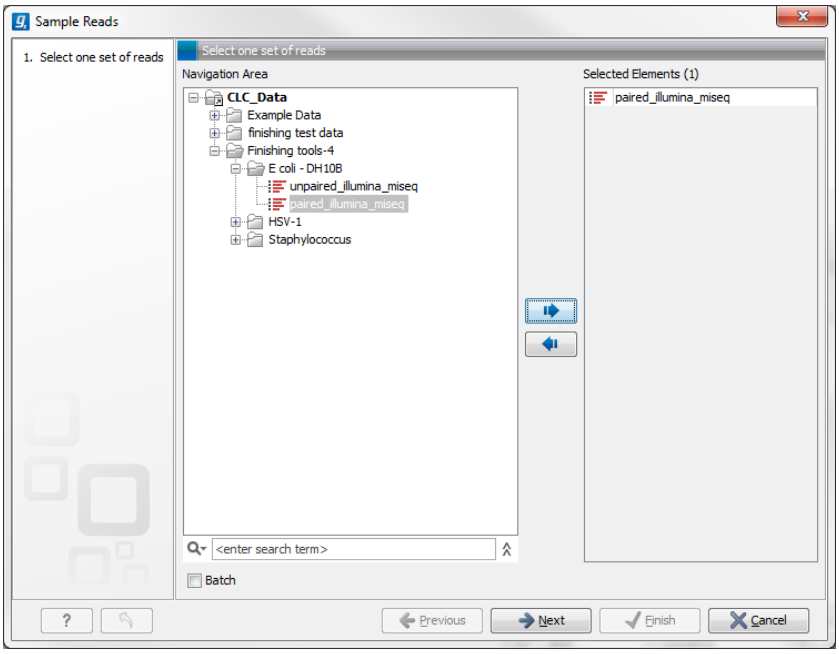



Figure 26.1: *Select the reads.*

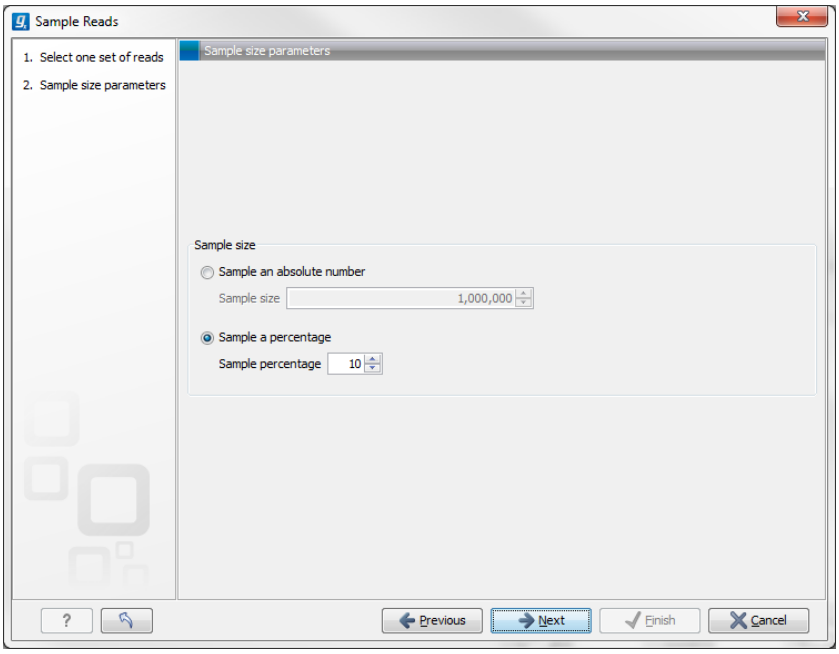



Figure 26.2: *Specify the sample size parameters.*

# Chapter 27

# Resequencing

## Contents


**27.1 Create Statistics for Target Regions . . . . . 620**
27.1.1 Running the Create Statistics for Target Regions . . . . . 620
27.1.2 Coverage summary report . . . . . 621
27.1.3 Per-region statistics . . . . . 624
27.1.4 Coverage table . . . . . 627
27.1.5 Coverage graph . . . . . 627
**27.2 Variant Detectors - overview . . . . . 628**
27.2.1 Differences among the variants called by the three variant callers . . . . 629
27.2.2 How the variant detectors work . . . . . 631
**27.3 Basic Variant Detection . . . . . 633**
**27.4 Fixed Ploidy Variant Detection . . . . . 633**
27.4.1 Ploidy and sensitivity . . . . . 634
**27.5 Low Frequency Variant Detection . . . . . 635**
**27.6 Variant Detectors - error model estimation . . . . . 636**
**27.7 Variant Detectors - filters . . . . . 637**
27.7.1 General filters . . . . . 637
27.7.2 Noise filters . . . . . 640
**27.8 Variant Detectors - the outputs . . . . . 644**
27.8.1 The variant track output . . . . . 644
27.8.2 The annotated table output . . . . . 649
27.8.3 The report . . . . . 649
**27.9 The Fixed Ploidy and Low Frequency variant callers: detailed descriptions . . 649**
27.9.1 The Fixed Ploidy Variant caller: Models and methods . . . . . 650
27.9.2 The Low Frequency Variant caller: Models and methods . . . . . 655
**27.10 InDels and Structural Variants . . . . . 658**
27.10.1 How to run the InDels and Structural Variants tool . . . . . 658
27.10.2 The Structural Variants and InDels output . . . . . 661
27.10.3 The InDels and Structural Variants detection algorithm . . . . . 665
27.10.4 The InDels and Structural Variants detection algorithm - Step 1: Creating Left- and Right breakpoint signatures . . . . . 665

27.10.5 The InDels and Structural Variants detection algorithm - Step 2: Creating Structural variant signatures . . . 666
27.10.6 Theoretically expected structural variant signatures . . . 667
27.10.7 How sequence complexity is calculated . . . 668
**27.11 Variant data . . . 672**
27.11.1 Variant tracks . . . 672
27.11.2 The annotated variant table . . . 674
27.11.3 Variant types . . . 676
27.11.4 Special notes upgrading to Genomics Workbench 7.5 . . . 676
**27.12 Detailed information about overlapping paired reads . . . 677**
**27.13 Annotate and filter variants . . . 678**
27.13.1 Filter against known variants . . . 679
27.13.2 Annotating from known variants . . . 681
27.13.3 Annotate with exon numbers . . . 681
27.13.4 Annotate with flanking sequence . . . 681
27.13.5 Filter marginal variant calls . . . 683
27.13.6 Filter reference variants . . . 684
**27.14 Comparing variants . . . 684**
27.14.1 Compare variants within group . . . 684
27.14.2 Compare sample variants . . . 685
27.14.3 Fisher exact test . . . 686
27.14.4 Trio analysis . . . 688
27.14.5 Filter Against Control Reads . . . 690
**27.15 Predicting functional consequences . . . 691**
27.15.1 Amino acid changes . . . 691
27.15.2 Predict splice site effect . . . 696
27.15.3 GO enrichment analysis . . . 696
27.15.4 Conservation score annotation . . . 699
27.15.5 Link Variants to 3D Protein Structure . . . 700
27.15.6 Download 3D Protein Structure Database . . . 710

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 25.

First comes a description of a tool to perform quality check of targeted resequencing approaches, next we describe the three variant callers that come with the *CLC Genomics Workbench* for finding variants, followed by a section describing a coverage analysis tool used to identify fluctuations in coverage. Next, the format of the variants are described, and finally we go through the various tools for filtering, comparing and annotating variants.

## 27.1 Create Statistics for Target Regions

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 e.g. 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 (e.g. 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 as well as 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 28.1.

### 27.1.1 Running the Create Statistics for Target Regions

To create the target regions statistics:

**Toolbox** | **Resequencing** ( ) | **Create Statistics for Target Regions** ( )

This opens a wizard where you can select mapping results ( )/ ( )/ ( ). Clicking **Next** will take you to the wizard shown in figure 27.1.

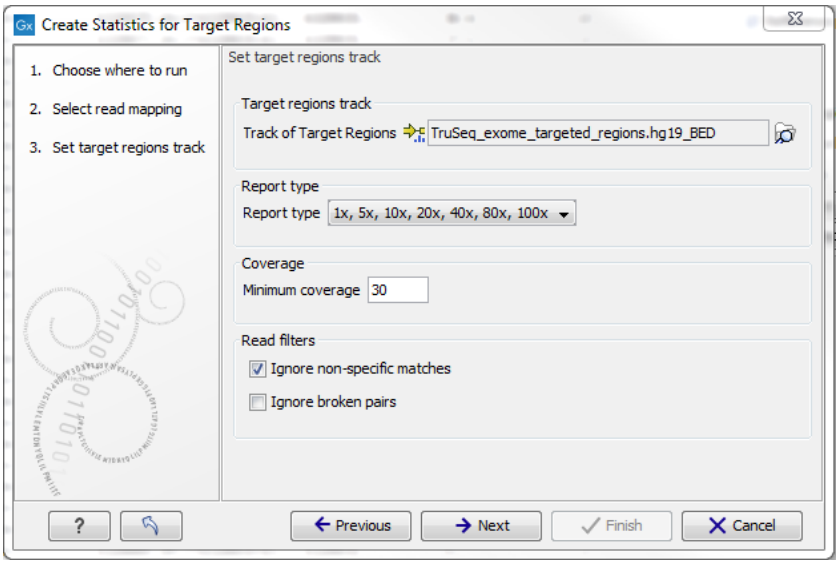



Figure 27.1: *Specifying the track of target regions.*

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 24.4) from annotations on a reference genome that is already stored in the **Navigation Area**.

The **Report type** allows you to select different sets of predefined coverage thresholds to use for reporting (see below). Furthermore, you will be asked to provide a **Minimum coverage** threshold. This will be used to provide the length of each target region that has at least this coverage.

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 27.2).

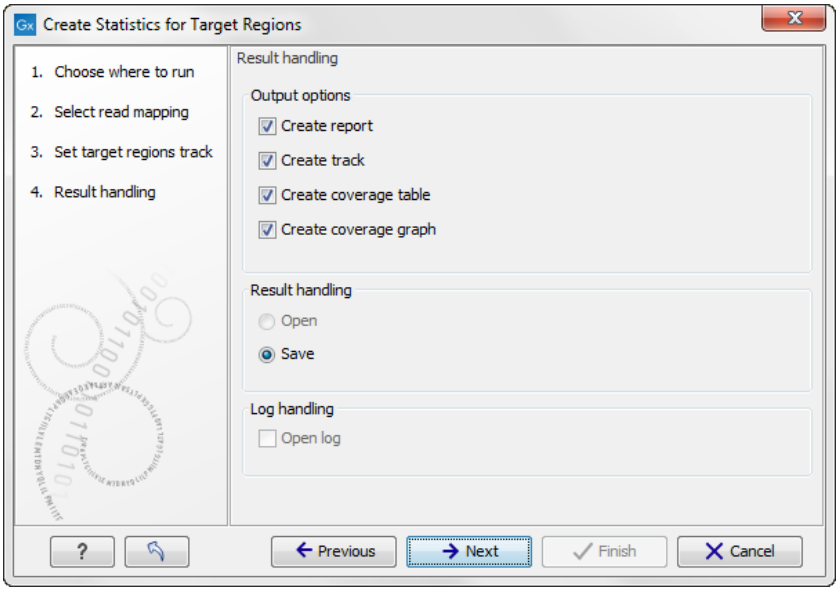



Figure 27.2: *Specifying how the result should be reported.*

There are three options:

- The report gives an overview of the whole data set as explained in section 27.1.2.
- The track gives information on coverage for each target region as described in section 27.1.3.
- The coverage table outputs coverage for each position in all the targets as described in section 27.1.4.
- 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.

### 27.1.2 Coverage summary report

An example of a coverage report is shown in figure 27.3).

This figure shows only the top of the report. The full content is explained below:

**Coverage summary** This table shows overall coverage information.

**Number target regions** The number of targeted regions.

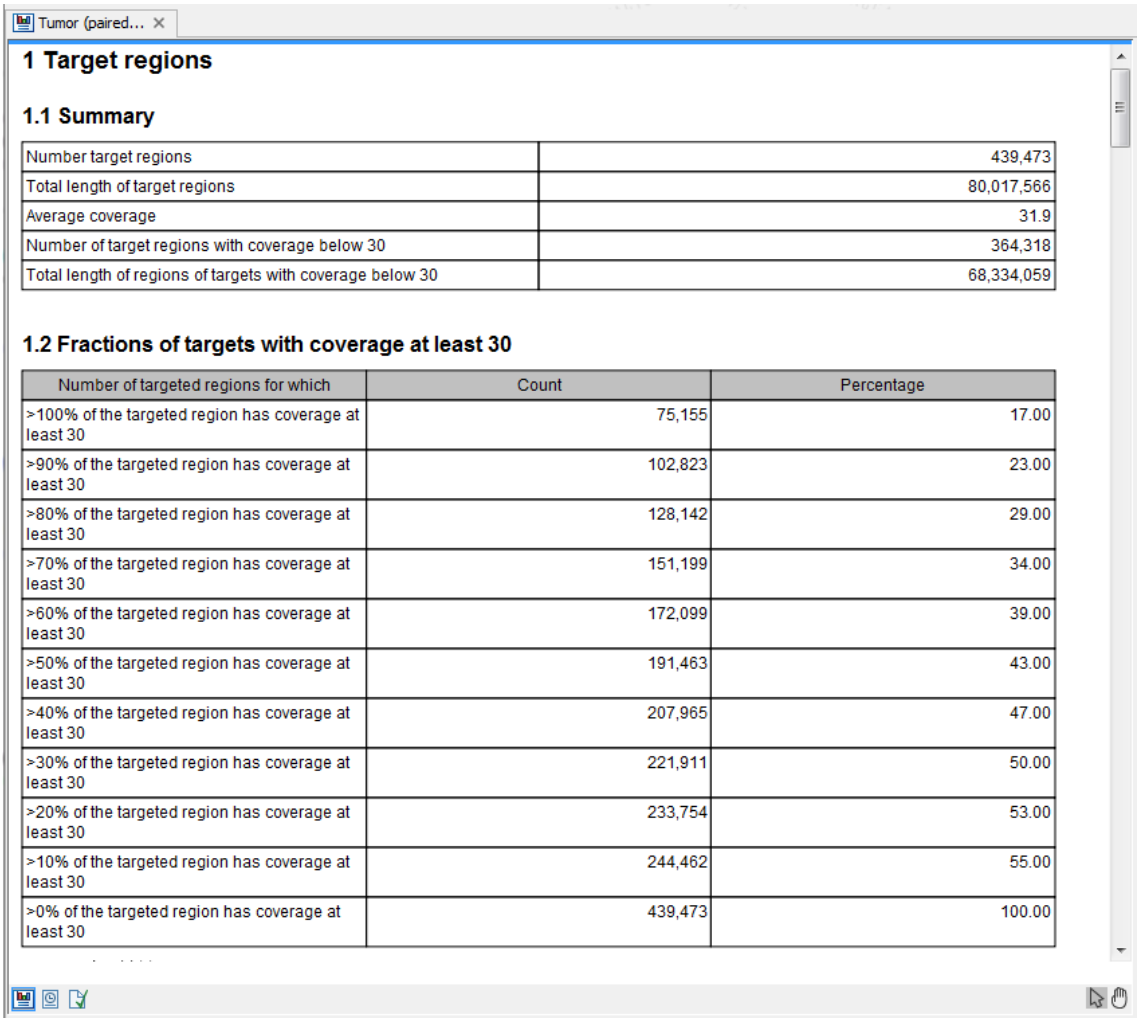



Figure 27.3: *The report with overviews of mapped reads.*

**Total length of target regions** The 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).

**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 27.8). 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 **Ignore non-specific matches** or **Ignore broken pairs** these reads will not contribute to the coverage. Note also that bases in over-lapping paired reads will only be counted as 1.

**Number of target regions with low coverage** The number of target regions which have positions with a coverage that is below the user-specified **Minimum coverage** threshold.

**Total length of target regions with low coverage** The total length of these regions.

**Fractions of targets with coverage at least...** This table shows how many target regions have a certain percentage of the region above the user-specified **Minimum coverage** threshold.

**Fractions of targets with coverage at least...** A histogram presentation of the table above in **Fractions of targets with coverage at least...**.

**Coverage of target regions positions** This plot shows the coverage level on the x axis, and the number of positions in the target regions with that coverage level.

**Coverage of target regions positions** 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 27.4 this means that 26.58 % of the positions on the target are covered by at least 40 bases.

**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 (figures 27.4 and 27.5). Note that, for the table that is concerned with *reads*, reads in over-lapping 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 27.6 and figure 27.7).

**Reference** The name of the reference sequence.

**Total mapped reads** The total number of mapped reads on the reference, including reads mapped outside the target regions.

**Mapped reads in targeted region** Total number of reads in the targeted regions. Note that if there are overlapping regions, reads covered by two regions will be counted twice. 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 that are in the targeted regions.

**Total mapped reads excl ingored** The total number of mapped reads on the reference, including reads mapped outside the target regions, excluding the non-specific matches or broken pairs, if the user has switched on the option to ignore those.

**Mapped reads in targeted region excl ingored** Total number of reads in the targeted regions, excluding the non-specific matches or broken pairs, if the user has switched on the option to ignore those.

**Specificity excl ingored** The percentage of the total mapped reads that are in the targeted regions.

**Reference** The name of the reference sequence.

**Total mapped bases** The total number of mapped bases on the reference, including bases mapped outside the target regions.

**Mapped bases in targeted region** Total number of bases mapped within in the targeted regions. Note that if there are overlapping regions, bases included in two regions will be counted twice.

**Specificity** The percentage of the total mapped bases that are in the targeted regions.

**Total mapped bases excl ingored** The total number of mapped bases on the reference, including bases mapped outside the target regions, excluding the bases in non-specific matches or broken pairs, if the user has switched on the option to ignore those.

**Mapped bases in targeted region excl ingored** Total number of bases in the targeted regions, excluding the bases in non-specific matches or broken pairs, if the user has switched on the option to ignore those.

**Specificity excl ingored** The percentage of the total mapped bases that are in the targeted regions.

**Distribution of target region length** A plot of the length of the target regions, and a version of the plot where only the target region lengths that lie within +3SDs of the median target length are shown.

**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 position** 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.

### 27.1.3 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 27.6:

**Chromosome** The name is taken from the reference sequence used for mapping.

**Region** The region of the

**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** level provided in figure 27.1.

**Percentage with coverage above...** The percentage of the positions in the region with coverage above the **Minimum coverage** level provided in figure 27.1.

### 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% |

### 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 |

Figure 27.4: *The report: mapped reads.*

| 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 27.5: *The report: mapped bases.*

**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 over-lapping pairs are counted individually (see figures 27.6 and figure 27.7).

**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 27.6 and figure 27.7).

**%GC** The GC content of the region.

**Min coverage** The lowest coverage in the region.

**Max coverage** The highest coverage in the region.

**Mean coverage** The average coverage in the region.

**Median coverage** The median coverage in the region.

**Zero coverage bases** The number of positions with no coverage.

**Mean coverage (excluding zero coverage)** The average coverage in the region, excluding any zero-coverage parts.

**Median coverage (excluding zero coverage)** The median coverage in the region, excluding any zero-coverage parts.

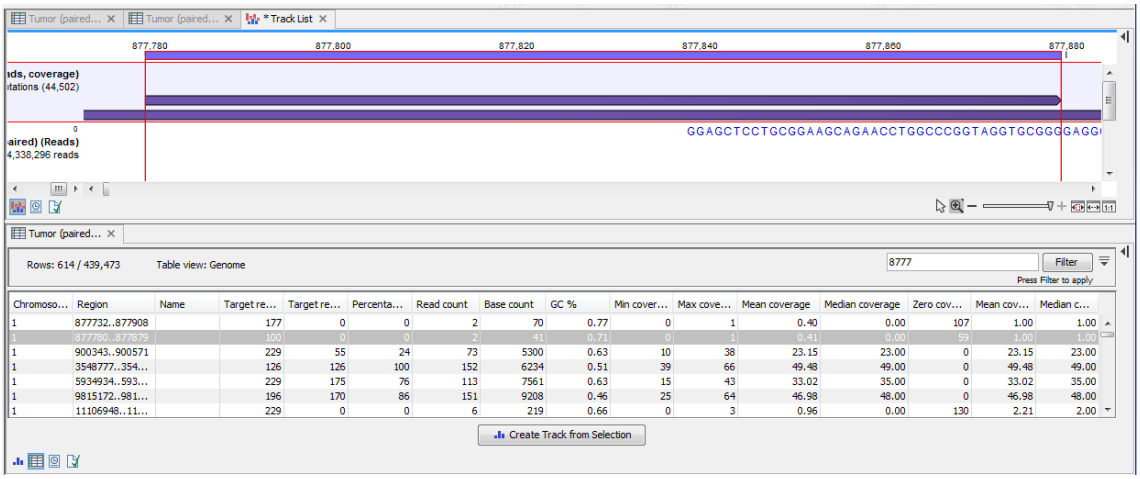



Figure 27.6: *A track list containing the target region coverage track and reads track. The target region coverage track has been opened from the track list and is shown in table view. Detailed information on each region is displayed. Only one paired read maps to the region selected.*

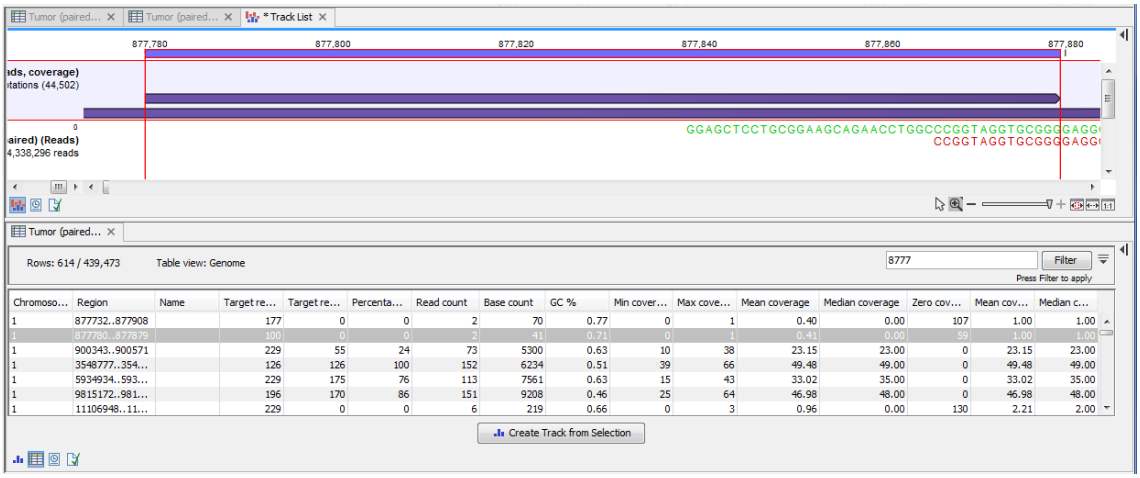



Figure 27.7: *The same data as shown in figure 27.6, but now the Disconnect paired reads option in the side-panel of the reads track has been ticked, so that the two reads in the paired read are shown disconnected.*

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 24.1.

### 27.1.4 Coverage table

Besides standard information such as position etc, the coverage table (figure 27.8) 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 over-lapping 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 25.2).

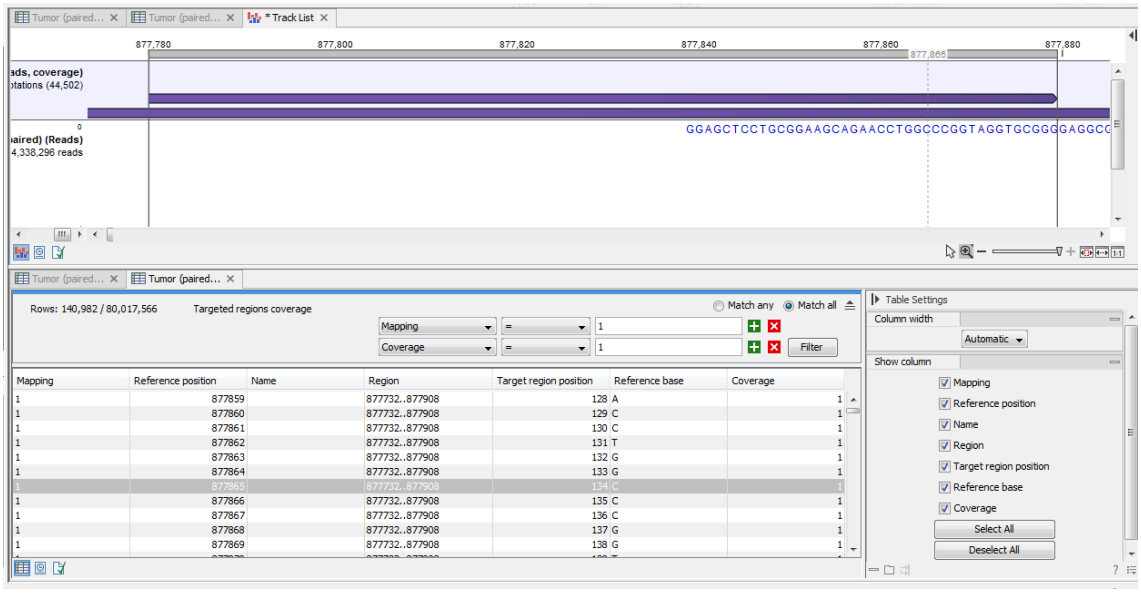



Figure 27.8: *The targeted region coverage table for the same region as shown in same as shown in figures 27.6 and figure 27.7.*

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 24.1.

### 27.1.5 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 27.9). The x-axis and y-axis values are identical to those found in the columns of the coverage table.

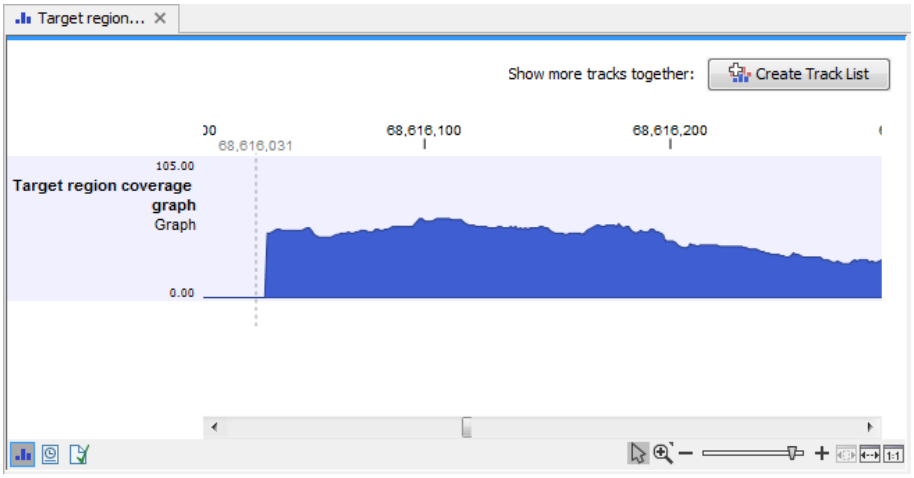



Figure 27.9: *An example of a targeted region coverage graph.*

## 27.2 Variant Detectors - overview

The Variant Detectors tools consist of three different tools for variant detection: The 'Basic Variant Detection' tool, the 'Fixed Ploidy Variant Detection' tool, and the 'Low Frequency Variant Detection' tool. The tools are designed for analysis of different types of samples. They differ in their underlying assumptions about the data, and hence differ in their assessments of when there is enough information in the data for a variant to be called. They share a number of options regarding which reads in the data should be considered, and options related to the filtering of called variants.

The Cancer and Genomics Workbenches offer three tools for detecting variants:

- The 'Basic Variant Detection' tool
- The 'Fixed Ploidy Variant Detection' tool and
- The 'Low Frequency Variant Detection' tool

The 'Basic Variant Detection' and the 'Fixed Ploidy Variant Detection' tools are new versions of the 'Quality-based Variant Detection' and the 'Probabilistic Variant Detection' tools, in which the filtering options have been unified and extended. The 'Quality-based Variant Detection' and the 'Probabilistic Variant Detection' are on their way to be retired and have been moved to the Legacy Tools.

The Basic Variant Detection, Fixed Ploidy Variant Detection and the Low Frequency Variant Detection tools *differ in their underlying assumptions about the data*, and are suitable for different types of applications. An overview is given in figure 27.10.

Below we first describe the tools. Each of the tools have a set of parameters that are specific to that tool. Next, we describe the filtering and output options that are shared among the tools (section 27.7).

To run the Variant Detection tools, go to:

**Toolbox** | **Resequencing Analysis (** ) | **Variant Detectors**

Here you are presented with the three tools (see figure 27.11).

| Variant caller | Applications | Data | Variant detected | Comments |
|---|---|---|---|---|
| Basic | Detection of germline and somatic variants | Any | Will detect any variant observed in the reads | None |
| Fixed ploidy | Detection of 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 |
| Low Frequency | Detection of 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 |

Figure 27.10: *An overview of the applicabilities of the variant detection tools.*

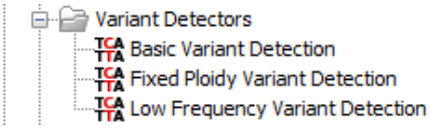



Figure 27.11: *The Variant Detection tools.*

When double-clicking one of the tools, a dialog is opened where you select the **reads track** or read mapping you want to analyze.

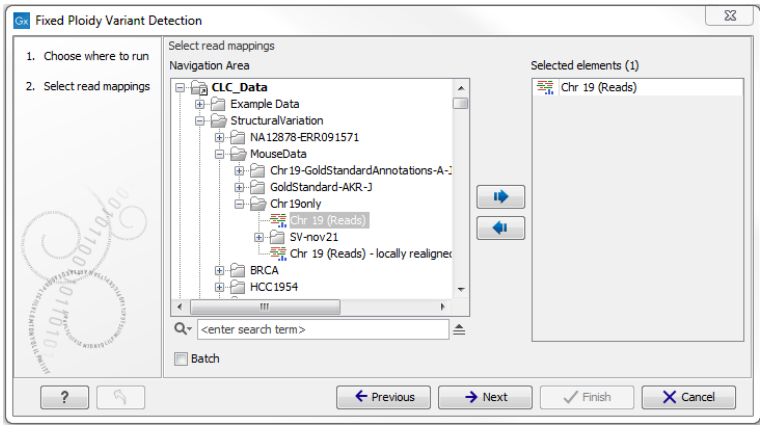



Figure 27.12: *Select the read mapping that you want to analyze.*

Click **Next** when the reads track is listed in the right-hand side of the dialog.

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 in the sections 27.3, 27.4, 27.5. First we will here give some examples of different variants called by the three variant detection tools and the overall mode by which the variant detection is carried out.

### 27.2.1 Differences among the variants called by the three variant callers

The Variant Detection tools will call SNVs, MNVs (which are neighboring SNVs for which there is evidence in the data that they occur together), small to medium-sized insertions and deletions (the size of the insertions and deletions that the variant detection tools are able to call is

restricted by the fact that they need to be represented within a single read), and replacements (which are neighboring SNVs and indels).

As the tools *differ in their underlying assumptions about the data*, they differ in their assessments of when there is enough information in the data for a variant to be called, and hence *will call different variants*. However, when run with the *same* filter settings (section 27.7 for a description of the filters), you will generally have that:

- The Basic Variant Caller will call the highest number of variants. It will also do this relatively quickly, as it does not do any error-model estimation.
- The Low Frequency Variant Caller will call *a subset of the variants called by the Basic Variant caller*. The variants called by the Basic Variant Caller that the Low Frequency Variant Caller will NOT call, are those that, according the error model that the Low Frequency Variant Caller estimates from the data, are likely to have been caused by sequencing errors. The Low Frequency Variant Caller will be the slowest of the three variant callers as it (1) estimates an error-model and (2) calls Low Frequency variants (and not just those that are in accordance with a specified ploidy model).
- The Fixed Ploidy Variant Caller will call *a subset of the variants called by the Low Frequency Variant caller*. The variants called by the Low Frequency Variant Caller that the Fixed Ploidy Variant Caller will NOT call, are those that, according to the assumed ploidy of the sample analyzed and the error model that the Fixed Ploidy Variant Caller estimates from the data, are likely to have been caused by either mapping errors or by sequencing errors.




Figure 27.13: *The differences in variants called by the three variant callers. The variant callers have all been run with same the filter settings (those that are the defaults for the Low Frequency Variant Caller).*

Figure 27.13 shows variant calls produced by the three variant callers when run with the same filter settings, more precisely those that are default for the Low Frequency Variant Caller. The numbers of called variants are shown in the left part of the figure, under the variant track names 'basicV2', 'LowFreq' and 'FixedV2'. The Basic Variant Caller calls most variants and the Fixed Ploidy the least. The Fixed Ploidy Variant Caller calls a subset of those called by the Low Frequency Variant caller, which in turn calls a subset of those called by the Basic Variant caller — in spite of the fact that there are 9 variants in the Low Frequency variant track that are not in the Basic Variant track. Although those 9 variants are in fact not in the Basic Variant track, they are 'sub-variants' of variants in that track. The highlighted variants in the figure is an example of this: The Basic variant caller has called a heterozygous 2bp MNV. The Low Frequency variant caller has judged that one on the SNVs constituting this 2bp MNV is likely to be the result of sequencing errors, and has only called one of the SNVs.

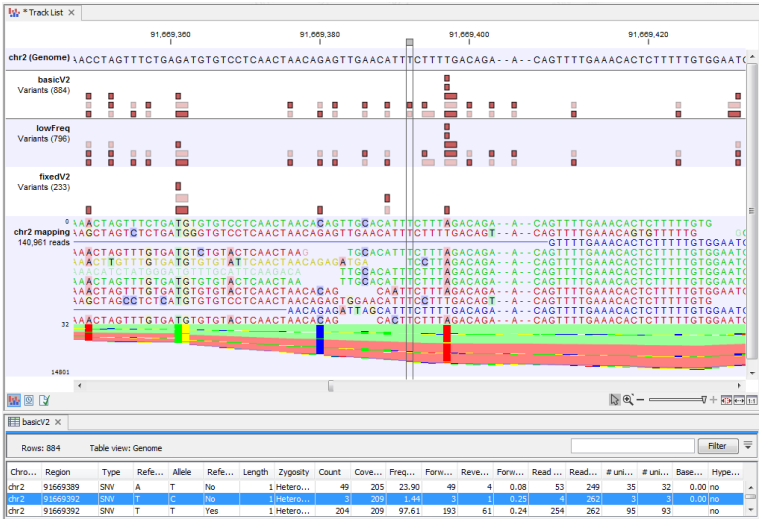



Figure 27.14: *A variant is highlighted that is detected by the Basic Variant Caller but not by the Low Frequency or the Fixed Ploidy Variant Caller. The variant track for the Basic variant Caller variants is opened in the table-view at the bottom of the figure. The variant is present at a low frequency in a high coverage position, and is likely to have been caused by sequencing error.*

In figure 27.14 a variant is highlighted that is detected by the Basic Variant Caller but not by the Low Frequency or the Fixed Ploidy Variant Caller. The variant is present at a low frequency in a high coverage position. The Low Frequency Variant Caller compares this evidence to the error model, and has decided that the three reads carrying the variant are likely to be the result of sequencing errors, rather than the result of a true variant. Figure 27.15 highlights a variant that is detected by both the Basic and the Low Frequency Variant Caller, but not the Fixed Ploidy. The variant is present at a higher frequency (14.22%) in a high coverage region (coverage 204). Observing the variant in 29 out of 204 reads is not likely to be due to sequencing errors. However, observing 29 reads from one allele and the remaining from the other in a diploid sample is highly unlikely, and the Fixed Ploidy Variant Caller judges that this variant is most likely caused by mapping errors (that is, a subset of the reads in the region being mapped there spuriously) and filters out this variant.

### 27.2.2 How the variant detectors work

The Variant detectors share a set of filters. They relate to (i) which areas and positions of the read mappings that should be inspected for variants, (ii) which reads in the data should

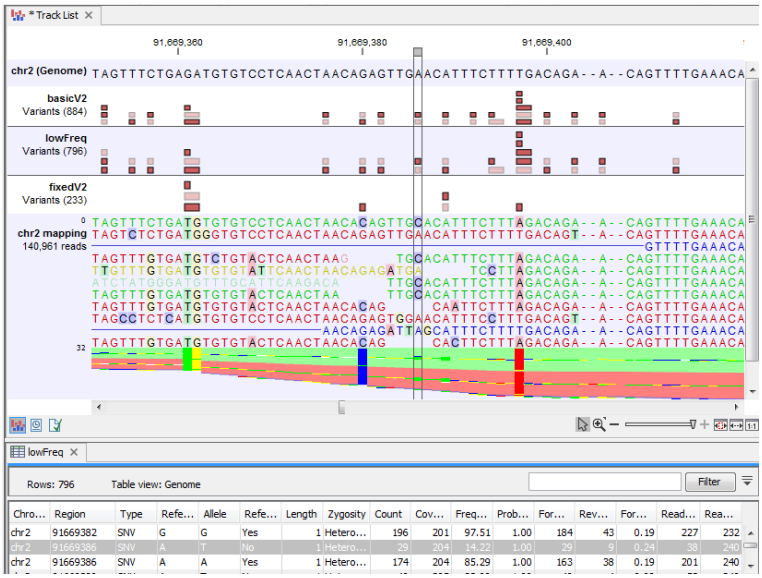



Figure 27.15: *A variant is highlighted that is detected by the Basic and the Low Frequency but not by the Fixed Ploidy Variant Caller. The variant track for the Low Frequency Variant Caller variants is opened in the table-view at the bottom of the figure. The variant is present at a moderate frequency in a high coverage position, and is,* under the assumed ploidy, *most likely to have been caused by mapping error.*

be considered when this assessment is done, (iii) requirements to the coverage, frequency and absolute counts of variant carrying reads and (iv) the quality and neighborhood composition of the area surrounding the variant. The filters are described in detail in section 27.7.

The variant detectors operate in the following step-wise fashion:

1. Estimate an error model (Fixed Ploidy and Low Frequency Variant Detectors only).
2. Examine each single nucleotide position for the presence of a potential variant while:
   (a) Ignoring the positions, regions and reads specified by the 'Reference masking' and 'Read filter' parts of the 'General filters' (section 27.7).
   (b) Applying half the cut-offs specified by the user in the 'Count and coverage filters' part of the 'General filters' (section 27.7).
   (c) Applying the cut-offs specified in the Noise filters (section 27.7).

   Discard the single nucleotide variant if it's presence in the reads is not 'significantly' better explained by being variant than by being due to sequencing and/or mapping errors (Fixed Ploidy and Low Frequency Variant Detectors only. For details see section 27.4 and 27.5.
3. For the single position variants that survived the initial screening in 2., examine if neighboring variants are present in the same reads. If so, 'join' them into MNVs, longer insertions or deletions, or into replacements.
4. Apply the full cut-offs of the 'Count and coverage filters' part of the 'General filters' to all the variants (single and multiple positions) that were obtained after 3.

The reason for first examining the read mapping for variants in single positions is that it is at the single position level that we can estimate an error-model, and hence it is at the single position

level that we can distinguish true variants from likely sequencing errors. The reason for joining the single position variants, *when they are present in the same reads*, is that this gives us evidence that they occur together (e.g., it is unsatisfactory to call three single base neighboring deletions if we can see that they occur in the same reads). The reason for applying only half of the cut-off of the 'Count and coverage filters' part of the 'General filters' on the initial single position examination (in step 2.), and wait with the full cut-off till after the variants have been joined (in step 4), is that it will sometimes happen that coverage drops within a longer variant. This can result in single position variants within a longer variant not being called, if the full filters were initially applied. By applying only half the cut-off in the initial examination, this risk is decreased.

Having described the overall mode of the variant detection tools, we will now describe individual characteristics and specific assumptions of the three variant detection tools. The filtering and output options that are shared among the tools are described in the (section 27.7) and section 27.8.

## 27.3 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 27.7). The tool has a single parameter (figure 27.16) 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 27.8 for a description of the outputs).

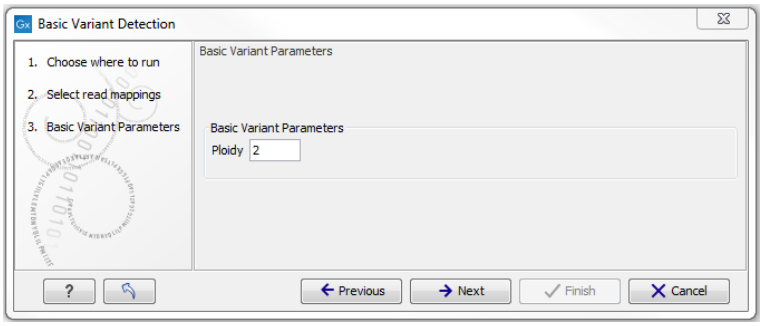



Figure 27.16: *The Basic Variant Detection parameters.*

## 27.4 Fixed Ploidy Variant Detection

The Fixed Ploidy Variant Detection tool relies on two models:

1. A model for the possible 'site-types' and
2. A model for the sequencing errors.

For (i), the set of possible 'site-types' depend 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 -/-. The error model, (ii), 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 27.6). 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 27.17):

- 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 27.4.1.
- The 'Required variant probability' is the minimum value of the variant probability required for the variant to be called. That is, only variants with a probability higher than the specified value will be called. That means that the higher the value you set, the fewer variants are called.

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.

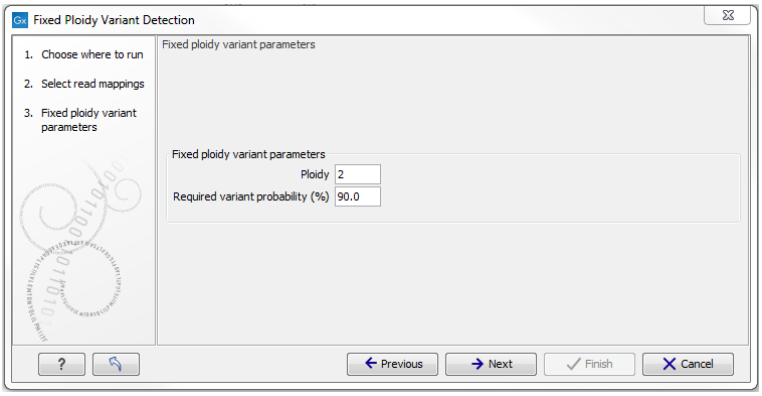



Figure 27.17: *The Fixed ploidy Variant Detection parameters.*

For a more in depth description of the Fixed Ploidy variant caller see Section 27.9.

### 27.4.1 Ploidy and sensitivity

Core to how this tool works is the ploidy level you set. This defines the statistical model that will be used during the variant detection analysis and thereby also defines what will be reported. When you set up a Fixed Ploidy Variant Detection, you provide a ploidy and you also provide a variant probability. 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, (note that it's 'the site' and not 'the variant' here), then the variant with the highest probability at that site will be reported. That is, at a given location, you get one variant 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). So how sensitive this tool is, in terms of what it detects and reports to you, will be down to:

- The statistical model. Here the ploidy you set is crucial to the model and what will be detected.
- The variant probability setting. This affects which locations will be reported.

So, to increase sensitivity, you could decrease the variant probability setting or increase the ploidy. The increased sensitivity due to decreasing the variant probability setting is due to more sites being reported on. This could decrease false negatives, but it could also increase your reporting of false positives.

The increased sensitivity you would get by increasing the ploidy level is down to the extra allele types that will be included in the model and thus reported to you in the results. For example, let's say you set a ploidy of 2 and at a particular location, where the evidence that the position was different than the reference was high enough, the reference was a T. The variant with the highest probability at this location will be reported. Let's say that it was a homozygote with C at that position. So that's C|C. However, let's say that in fact, there are some Gs reported at that site in some reads. 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). However, in this example, C|C had the highest probability, so it is reported. It doesn't really matter how much deeper the mapping is at that location if the relative prevalence of Gs is low compared to Cs. (That is, the probability of C|C will stay higher than C|G if there were relatively few Gs at this site, so C|C would be reported as long as this stayed the case.) Let's say that you chose a ploidy of 3 instead. Now 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). Now, for the same site, say that the evidence in the reads resulted in the variant C|C|G having a higher probability than C|C|C, then it would be the variant reported. So in this way, you have increased the sensitivity. i.e you have reported a variant that represents the evidence of the reads with G at that position as well as the ones reporting a C at that position.

## 27.5 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.

The method employed in the Low Frequency Variant Detection tool for estimating the sequencing error rates is similar to that of the Fixed Ploidy Variant Detection tool (see section 27.6), but the statistical model for the sample is different. It does not make any assumptions about the ploidy of the sample. Instead 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 than the reference present in the sample at some unknown frequency. If the latter is the case, a variant corresponding to the significant allele will be called, with estimated frequency.

The Low Frequency Variant Detection tool has one parameter (figure 27.18):

- '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. That means that, 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, and 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).

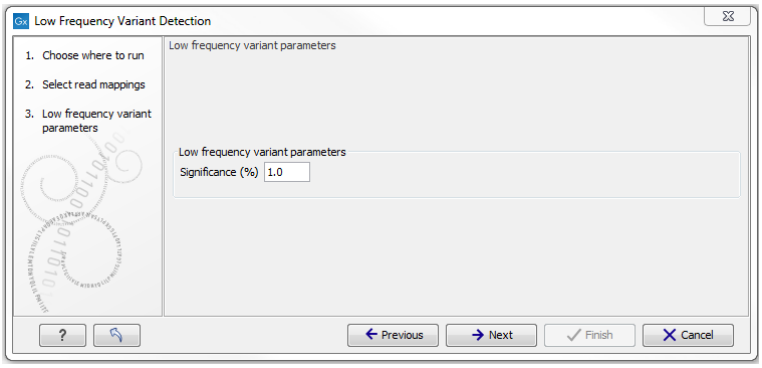



Figure 27.18: *The Low Frequency Variant Detection parameters.*

For a more in depth description of the Low Frequency variant caller see Section 27.9.

## 27.6 Variant Detectors - 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 27.8).

**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 27.19: *Example of estimated error rates. The figure shows average estimated error rates across bases in the given quality score intervals (20-29 and 30-39, respectively). Higher error rates are estimated for bases with lower quality scores.*

An example of error rates estimated from a whole exome sequencing Illumina data set is shown in figure 27.19. As expected, the estimated error rates (that is, the off-diagonal elements in the matrices in the figure) are higher for the lower quality nucleotides than for higher. Note 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.

## 27.7 Variant Detectors - filters

The variant detectors offer a number of filters. These relate both to which reads should be used, and how much evidence should be required for a variant to be called. The user is asked to set the values of these filters in two wizard steps: the 'General filters' step (figure 27.20) and the 'Noise filters' step (figure 27.21). Note that, for most of the filter, the values that they calculate and base their filtering on are also added as annotations to the variants (see section 27.8). This means that they are 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 choosing to apply the filtering within the variant calling detection it self. The filters are described below.

### 27.7.1 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:

**Note on the use of the rare variant caller with WGS data:** The default settings for the rare variant caller are optimized for targeted resequencing protocols, not whole genome sequencing (e.g. cancer gene panels). In order to run the tool on WGS data the the parameter 'Ignore positions with coverage above' should be adjusted (to e.g. 1000) as 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 simply because there will be many low frequency variants (due to some reads originating from near identical repeat sequences or simple sequencing errors). As these 'high coverage' regions typically are not of interest, disregard these areas by using a lower

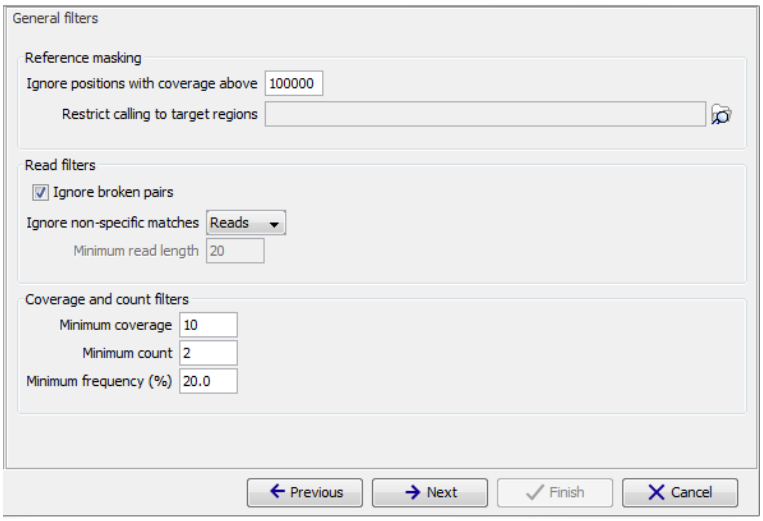



Figure 27.20: *General filters. The values shown are those that are default for Fixed Ploidy Variant detection.*

'Ignore positions with coverage above' filtering setting is recommended for WGS data.

### Reference masking

The 'Reference masking' filters allow the user to only perform variant calling (incl. error model estimation) in specific regions. There are two parameters to specify this:

- **Ignore positions with coverage above:** All positions with coverage above this value will be ignored when inspecting the read mapping for variants.
- **Restrict calling to target regions:** Only positions in the regions specified will be inspected for variants.

Note that when the option **Ignore positions with coverage above** is chosen, any position with coverage above the specified value will be ignored by the algorithm, and no variants will be called at these positions. The option is highly useful in cases where you have a read mapping which has areas of extremely high coverage, that you are not really interested in. An example is areas around centromeres in whole genome sequencing applications. The computational requirements may be prohibitive if you consider the full read mapping, but manageable if you exclude the (un-interesting) high-coverage areas.

Also note that the **Restrict calling to target regions** parameter is optional. When not specified, the full read mapping will be examined.

### 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.

- **Non-specific match filter:** Non-specific matches are likely to come from some type of repeat region, and the exact mapping location of them 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': when this option is chosen they are not ignored.
  - 'Reads': when this option is chosen they are ignored.
  - 'Region': when this option is chosen no variants are called in regions covered by at least one non-specific match.

  When ignoring regions containing a non-specific match (the last of the options mentioned above), the minimum length of reads that are allowed to trigger this effect has to be stated. The reason is that we want to avoid really short reads triggering the effect, as really short reads will usually be non-specific even if they do not stem from repeat regions.

Please note that ignored broken pair reads will not be considered for any non-specific match filters.

#### 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, at least, this 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' are considered. The 'countable reads' are those that the user has not chosen to ignore. This means that, if the user, in the read filter, has specified that reads from broken pairs should be ignored, broken pair reads will not be countable. Similarly goes for the 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).

### 27.7.2 Noise filters

The 'Noise filters' examine each candidate variant at a more detailed level. They are intended as a means of filtering out variants that are likely to be the result of various types of systematic errors and/or biases, e.g. induced by the amplification or sequencing protocol, that may occur in samples. They should be used with care, as there is always the risk that a real variant has the characteristics of systematically induced variant.

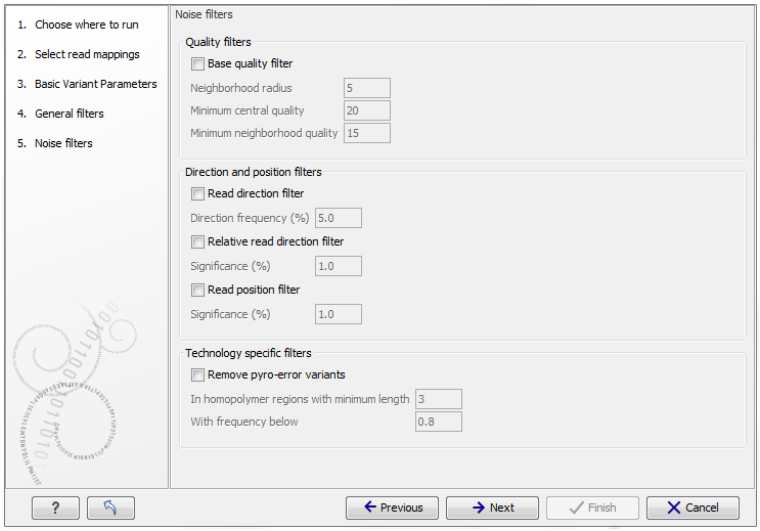



Figure 27.21: *Noise filters.*

#### 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 read 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: when a neighborhood radius of five is used, each nucleotide in a read is evaluated based on the nucleotides in the read 5 positions upstream and 5 positions downstream of the examined site - a total of 11 nucleotides. (Note that, near the end of the reads, eleven nucleotides are still considered, by changing the region offset relative to the nucleotide in question).
  - **Minimum central quality:** Reads whose central base has a quality below this value are ignored. This parameter does not apply to deletions, since there is no 'central base' in these cases.
  - **Minimum neighborhood quality:** Read for which the minimum quality of the bases *within the specified neighborhood radius* is below this value, are ignored.

Figure 27.22 gives an example of a variant that is called when the base quality filter is NOT applied, and not called when it is. To understand why it is not called when the base quality filter is applied look at the data in figure 27.23. This figure shows the same data as in figure 27.22, however, now with the 'Show quality scores' option in the side panel of the reads track switched on. This reveals that the reads that carry the potential 'G' variant tend to have poor quality. 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. 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 'drown' the signal of the 'T'.

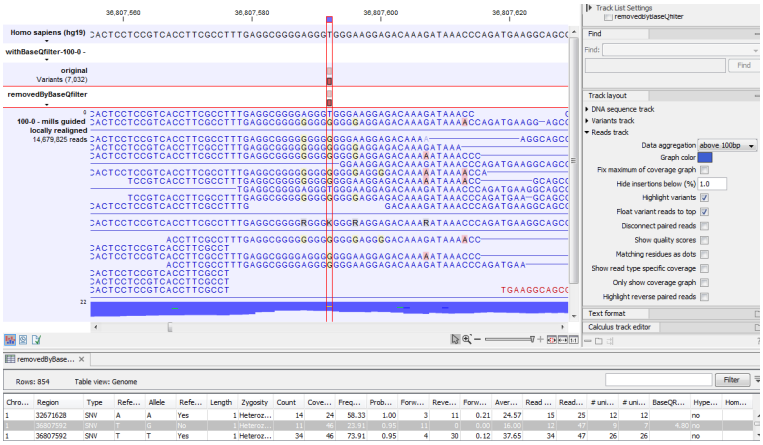


Figure 27.22: *An example of a variant that is removed by the base quality filter.*



Figure 27.23: *The same data as in figure 27.22, now with the 'Show quality scores' option in the reads track switched on.*

### 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 27.24 shows an example of a variant that is removed by the 'Read direction' filter.

Note that variant calling was done ignoring non-specific matches and broken pair reads, so only the 16 intact paired reads (the blue reads) are considered. To see the direction of the reads, you must adjust the viewer settings in the 'Reads track' side panel, to 'Disconnect paired reads'. This has been done in figure 27.25. Now it becomes apparent that the variant is found in the forward reads (that is, the green reads) of the 16 intact paired reads, and in no reverse reads (except the three that come from broken pairs, and which were ignored), and therefore removed by the read direction data.

Figure 27.26 shows an example of a variant that is removed by the read position filter, but not by the read direction filter. The variant is only present in a portion of the reads that cover the variant, and the portion or the reads that carry the variant have the variant occurring in read positions that are systematically different from what you would expect, given the general placement of reads covering the variant (e.g., none of the reads that start after position 186,641,600 carry the variant).

#### 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

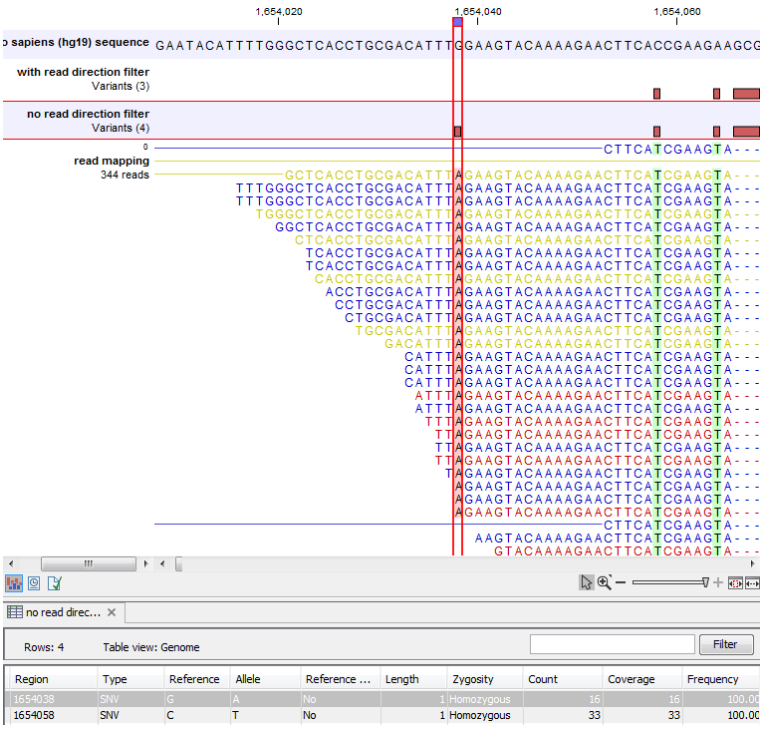



Figure 27.24: *An example of a variant that is filtered out by the Read Direction filter.*

regions or (2) as an 'overspill' a few nucleotides downstream of a homopolymer region. In case (1) the exact numbers of the same number of nucleotide is uncertain and a sequence like "AAAAAAAA" is sometimes reported as "AAAAAAAAA". In case (2) a sequence like "CGAAAAAGTCG" may sometimes get an 'overspill' insertion of an A between the T and C so that the reported sequence is C "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 maximum frequency** parameter, the more variants will be removed. Figure 27.27 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.

In addition to the example above, a simple example is provided below in figure 27.28 to illustrate the difference between variant frequency and pyro-variant removal frequency (where non-reference and non-homopolymer variant reads are ignored).

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.

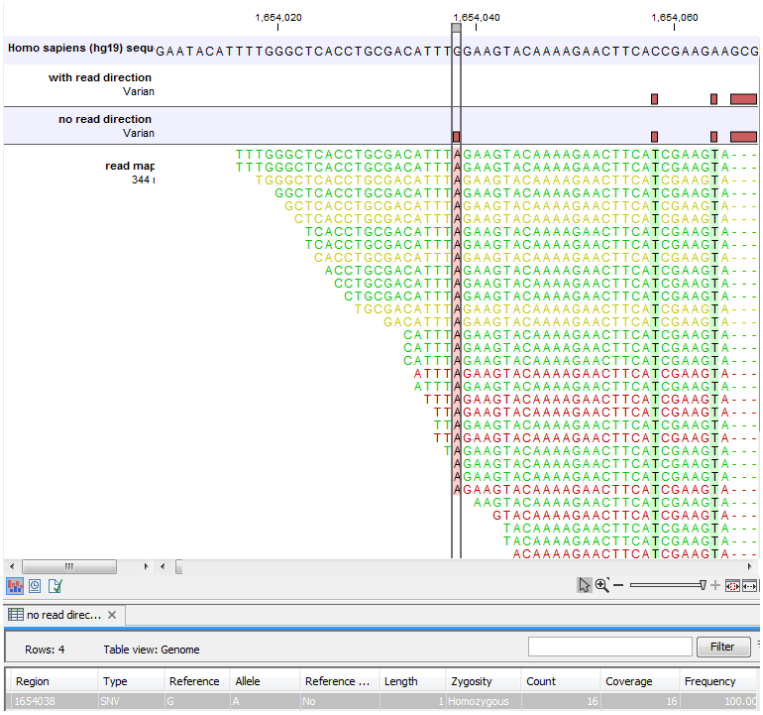



Figure 27.25: *The same data as shown in figure 27.24, but now with 'Disconnect paired reads' option switched on in the 'reads track' side panel.*

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).

## 27.8 Variant Detectors - the outputs

The Variant Detection Tools have the following outputs: a variant track, an annotated variant table and a report (figure 27.29). The report contains information on the estimated error model and, as only the Fixed ploidy and the Low Frequency variant callers uses an error model, the report is only available for those, and not for the Basic Variant caller. The outputs are described below.

### 27.8.1 The variant track output

The variant track contains information on each of the variants called. When opened in the table view there is a number of columns for each of the variants (see figure 27.30).

The contents of these are:

**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'.

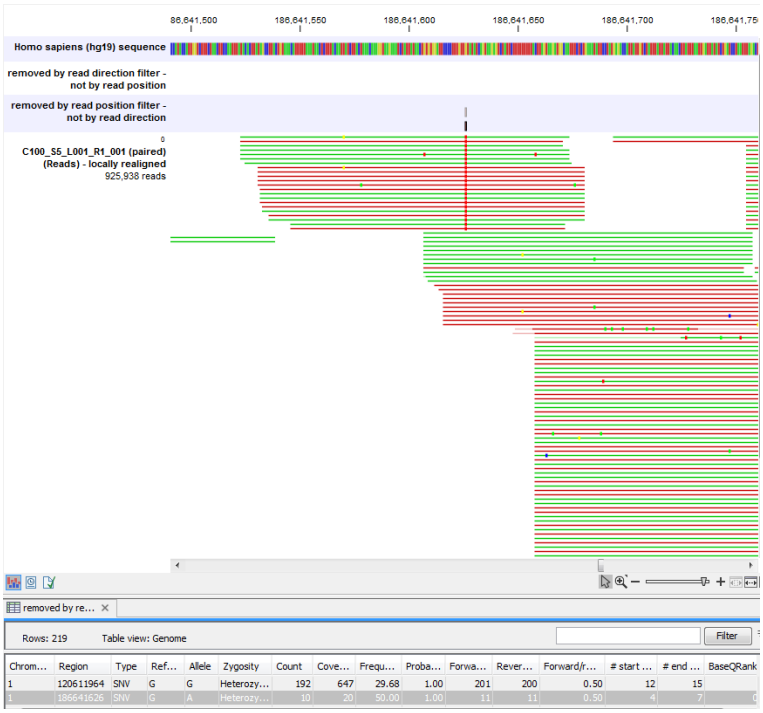



Figure 27.26: *A variant that is filtered out by the Read position filter but not by the Read direction filter.*

**Type** The type of variant. This can either be SNV (single-nucleotide variant), MNV (multi-nucleotide variant), insertion, deletion, or replacement.

**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 for one of the alleles for most, but not all, detected heterozygous variants (e.g. the variant caller 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 caller called the two variants 'C' and 'G' at the position, both would have had 'No' in the 'Reference allele' column).

**Length** The length of the variant. The length is 1 for SNVs, and for MNVs it is the number of allele or reference bases (which will always be the same). For deletions, it is the length of the deleted sequence, and for insertions it is the length of the inserted sequence. For replacements, both the length of the replaced reference sequence and the length of the inserted sequence are considered, and the longest of those two is reported.

**Zygosity** The zygosity of the variant called, as determined by the variant caller. 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' fragments supporting the allele. The 'countable' fragments are those that are used by the variant caller when calling the variant. Which fragments are 'countable' depends on the user settings when the variant calling is performed - if e.g. the

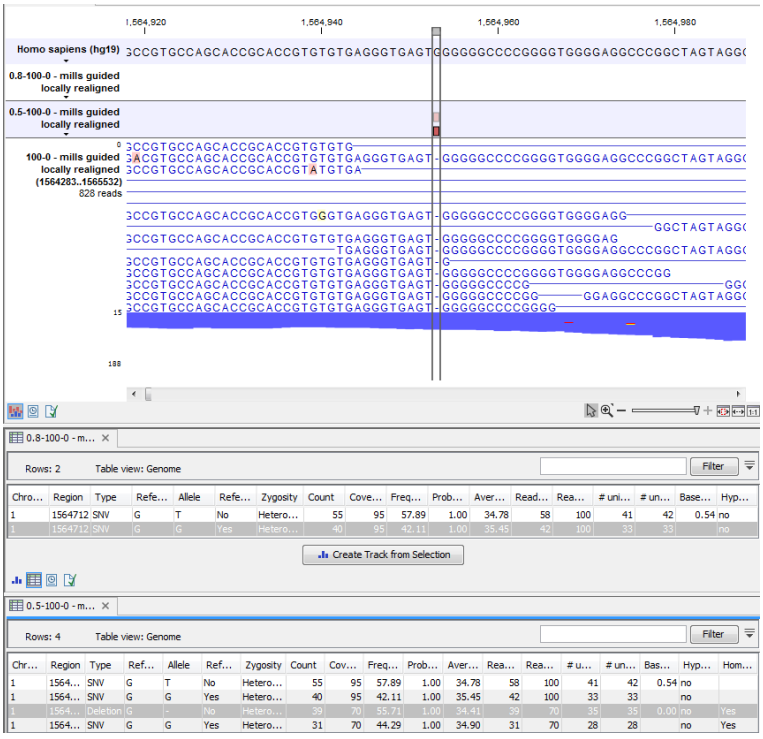



Figure 27.27: *An example of a variant that is filtered out when the pyro-error filter is applied with settings 3 and 0.8, but not with settings 3 and 0.5.*

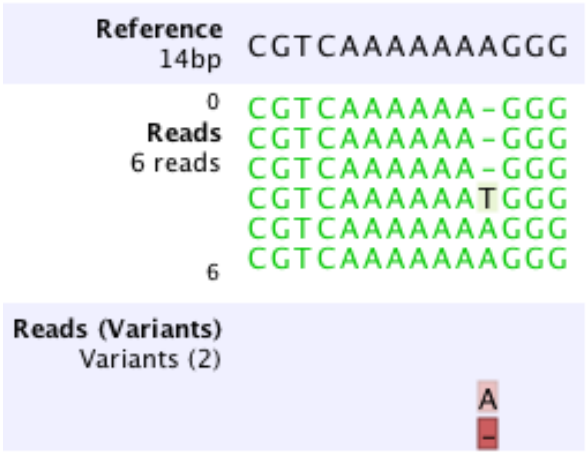



Figure 27.28: *An example of a simple read mapping with 6 mapped reads. Three of them indicate a deletion, two match the reference, and one read is an A to T SNP*

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').

**Coverage** The fragment coverage at this position. Only 'countable' fragments are considered

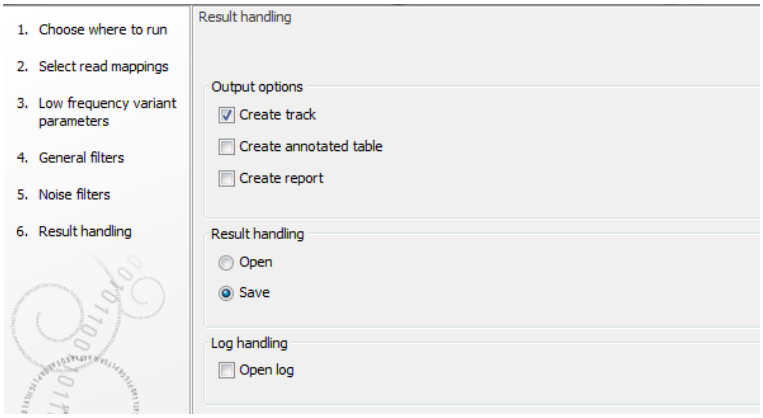



Figure 27.29: *Output options.*



Figure 27.30: *A variant track shown in the table view.*

(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').

**Frequency** 'Count' divided by 'Coverage'.

**Probability** The contents of the Probability column (for Low frequency and Fixed Ploidy variant callers only) depend on the variant caller 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 rare variant 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.

**Forward read count** The number of 'countable' forward reads supporting the allele (see under 'Count' above for an explanation of 'countable' reads).

**Reverse read count** The number of 'countable' reverse reads supporting the allele (see under 'Count' above for an explanation of 'countable' reads).

**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).

**Average quality** The average base quality score of the bases supporting a variant. In the case of a deletion, the quality score is taken from the average quality of the two bases neighbouring the deleted one, and the lowest is reported. Similarly for insertions, the quality in reads where the insertion is absent, is taken from the minimum average of the two bases either side of the position. It can be possible in rare cases, 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 **Filter Marginal Variant Calls** tool (see section 27.13.5).

**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 has a called variant compared with the quality scores of the reference allele. Variants for which no corresponding reference allele is called does not have a

BaseQRankSum value. Likewise, no values are calculated for reference alleles. The score is a Z score, so a value of 2.0 means that the observed qualities for the variant two standard deviations below the qualities for the reference allele. The scoring is performed using a Mann-Whitney U for comparing the two sets of quality scores from the reference allele and the variant.

**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** The test probability for the test of 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.

**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 variant, then all of them are marked as homopolymer.

### 27.8.2 The annotated table output

The 'Annotated table' output contains an 'old' style variant format output. For a description of the output, please see section 27.11.2.

### 27.8.3 The report

In addition to the estimated error rates of the different types of errors shown in figure 27.19, 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 27.31).

## 27.9 The Fixed Ploidy and Low Frequency variant callers: detailed descriptions

This section provides a detailed description of the models, methods and estimation procedures behind the Fixed Ploidy and Low frequency variant callers. For less detailed descriptions please see sections 27.4 and 27.5.

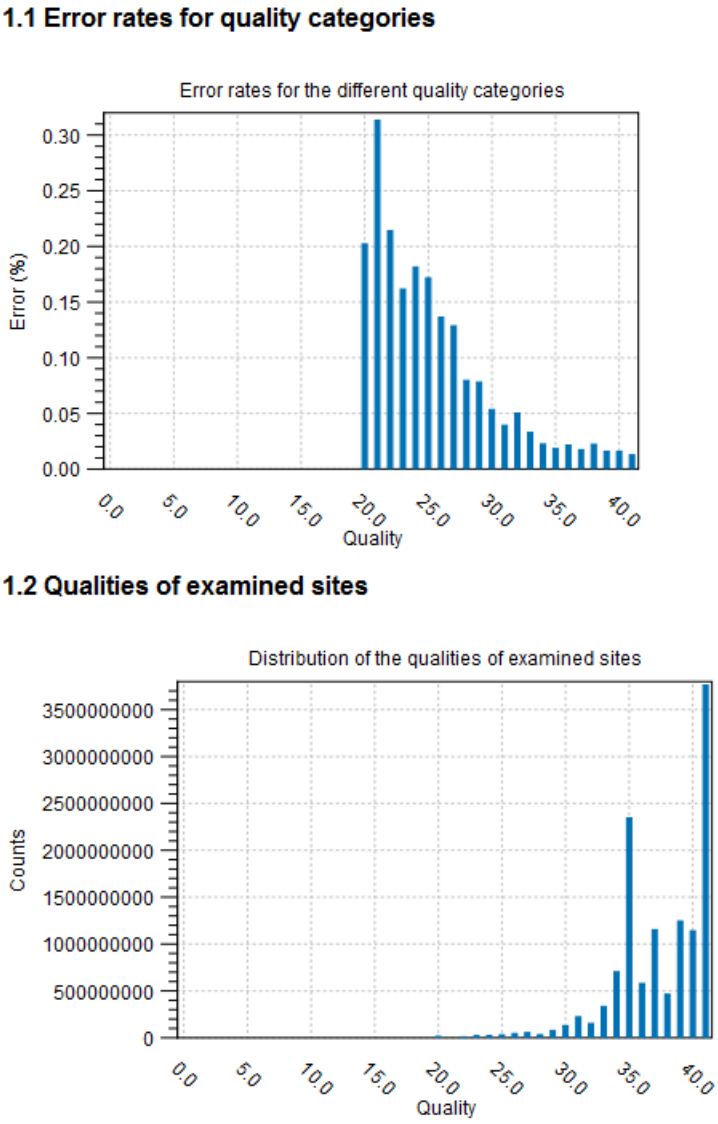



Figure 27.31: *Part of the contents of the report on the variant calling.*

### 27.9.1 The Fixed Ploidy Variant caller: Models and methods

This section describes the model, method and estimation procedure behind the Fixed Ploidy Variant caller. The Fixed Ploidy Variant caller is designed for detecting variants in samples for which the ploidy is known. As the Fixed Ploidy Variant caller 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 caller

The purpose of the Fixed Ploidy Variant Caller 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 caller 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 Caller

The statistical model for the Fixed ploidy variant caller 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:

$$\begin{aligned} P(t|data) &= \frac{P(data|t)P(t)}{P(data)} \\ &= \frac{P(data|t)P(t)}{\sum_{s\in S} P(data|s)P(s)}, \end{aligned} \tag{27.1}$$

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:

$$P(n_1, ..., n_k|t) = \prod_{i=1}^{k} \sum_{N\in\{A,C,G,T,-\}} P_t(N) \times e_q(N \to n_i). \tag{27.2}$$

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 (27.1), we thus have the following equation for the posterior probabilities of the site types:

$$\begin{aligned} P(t|n_1, ..., n_k) &= \frac{P(n_1, ..., n_k|t)f_t}{\sum_{s\in S} P(n_1, ..., n_k|s)f_s} \\ &= \frac{\prod_{i=1}^{k} \sum_{N\in\{A,C,G,T,-\}} P_t(N) \times e_q(N \to n_i)f_t}{\sum_{s\in S} \prod_{i=1}^{k} \sum_{N\in\{A,C,G,T,-\}} P_s(N) \times e_q(N \to n_i)f_s} \end{aligned} \tag{27.3}$$

The unknowns in this equation are the prior site type probabilities, $f_s, s \in S$, and the error rates $\{e(N \to M)|N, M \in \{A, C, G, T, -\}\}$. Once these have been estimated, we can calculate the posterior site type probabilities using the equation 27.3 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 caller

The Fixed Ploidy Variant Caller 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 $A$s and $C$s at site types $A/C$) and increase the likelihood of observing 'noisy' patterns (e.g. patterns of other than only $A$s, 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:

$$\begin{aligned} P(t|n_1, ..., n_k) &= \frac{P(t, n_1, ..., n_k)}{\sum_{s \in S} P(s, n_1, ..., n_k)} \\ &= \frac{P(t)P(n_1, ..., n_k|t)}{\sum_{s \in S} P(s)P(n_1, ..., n_k|s)} \end{aligned} \tag{27.4}$$

Now, for $P(t)$ we use our current value for $f_t$, and if we further insert the expression for $P(n_1, ..., n_k|t)$ (27.2) we get:

$$P(t|n_1, ..., n_k) = \frac{f_t \prod_{i=1}^{k} \sum_{N \in \{A,C,G,T,-\}} P_t(N) \times e_q(N \rightarrow n_i)}{\sum_{s \in S} f_s \prod_{i=1}^{k} \sum_{N \in \{A,C,G,T,-\}} P_s(N) \times e_q(N \rightarrow n_i)} \tag{27.5}$$

We get the updating equation for the prior site type probabilities, $f_t, t \in S$, from equation 27.5: 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 27.5) across all sites in the alignment:

$$f_t^* = \frac{\sum_{h=1}^{H} \frac{f_t \prod_{i=1}^{k} \sum_{N \in \{A,C,G,T,-\}} P_t(N) \times e_q(N \rightarrow n_i^h)}{\sum_{s \in S} f_s \prod_{i=1}^{k} \sum_{N \in \{A,C,G,T,-\}} P_s(N) \times e_q(N \rightarrow n_i^h)}}{H} \tag{27.6}$$

#### 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:

$$
\begin{aligned}
& P(r_i^h = N, n_1^h, ..., n_{k_h}^h) \\
&= \sum_{s \in S} f_s P(r_i^h = N, n_1^h, ..., n_{k_h}^h | s) \\
&= \sum_{s \in S} f_s \prod_i P(r_i^h = N, n_i^h | s) \\
&= \sum_{s \in S} f_s P(r_i^h = N, n_i^h | s) \prod_{j \neq i} P(n_j^h | s) \\
&= \sum_{s \in S} f_s (P_s(N) \times e_{q_{i^h}}(N \to n_i^h) \prod_{j \neq i} \sum_{N' \in \{A,C,G,T,-\}} P_s(N') \times e_{q_j}(N' \to n_j^h))
\end{aligned} \tag{27.7}
$$

Using Bayes formula again, as we did above in 27.4, we get:

$$
\begin{aligned}
P(r_i^h = N | n_1^h, ..., n_{k_h}^h) &= \frac{P(r_i^h = N, n_1^h, ..., n_k^h)}{P(n_1^h, ..., n_{k_h}^h)} \\
&= \frac{P(r_i^h = N, n_1^h, ..., n_{k_h}^h)}{\sum_{N' \in \{A,C,G,T,-\}} P(r_i^h = N', n_1^h, ..., n_{k_h}^h)}
\end{aligned} \tag{27.8}
$$

and inserting the expression from equation 27.7:

$$
\begin{aligned}
& P(r_i^h = N | n_1^h, ..., n_{k_h}^h) \\
&= \frac{\sum_{s \in S} f(s)(P_s(N) \times e_{q_{i^h}}(N \to n_i^h) \prod_{j \neq i} \sum_{N' \in \{A,C,G,T,-\}} P_s(N') \times e_{q_j^h}(N' \to n_j^h))}{\sum_{N' \in \{A,C,G,T,-\}} (\sum_{s \in S} f(s)(P_s(N') \times e_{q_{i^h}}(N' \to n_i^h) \prod_{j \neq i} \sum_{N' \in \{A,C,G,T,-\}} P_s(N') \times e_{q_j^h}(N' \to n_j^h)))} \\
&= \frac{\sum_{s \in S} f(s)(P_s(N) e_{q_{i^h}}(N \to n_i^h) \prod_{j \neq i} \sum_{N' \in \{A,C,G,T,-\}} P_s(N') \times e_{q_j^h}(N' \to n_j^h))}{(\sum_{s \in S} f(s) \prod_j \sum_{N' \in \{A,C,G,T,-\}} P_s(N') \times e_{q_j^h}(N' \to n_j^h)}
\end{aligned} \tag{27.9}
$$

The equation 27.9 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:

$$
e_q^*(N \to M) = \frac{\sum_h \sum_{i=1,...,k_h : n_i^h = M} P(r_i^k = N | n_1^h, ..., n_{k_h}^h)}{\sum_h \sum_{i=1,...,k_h} P(r_i^k = N | n_1^h, ..., n_{k_h}^h)}
$$

### 27.9.2 The Low Frequency Variant caller: Models and methods

This section describes the model, method and estimation procedure behind the Low Frequency Variant caller. The Low Frequency variant caller is designed to detect variants in a sample for which the ploidy is unknown. The low Frequency variant caller has a particularly high *sensitivity* for detecting variants that are present at any, and in particularly at low, allele frequencies.

#### The Low Frequency Variant caller

The purpose of the Low Frequency Variant Caller is to call variants in samples with unknown ploidy (e.g. samples of limited tumor purity from cancer patients) 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 Caller 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 The Low Frequency Variant Caller combines Multinomial models for the presence of variants and an error model for the sequencing (identical to that of the Fixed Ploidy variant caller). The Multinomial models are of the kind "there are $k$ different variants with frequencies $f_i$, $i = 1, ...k$, $\sum_{i=1}^{k} f_i = 1$", where the number variants,$k$, differ. Parameter estimation relies on the Maximum Likelihood principle, and, as the Fixed Ploidy Variant Caller, the EM algorithm is used for estimating the parameters of the model. Given an initial set of parameter values for the error rates, preliminary variants are called by examining the Log Likelihoods (evaluated at the Maximum likelihood estimates for the parameters of the multinomial models) for the data at each site for a set of Multinomial models. The model that offers the best explanation of the data (while taking care to adjust for the numbers of parameters in the Multinomial model) is chosen as the current guess of the true situation at that site, and given that, the error rates are re-estimated. Given the new error estimates, the Log Likelihoods for all possible multinomial models are again evaluated and updated variant calls are produced. This procedure is performed a total of four times. In contrast to the Fixed Ploidy variant caller it does not use a Bayesian approach for choosing among the multinomial models (since there are no meaningful prior site types), but instead uses a criterion adopted from the Akaike Information criterion, to choose among competing Mulitnomial model hypotheses for explaining the data.

#### The model for the Low Frequency Variant Caller

For the low frequency variant caller, we have a large number of possible hypotheses which can explain why the observed set of nucleotides are present in the sample at a given site. The most general situation is that the sample is a mixture of all variants corresponding to all four nucleotides and some missing nucleotides (gaps in the read mapping). We can write this hypothesis as [A/C/G/T/-]. At the other end of the spectrum, we have sites that are only made up of a single nucleotide (or only gaps in the read mapping). We can write these as [A], [C], etc. Each of these hypothesis can be described by a Multinomial model.

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.

#### Updating the choice of favored Multinomial model for each site

Given a set of error rates, we can find the maximum likelihood estimates of the underlying frequencies for each possible hypothesis for a given site. This also gives us the maximum likelihood value that can be obtained for the site under that hypothesis. Since the hypotheses with few free parameters are special cases of hypotheses with more free parameters, the hypotheses with the most free parameters will also have the highest likelihoods.

We will only favor a hypothesis with many free parameters if it offers a significantly higher likelihood than a hypothesis with fewer parameters. Let us consider a simple case where we have 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 (the hypotheses are nested). We can calculate the log likelihood ratio:

$$\frac{L(H_1)}{L(H_0)}$$

If this ratio is high, we tend to prefer hypothesis $H_1$ and if it is low (i.e. close to 1), we prefer $H_0$. It turns out that twice the log likelihood ratio is often $\chi^2$ distributed with a parameter given by the difference between the number of free parameters in the hypotheses, $n$. In our example $n = 1$ so:

$$2 \log \frac{L(H_1)}{L(H_0)} \sim \chi^2(1)$$

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 generalize this to apply to any two Multinomial model hypothesis $H_x$ and $H_y$. For these two, calculate the values (where $df$ is the degrees of freedom in a hypothesis):

$$v_x = 2 \log L(H_x) - c_{df_x}(p)$$

$$v_y = 2 \log L(H_y) - c_{df_y}(p)$$

(use $c_0(p) = 0$ for zero degrees of freedom). We now prefer the hypothesis with the highest value of $v$. When comparing a hypothesis with zero free parameters to another hypothesis, we get exactly the same results as with the log likelihood ratio approach.

We use this approach when comparing the many hypotheses that are present in the low frequency variant caller. For each one we calculate a $v$ value as twice the log likelihood and subtract a cutoff value $c$ which is based an the $p$ value and the degrees of freedom for that hypothesis. We then choose the hypothesis with the highest $v$ as the one that best describes the site in question.

For stringent $p$ values (i.e. values close to zero) we tend to prefer hypotheses with few free parameters which means that more sites tend to be called as homozygous.

The approach used here is similar to the Akaike Information Criteria except that we have introduced a way to use a $p$ value with the comparisons.

### Updating equations for the error rates

For the updating equations for the error probabilities we use the same procedure as for the Fixed Ploidy variant caller: We will use equation 27.7, however, in the case of the Low Frequency variant caller we do not have prior site type probabilities but instead have Maximum Likelihood estimates of the fractions of each nucleotide at the sites under the currently favored Multinomial model for that site. Let $P_h(N)$ be the frequency of the nucleotide $N$ at site $h$ as determined by our current Maximum Likelihood estimates for these. 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 $n_1^h, ..., n_{k_h}^h$:

$$\begin{aligned} & P(r_i^h = N, n_1^h, ..., n_{k_h}^h) \\ &= P_h(N) \prod_i P(r_i^h = N, n_i^h) \\ &= P_h(N) P(r_i^h = N, n_i^h) \prod_{j \neq i} P(n_j^h) \\ &= P_h(N) \times e_{q_{i^h}}(N \to n_i^h) \prod_{j \neq i} \sum_{N' \in \{A,C,G,T,-\}} P_h(N') \times \end{aligned} \tag{27.10}$$

As for the Fixed Ploidy variant caller we use Bayes formula to get:

$$P(r_i^h = N | n_1^h, ..., n_{k_h}^h) = \frac{P(r_i^h = N, n_1^h, ..., n_{k_h}^h)}{\sum_{N' \in \{A,C,G,T,-\}} P(r_i^h = N', n_1^h, ..., n_{k_h}^h)} \tag{27.11}$$

and insert the expression from equation 27.10 to get:

$$\begin{aligned} & P(r_i^h = N | n_1^h, ..., n_{k_h}^h) \\ &= \frac{P_h(N) \times e_{q_{i^h}}(N \to n_i^h) \prod_{j \neq i} \sum_{N' \in \{A,C,G,T,-\}} P_h(N') \times e_{q_j}(N' \to n_j^h)}{\sum_{N' \in \{A,C,G,T,-\}} P_h(N') \times e_{q_{i^h}}(N' \to n_i^h) \prod_{j \neq i} \sum_{N' \in \{A,C,G,T,-\}} P_h(N\prime) \times e_{q_j}(N' \to n_j^h)} \\ &= \frac{P_h(N) \times e_{q_{i^h}}(N \to n_i^h) \prod_{j \neq i} \sum_{N' \in \{A,C,G,T,-\}} P_h(N\prime) \times e_{q_j}(N' \to n_j^h)}{\prod_j \sum_{N' \in \{A,C,G,T,-\}} P_h(N\prime) \times e_{q_j}(N' \to n_j^h)} \end{aligned} \tag{27.12}$$

We calculate the $P(r_i^h = N | n_1^h, ..., n_{k_h}^h)$ values in equation 27.12, using our current error estimates and our current Maximum likelihood values of the frequencies for our current multinomial models of choice at each of the sites. We then use these to get new updated values for the error rates in a manner similar to that of the Fixed Ploidy Variant Caller we use the equation:

$$e_q^*(N \to M) = \frac{\sum_h \sum_{i=1,...,k_h : n_i^h = M} P(r_i^k = N | n_1^h, ..., n_{k_h}^h)}{\sum_h \sum_{i=1,...,k_h} P(r_i^k = N | n_1^h, ..., n_{k_h}^h)}$$

## 27.10 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 CLC 'Map reads to reference' tool with the *'global'* option switched on will have NO unaligned ends and the Structural Variation 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 Mapper are NOT optimal for the detection of structural variants with this tool. This is due to the fact that, the Large Gap Read Mapper will map some reads with (large) gaps, that would be mapped with unaligned ends with standard read mappers, and thus will leave a weaker unaligned end signal in the mappings for the Structural Variation tool to work with.

In it's current version the InDels and Structural Variants tool has the following known limitations:

- It will only detect intra-chromosomal structural variants.

### 27.10.1 How to run the InDels and Structural Variants tool

To start the structural variant detection:

**Toolbox** | **Resequencing** ( ) | **InDels and Structural Variants tool** ( )

This will open up a dialog. Select the read mapping of interest as shown in figure 27.32 and click on the button labeled **Next**.

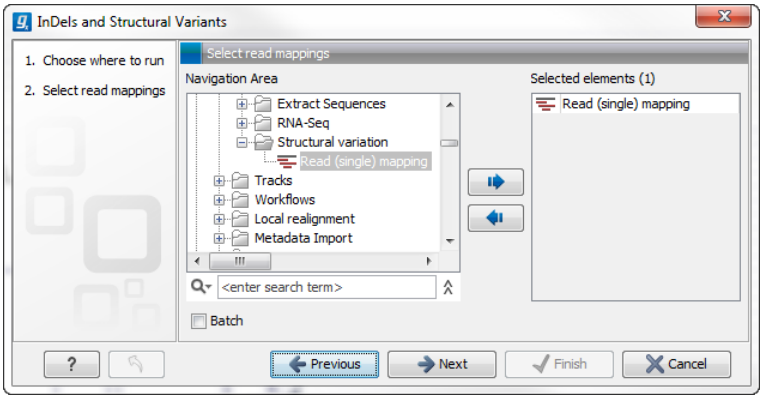



Figure 27.32: *Select the read mapping of interest.*

The next wizard step (Figure 27.33) 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. For further details about the algorithm see Section 27.10.3.

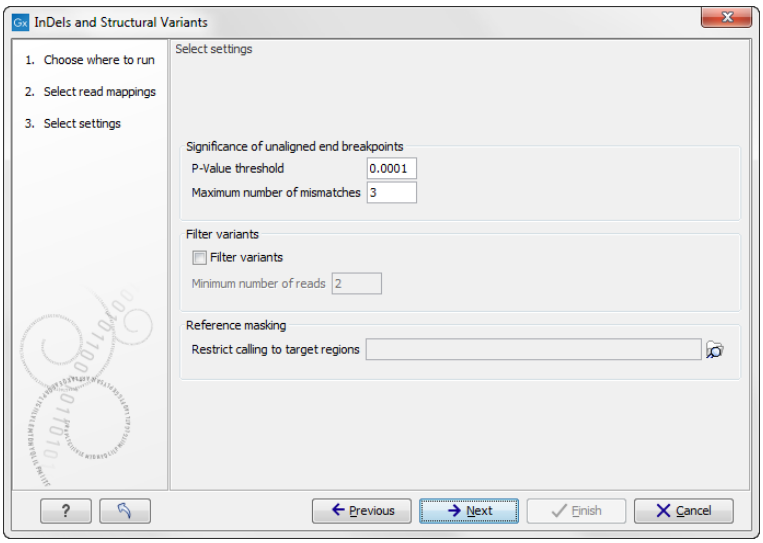



Figure 27.33: *Select the relevant settings.*

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 **'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.

**'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 27.34) 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, or only partly, known. Also, only deletions of six up to 200 bp are included in the variant track. See section 27.11.1 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 variants track' (described below).
- **Create structural variations** When ticked, a track containing the detected structural variants is created.

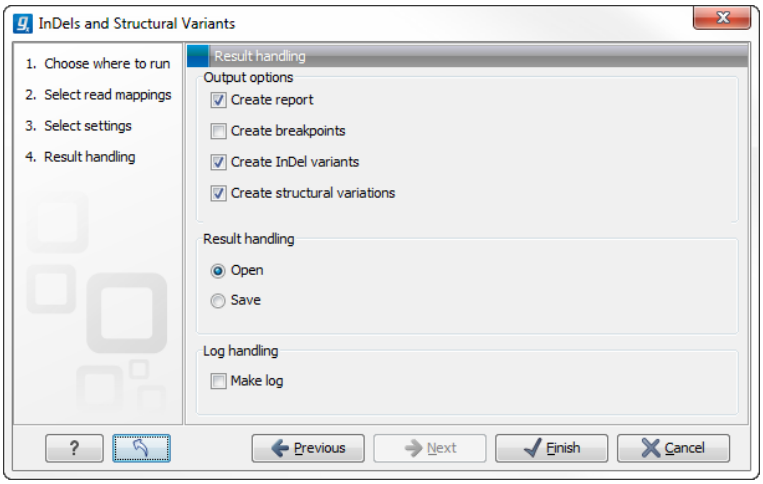



Figure 27.34: *Select output formats.*

An example of the output from the InDel and Structural Variant tool is shown in Figure 27.35. The output is described in detail in the next section (Section 27.10.2).

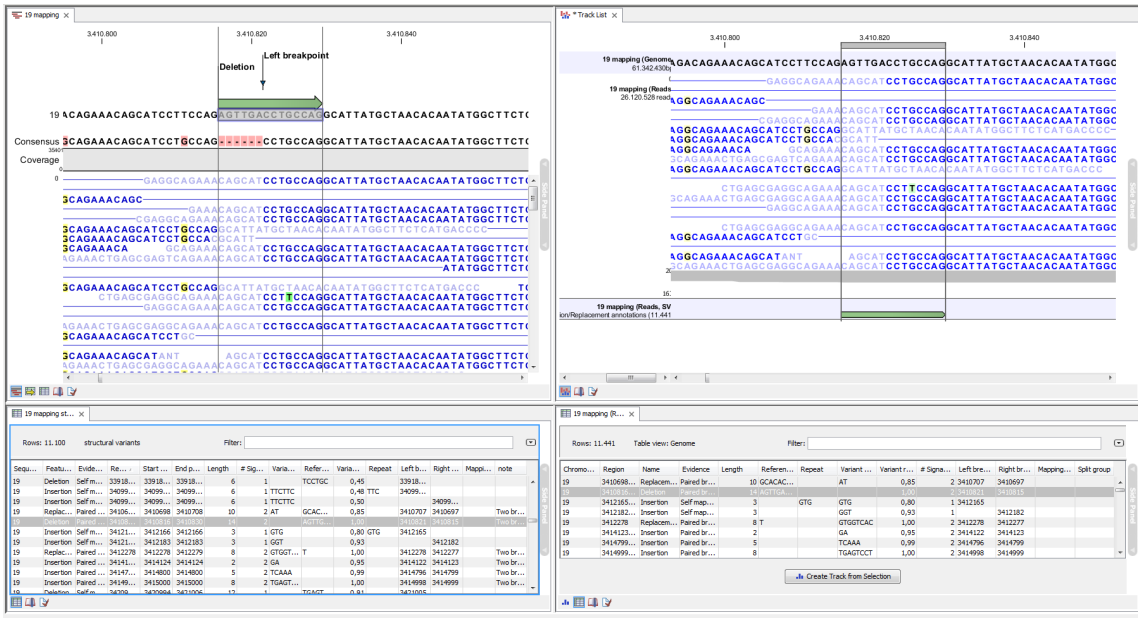



Figure 27.35: *Example of the result of an analysis on a standalone read mapping (to the left) and on a reads track (to the right).*

### 27.10.2 The Structural Variants and InDels output

#### The 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.
- A distribution of the logarithm of the sequence complexity of the unaligned ends of the left and right breakpoint signatures (see Section 27.10.7 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 Breakpoints track (BP):

The breakpoints 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. 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. 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 27.10.7 for how the sequence complexity is calculated).
- 'Reads': The number of reads supporting the breakpoint.

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 variants track (InDel):

The Indel variants track contains a row for each of the called insertions or deletions that *fulfills the requirements for being of a 'variant' type* (see section 27.11.3 for a description of the variant

types "Insertion" and "Deletion"). These are the small to medium sized insertions and deletions (up to 200 bp in length) for which the algorithm was able to identify the allele sequence (that is, the exact inserted sequence, or the exact 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 Variants 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 27.11.1). In addition the following information, which is primarily intended to allow the user to assess the degree of evidence supporting each predicted InDel, is provided:

- '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 breakpoints 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.
- 'Sequence complexity': The sequence complexity of the unaligned end of the breakpoint (see Section 27.10.7). InDels with higher complexity are typically more reliable than those with low complexity.

The 'Zygosity' field is set to 'Homozygous' if the 'Variant ratio' is 0.80 or above, and 'Heterozygous' otherwise.

**The Structural variants track (SV):**

The Structural variants track contains a row for each of the called Structural variants that is 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 ('that is, 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 27.10.6). 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 27.10.7).
- '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.

### 27.10.3 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.

The two steps of the algorithm are described in detail in sections 27.10.4 and 27.10.5.

### 27.10.4 The InDels and Structural Variants detection algorithm - 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 27.36. 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.

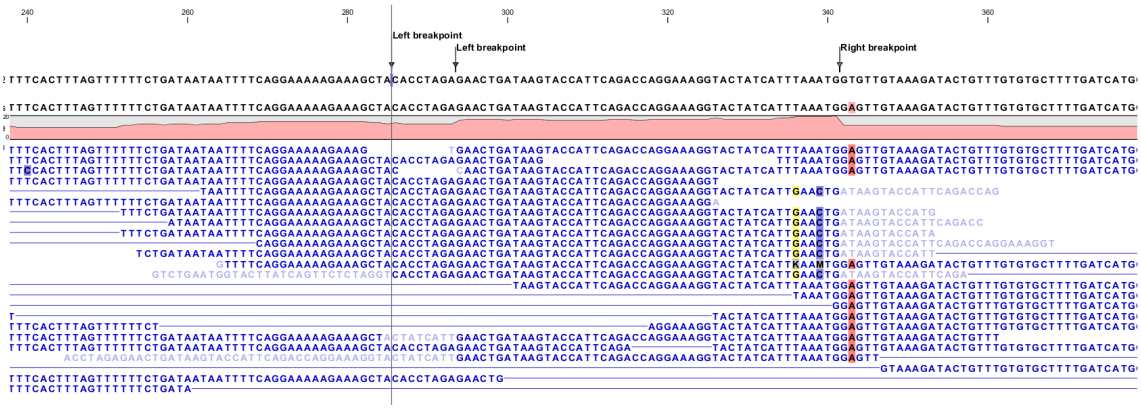



Figure 27.36: *Example of a read mapping containing unaligned ends with three unaligned end signatures.*

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 27.10.1 and Figure 27.33). 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 27.36, 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).

### 27.10.5 The InDels and Structural Variants detection algorithm - 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 27.10.6).

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.

### 27.10.6 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 Indel and Structural variant tool are drawn in Figures 27.37, 27.38, 27.39, 27.40, 27.41, 27.42, 27.43, 27.44 and 27.45.

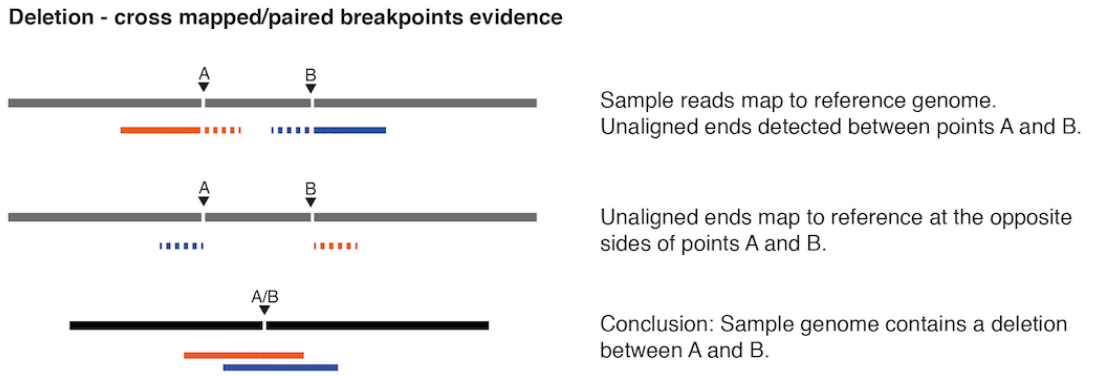



Figure 27.37: *A deletion with cross-mapping breakpoint evidence.*

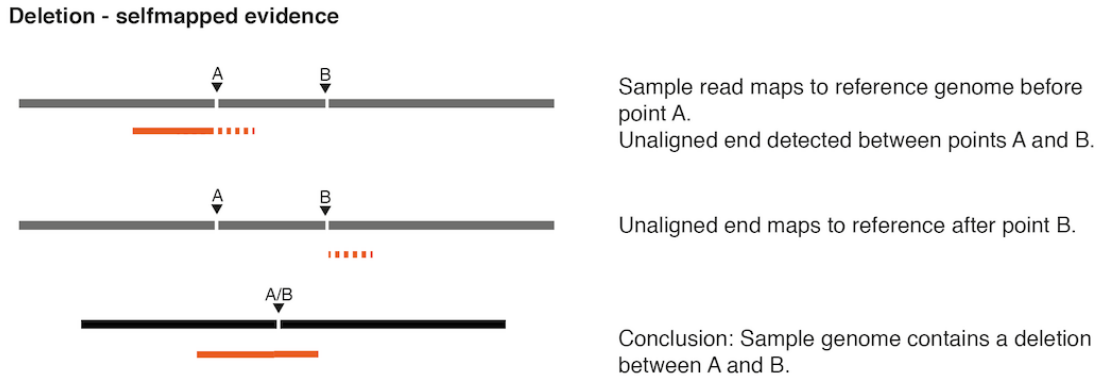



Figure 27.38: *A deletion with selfmapping breakpoint evidence.*

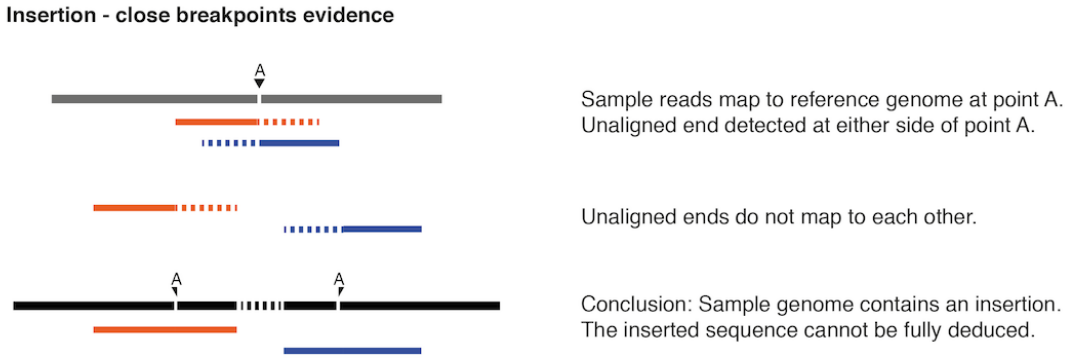



Figure 27.39: *An insertion with close breakpoint evidence.*

### 27.10.7 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

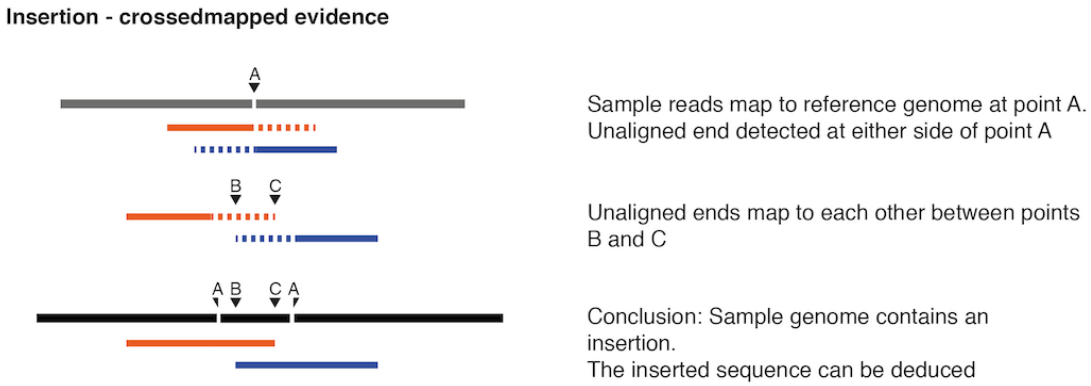



Figure 27.40: *An insertion with cross-mapped breakpoints evidence.*

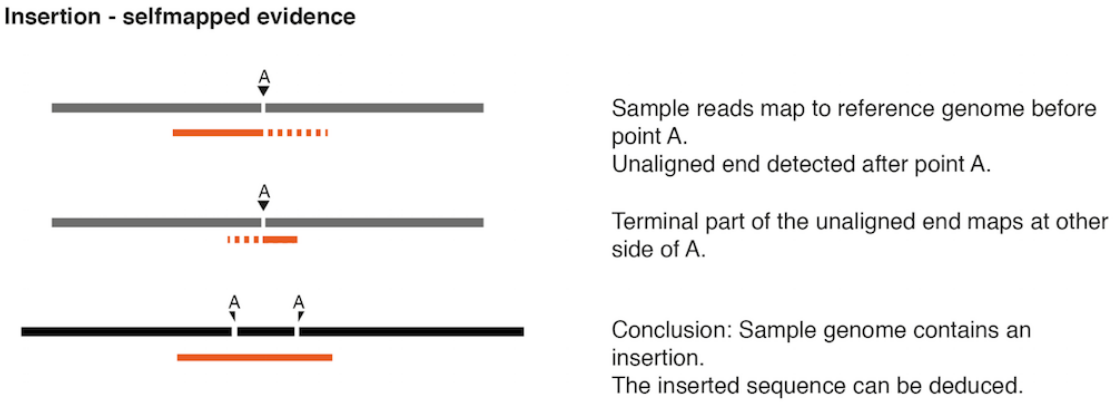



Figure 27.41: *An insertion with selfmapped breakpoint evidence.*

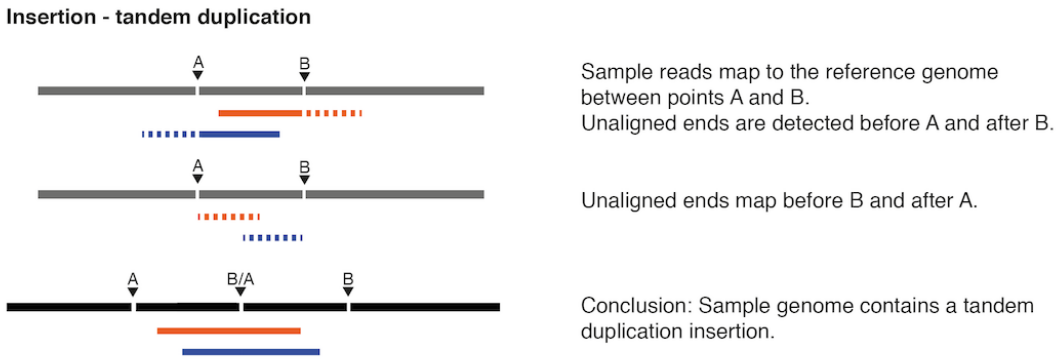



Figure 27.42: *An insertion with breakpoint mapping evidence corresponding to a 'Tandem duplication'.*

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

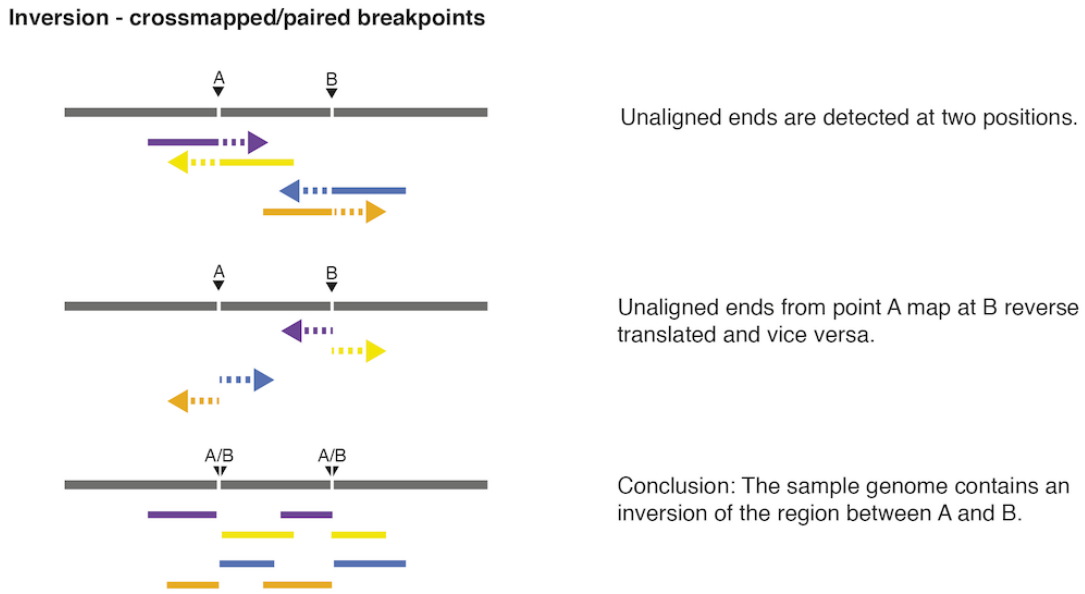



Figure 27.43: *The unaligned end mapping pattern of an inversion.*

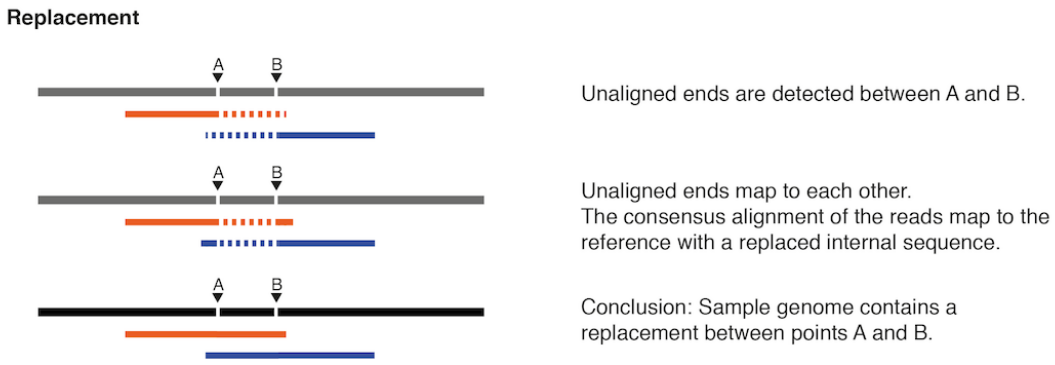



Figure 27.44: *The unaligned end mapping pattern of a replacement.*

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’ )

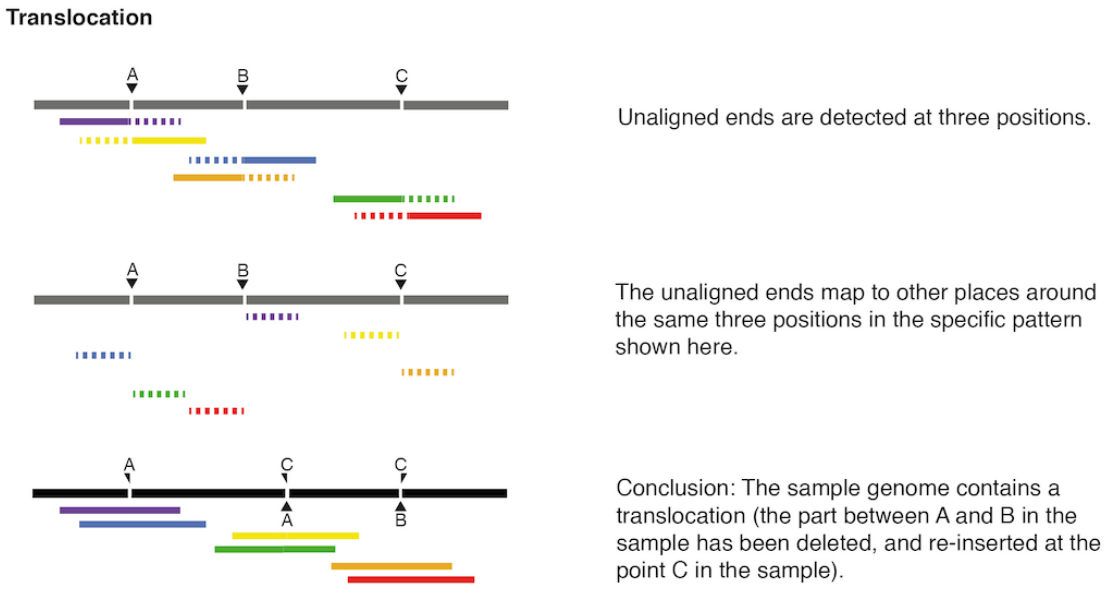



Figure 27.45: *The unaligned end mapping pattern of a translocation.*

- 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 lengths 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 length from 1 to 7 nucleotides and dividing them by the maximum possible number of words for each word length 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

## 27.11 Variant data

Variant data may be obtained either by importing variants from files (e.g. gvf or vcf files - as described in section 6.2), by downloading variants from external databases (e.g. dbSNP, HapMap, 1000genomes or COSMIC - (described in section 6.2)) or by calling variants on read tracks or read mappings using the CLC Basic Variant Detection (section 27.3), Fixed Ploidy Variant Detection (section 27.4), or the Low Frequency Variant Detection (section 27.5) tools.

Variant types include SNVs, MNVs, insertions, deletions or replacements. They may be presented either in a variant track (see figure 27.46) or in an annotated variant table (see figure 27.49).

### 27.11.1 Variant tracks

A variant track (figure 27.46), created with the *CLC Genomics Workbench* variant callers (see section 27.2), has the following information for each 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 27.47. An extract of a gvf-file giving rise to these three variants after import is shown in figure 27.48.

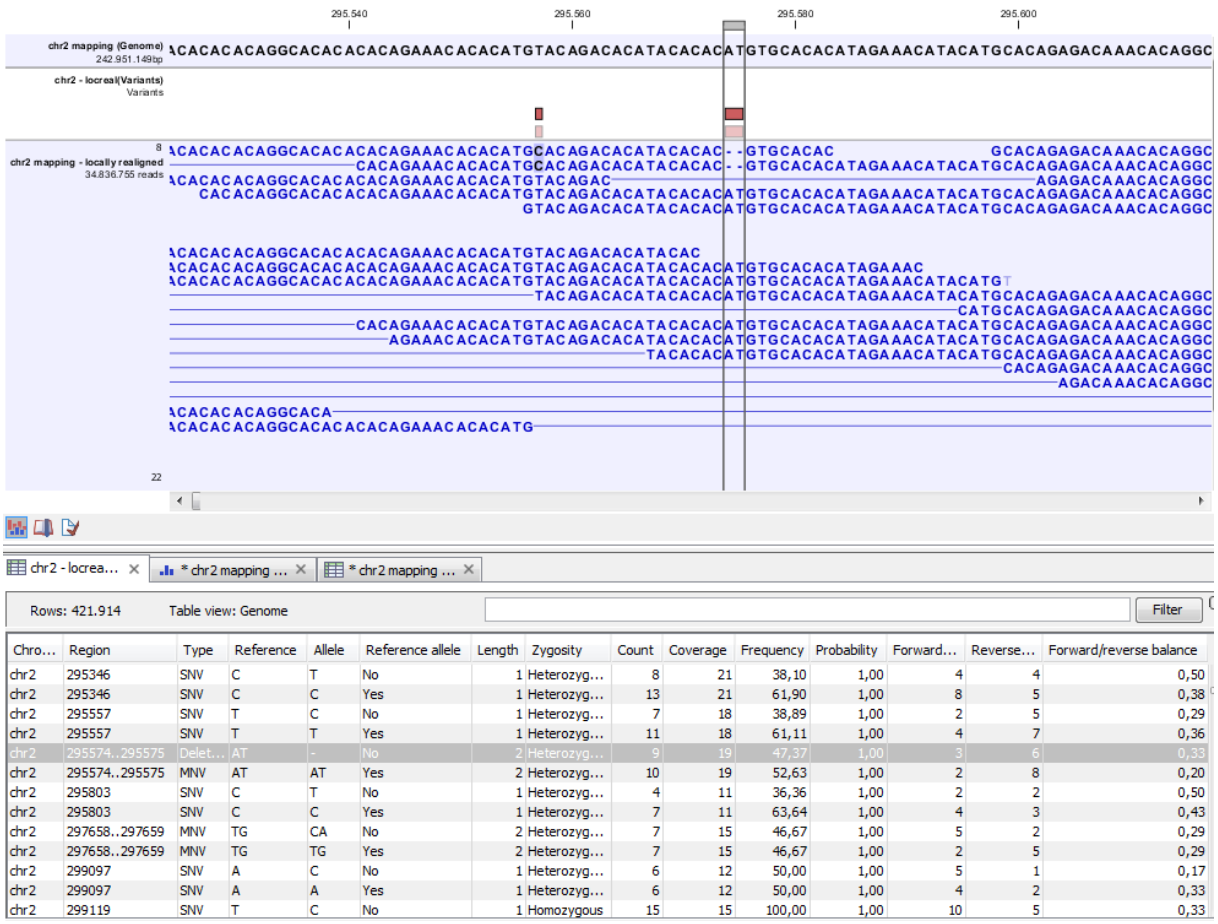
**Variant type** The type of variant. This can either be SNV (single-nucleotide variant), MNV (multi-nucleotide variant), insertion, deletion, or replacement. Learn more in section 27.11.3.

**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 caller 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 caller 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




| 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 27.46: *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 Probabilistic Variant Caller on the read 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.*

replacements, both the length of the replaced reference sequence and the length of the inserted sequence are considered, and the longest of those two is reported.

**Zygosity** The zygosity of the variant called, as determined by the variant caller. 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 caller 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'.

**Coverage** The read coverage at this position. Only 'countable' reads are considered (see under 'Count' above for an explanation of 'countable' reads. Also see overlapping pairs in section 27.12 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 27.13.5 for a description of how to remove low-frequency variants.

**Probability** The probability that this particular variant exists in the sample. (For further information please refer to the White paper on Probabilistic Variant Caller: http://www.clcbio.com/files/whitepapers/whitepaper-probabilistic-variant-caller-1.pdf).

**Forward read count** The number of 'countable' forward reads supporting the allele (see under 'Count' above for an explanation of 'countable' reads). Also see more information about overlapping pairs in section 27.12.

**Reverse read count** The number of 'countable' reverse reads supporting the allele (see under 'Count' above for an explanation of 'countable' reads). Also see more information about overlapping pairs in section 27.12.

**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).[1]

**Average quality** The average read quality score of the bases supporting a variant.See section 27.13.5 on how to remove variants that have a low average quality. 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). For deletions, the quality scores of the two surrounding bases are taken into account, and the lowest value of these two is reported.

**Hyper-allelic** Relevant for "Quality-based Variant Detection". Reports hyper-allelic status of variants based on the specified threshold "Maximum expected allele" in the "Set genome information" wizard under "Ploidy". The output in the table is "Yes" or "No" with respect to whether the threshold has been exceeded.

Variant tracks that have been created with Genomics Workbench 6.0 will have an additional column with the header 'Linkage'. See section **??** for details.

Please note that the variants in the variant track can be enriched with information using the annotation tools in section 27.13.

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 27.47 is given in figure 27.48.

### 27.11.2 The annotated variant table

The annotated variant table (see figure 27.49) contains a subset of the columns of the variant track table and additionally the three columns below.

[1] 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 (see [Nguyen et al., 2011] for a recent study with Illumina data). 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 27.12.)

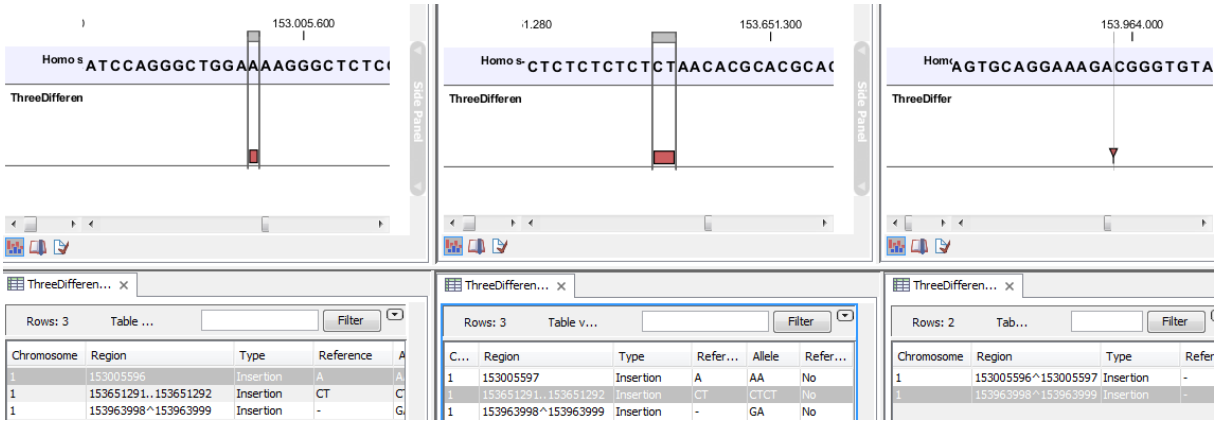



Figure 27.47: *Examples of variants with different types of 'Region' column contents. The left-most variant has a 'single position' region, the middle variant has a 'region' region and the right-most has a 'between positions' region.*

```
##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 27.48: *A gvf file giving rise to the variants in the figure above.*

| 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 27.49: *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 e.g. 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://www.hgvs.org/mutnomen/`.

**Amino acid change** If the reference sequence of the mapping is annotated with ORF or CDS annotations, the variant caller 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://www.hgvs.org/mutnomen/`.

If the reference sequence has no gene and cds annotations these columns will have the entry

'NA'. Note that 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 27.13).

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 2.1.6), 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 27.11.1).

### 27.11.3 Variant types

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.

**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.

### 27.11.4 Special notes upgrading to Genomics Workbench 7.5

This section is a special note on upgrading to CLC Genomics Workbench 7.5 and CLC Genomics Server 6.5. This is intended for those upgrading from earlier versions and will provide information about how this change affects both existing and new data.

With the new version we have introduced three new variant callers: "Basic Variant Detection", "Fixed Ploidy Variant Detection", and "Low Frequency Variant Detection" (see section 27.2). The Basic Variant Detection and Fixed Ploidy Variant Detection are similar in nature to the Quality-based and Probabilistic Variant Detection tools respectively. The main difference is that

all filters, previously employed in either the Quality-based or Probabilistic Variant Detector, are now available in all three variant callers, in addition to a new filter: the relative read direction filter. The changes relating to the filters have some few consequences, and care must be taken when attempting to run the new variant callers with parameters that are equivalent to those used in the old ones (see figure 27.51) for details about the Fixed Ploidy Variant Detection and figure 27.50 for details about the Basic Variant Detection).

In particular:

- Regions with coverage above a certain threshold can be masked out in the new callers. To avoid this, you should set the 'Maximum coverage' parameter as high as the maximum coverage observed in the read mapping.
- A parameter (maximum number of gaps and mismatches) has been deleted from the quality filter. This parameter made the algorithm ignore reads with more than the specified number of nucleotide differences to the reference, and had the unfortunate effect that read aligned across longer deletions or insertions would not be considered, which in turn meant that these would be missed. The fact that this parameter has been dropped will imply that you cannot expect to get the same variant calls with the basic variant caller as with the quality-based variant caller, unless the value in the Quality-based variant caller has been set to the maximum read length.
- The Fixed Ploidy Variant Detection algorithm has a 'minimum frequency' parameter that will remove variants with a frequency below the specified threshold. In order for that parameter to have no effect you should set it to zero.
- The Fixed Ploidy variant caller by default employs the new 'Relative read direction filter'. This filter is less stringent than the 'read direction filter' which by default is employed by the old Probabilistic variant caller. To obtain similar filtering settings related to read direction, you should either employ none of the read direction filters in the two variant callers, or employ the 'read direction filter' (with the frequency parameter specified to 5% in the Fixed ploidy) in both of them.
- The Basic variant caller by default employs the new 'Relative read direction filter'. This filter is less stringent than the 'read direction filter' which by default is employed by the old Quality-based variant caller. To obtain similar filtering settings related to read direction, you should either employ none of the read direction filters in the two variant callers, or employ the 'read direction filter' (with the frequency parameter specified to 5% in the Basic variant caller) in both of them.

The table shown in figure 27.50 contains a comparison of the Basic Variant Detection tool with the Quality-based Variant Detection tool, whereas the table in figure 27.51 shows a comparison of the Fixed Ploidy Variant Detection tool with the Probabilistic Variant Detection tool.

## 27.12 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

| Basic Variant Detection parameter corresponding to Quality-based Variant detection parameter | Basic Variant Detection parameters with no equivalent in Quality-based Variant Detection tool | Basic Variant Detection default setting | Basic Variant Detection value corresponding to Quality-based Variant Detection default setting | Quality-based Variant Detection parameter name | Quality-based Variant Detection default setting | Quality-based Variant Detection parameters with no equivalent in Basic Variant Detection |
|---|---|---|---|---|---|---|
| Base quality filter | | Off (Radius: 5, neighborhood quality: 15, central quality: 20) | On (Radius: 5, neighborhood quality: 15, central quality: 20) | Quality | On (Radius: 5, neighborhood quality: 15, central quality: 20) | **Gap mismatch count (default 2. Value to get Basic equivalent : Inf (100000)).** |
| Ignore non-specific mathces 'Reads' | | On | On | Ignore non-specific matches | On | - |
| Ignore broken pairs | | On | On | Ignore broken pairs | On | - |
| Minimum coverage | | 10 | 10 | Minimum coverage | 10 | - |
| Minimum frequency | | 20 | 35 | Minimum variant frequency | 35 | - |
| Ignore positions with coverage above | | On (When on: Max coverage 100,000, Required variant count (=Minimum count): 2) | Off | Advanced | Off (When on: Max coverage: 99999, Required variant count: 1) | **Sufficient variant count (default 5. Value to get Basic equivalent: Inf (Max coverage)** |
| Read direction filter | | Off (When on: 5) | On | Require presence in both forward and reverse reads | On (When on: only variants with at least 5 percent of reads in either direction carrying the variant is required) | |
| Ignore non-specific mathces 'Regions' | | Off | Off | Ignore variants in non-specific regions | Off (When on, all variants covered by at least one non-specific reads are ignored) | |
| Remove pyroerror variants | | Off (When on: In hom.pol.reg minimum length 3, With frequency below: 0.8 | Off | Filter 454/Ion homopolymer reads | Off (When on: Indels found in homopolymer regions are ignored) | |
| Ploidy | | 2 | - | Ploidy | 2 | |
| | Relative read direction filter | On | Off | | | |
| | Read position filter | Off (When on: 1%) | Off | | | |

Figure 27.50: *Comparison of the Basic Variant Detection tool with the Quality-based Variant Detection tool.*

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 quality-based 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.

## 27.13 Annotate and filter variants

In addition to the general filter for track tables, including the ability to create a new track from a selection (see section 24.1.3), there are a number of tools for general filtering and annotation of variants (for functional annotation and filtering, see section 27.15).

| Fixed Ploidy Variant Detection parameter name | Fixed Ploidy Variant Detection parameters with no equivalent in Probabilistic Variant Detection tool | Fixed Ploidy Variant Detection default setting | Fixed Ploidy Variant Detection value corresponding to Probabilistic Variant Detection default setting | Probabilistic Variant Detection parameter name | Probabilistic Variant Detection default setting |
|---|---|---|---|---|---|
| Ignore non-specific mathces 'Reads' | | On | On | Ignore non-specific matches | On |
| Ignore broken pairs | | On | On | Ignore broken pairs | On |
| Minimum coverage | | 10 | 10 | Minimum coverage | 10 |
| Required variant probability | | 90 | 90 | Variant probability | 90 |
| | Minimum frequency | 20 | 0 | | |
| Read direction filter | | Off (When on: 5) | On | Require presence in both forward and reverse reads | On (When on: only variants with at least 5% of reads in either direction carrying the variant is required) |
| Ignore non-specific mathces 'Regions' | | Off | Off | Ignore variants in non-specific regions | Off |
| Remove pyroerror variants | | Off | Off | Filter 454/Ion homopolymer reads | Off |
| | Minimum count | 2 | 2 | Required variant count | 2 |
| Ploidy | | 2 | 2 | Ploidy | 2 |
| | Ignore positions with coverage above | 100,000 | The maximum coverage in the read mapping | | |
| | Restrict calling to target regions | Off | Off | | |
| | Base quality filter | Off | Off | | |
| | Relative read direction filter | On | Off | | |
| | Read position filter | Off | Off | | |

Figure 27.51: *Comparison of the Fixed Ploidy Variant Detection tool with the Probabilistic Variant Detection tool.*

### 27.13.1 Filter against known variants

Comparison with known variants from variant databases is a key concept when working with resequencing data. The *CLC Genomics Workbench* provides two tools for facilitating this task: one for *annotating* your experimental variants with information from known variants (e.g. adding information about phenotypes like cancer associated with a certain variant allele), and one for *filtering* your experimental variants based on this information (e.g. for removing common variants). The first tool is explained in the next section, while this section explains the latter.

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 11.4). **Please note that there is also a plugin for annotating with data from HGMD and other databases via Biobase Genome Trax: http://www.clcbio.com/clc-plugin/biobase-genome-trax/**.

This section will use the filter tool as an example, since the core of the tools are the same:

**Toolbox** | **Resequencing** (  ) | **Annotate and Filter** | **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 27.52

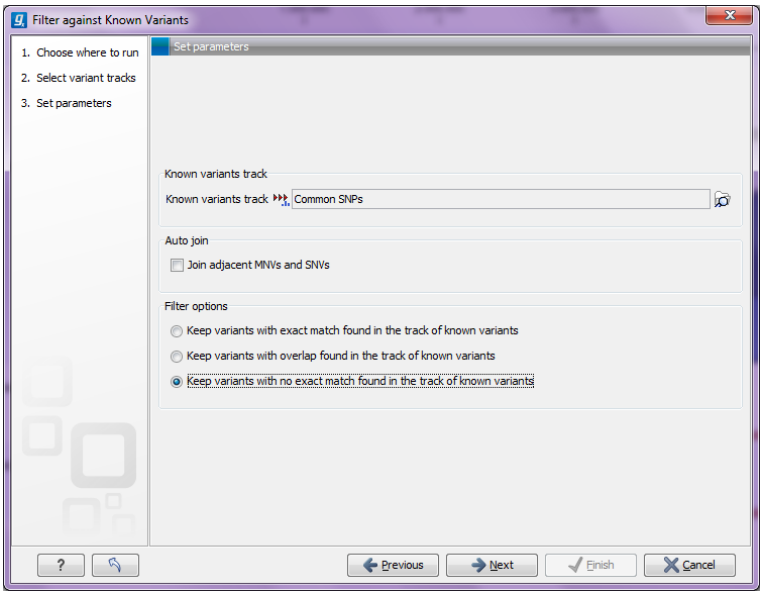



Figure 27.52: *Specifying a variant track to filter against.*

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. There are three modes of filtering:

**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.

### 27.13.2 Annotating from known variants

Section 27.13.1 describes how to filter against known variants, but the *CLC Genomics Workbench* also includes a tool to annotate from known variants:

To run the Annotating from known variants tool, go to:

**Toolbox** | **Resequencing** (  ) | **Annotate and Filter** | **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.

### 27.13.3 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 27.53).

When there are multiple isoforms, a comma-separated list of the exon numbers is given.

### 27.13.4 Annotate with flanking sequence

In some situations, it is useful to see a variant in the context of the bases of the reference sequence. This information can be added using the **Annotate with Flanking Sequence** tool:

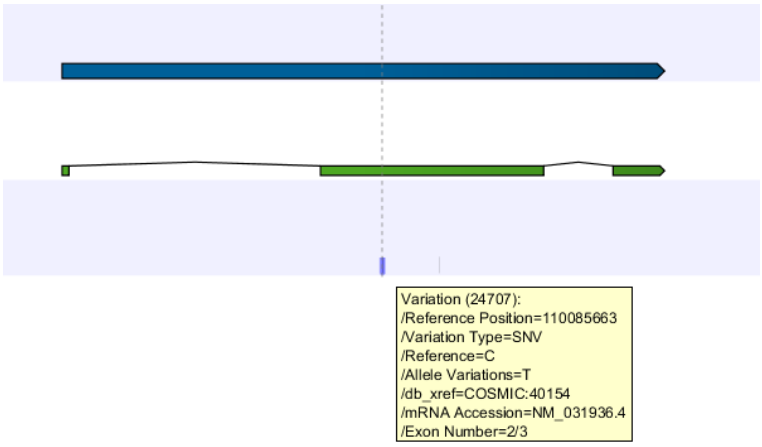



Figure 27.53: *A variant found in the second exon out of three in total.*

**Toolbox** | **Resequencing** ( ) | **Annotate and Filter** | **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 27.54

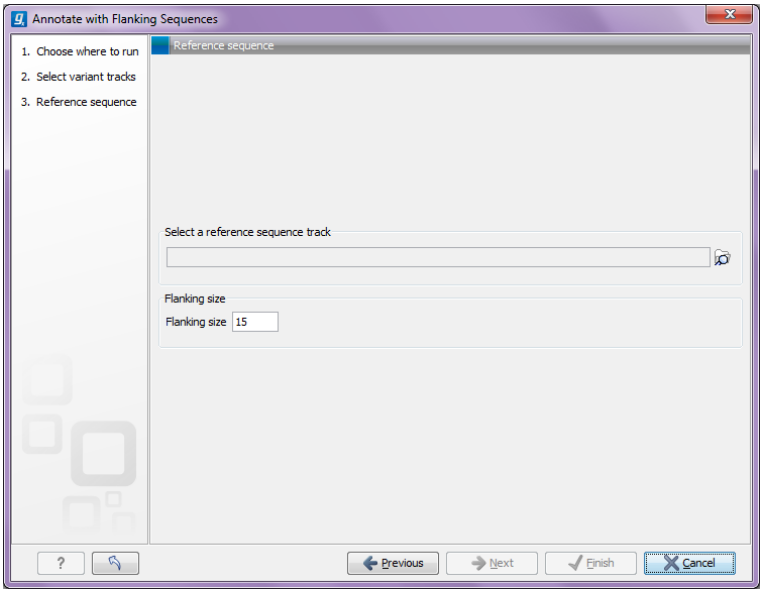



Figure 27.54: *Specifying a reference sequence and the amount of flanking bases to include.*

Select a sequence track that should be used for adding the flanking sequence, and specify how large the flanking region should be.

The result will be a new track with an additional column for the flanking sequence formatted like this: `CGGCT[T]AGTCC` with the base in square brackets being the variant allele.

### 27.13.5 Filter marginal variant calls

Variant calling is always a balance between sensitivity and specificity. To get rid of potential false positive variants, you can use this tool on a variant track to remove some of the variant calls, which are supported by only low quality bases, have low frequency or a skewed forward-reverse reads balance. In this way, you can try different strategies for filtering without re-running the variant detection.

**Toolbox** | **Resequencing** ( ) | **Annotate and Filter** | **Filter Marginal Variant Calls**

This opens a dialog where you can select a variant track ( ) with experimental data that should be filtered.

Click **Next** to set the filtering thresholds as shown in figure 27.55

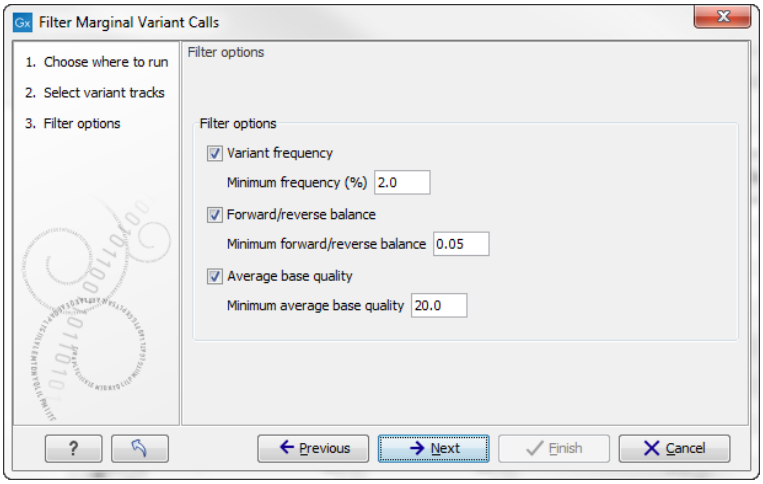



Figure 27.55: *Specifying thresholds for filtering.*

The following thresholds can be specified. All alleles except the reference allele are investigated separately, but in order to remove a variant, all non-reference alleles have to fulfill the requirements.

**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 27.11.1.

The result is a new track where all variants (or at least one non-reference allele of the variant) fulfill the criteria.

### 27.13.6 Filter reference variants

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

**Toolbox** | **Resequencing** ( ) | **Annotate and Filter** | **Filter Reference Variants**

This opens a dialog where you can select a variant track ( ) that should be filtered.

Clicking **Next** and **Finish** to create a new track without the reference variants.

## 27.14 Comparing variants

In the toolbox, the folder **Compare Variants** contains tools that can be used to compare experimental variants. The two tools **Compare Sample Variant Tracks** and **Compare Variants within Group** are similar to the **Filter against Known Variants** found in the **Annotate and Filter Variants** 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 other tools when comparing experimental variants with other experimental variants.

### 27.14.1 Compare variants within group

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 patients with the same disease and would like to identify variants that are present in at least 70% of all patients. It can also be used to do an overall comparison between samples (a frequency threshold of 0% will report all alleles).

**Toolbox** | **Resequencing** ( ) | **Compare Variants** | **Compare Variants within Group**

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 27.56.

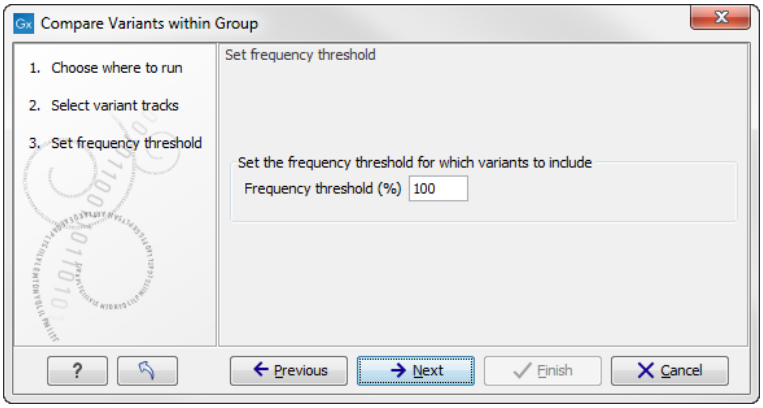



Figure 27.56: *Frequency treshold.*

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** The number of samples that have the variant

**Total number of samples** The total number of samples (this will be identical for all variants).

**Sample frequency** This is the same frequency that is also used as a threshold (see figure 27.56).

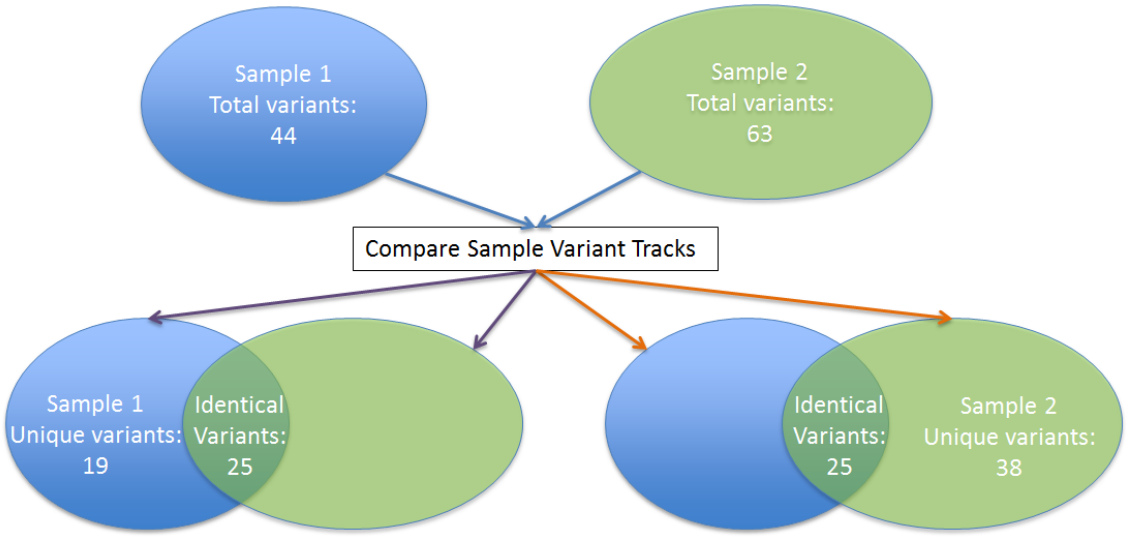
**Origin tracks** A comma-separated list of the name of the tracks that contain the variant.

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.

### 27.14.2 Compare sample variants

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 27.57.




*Variant track of (Sample 1) and Comparison track for filtering (Sample 2)*
- Keep variants that are the same → 25 variants
- Keep variants that are different → 19 variants

*Variant track of (Sample 2) and Comparison track for filtering (Sample 1)*
- Keep variants that are the same → 25 variants
- Keep variants that are different → 38 variants

Figure 27.57: *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:

**Toolbox** | **Resequencing** (  ) | **Compare Variants** | **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 27.58.

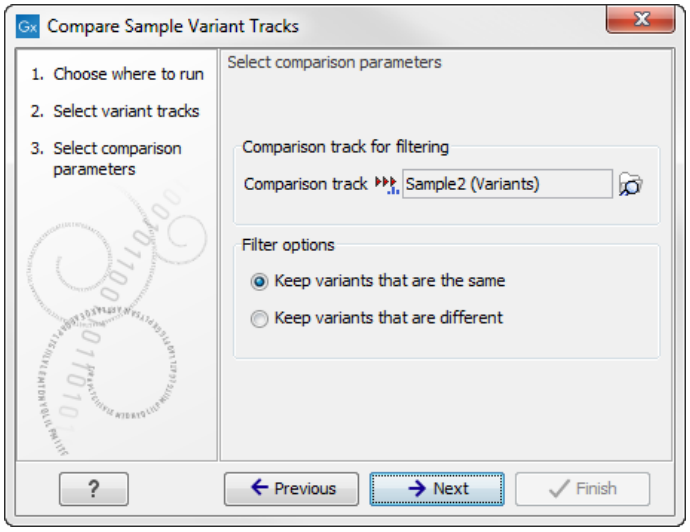



Figure 27.58: *Comparing against variants in "Sample 2".*

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.

### 27.14.3 Fisher exact test

This tool should be used if you have a case-control study. This could be patients with a disease (case) and healthy individuals (control). The Fisher Exact Test will identify variants that are significantly more common in the case samples than in the control samples.

**Toolbox** | **Resequencing** (  ) | **Compare Variants** | **Fisher Exact Test**

In the first step of the dialog, you select the case variant tracks. Clicking **Next** shows the dialog in figure 27.59.

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 27.60), reporting only those variants with p-values lower than the threshold. All corrected and uncorrected p-values are shown here, so alternatively, variants with non-significant p-values can also be filtered out or more stringent thresholds can be applied at this stage, using the manual filtering options.

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

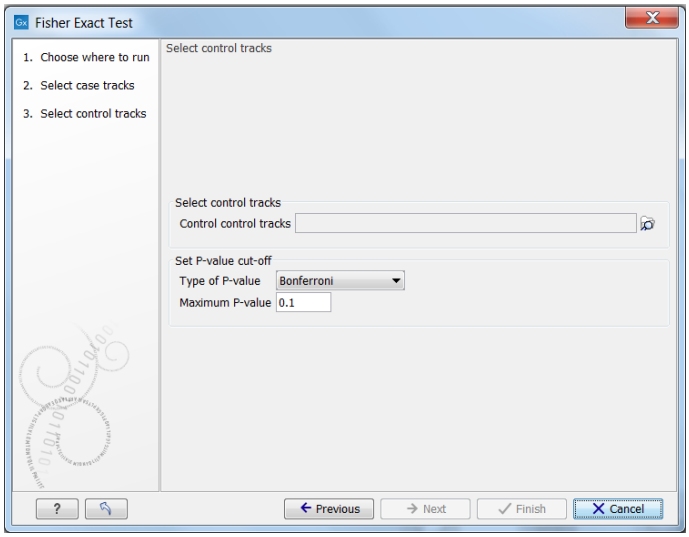



Figure 27.59: *In this dialog you can select the control tracks, a p-value correction method, and specify the p-value threshold for the fisher exact test.*

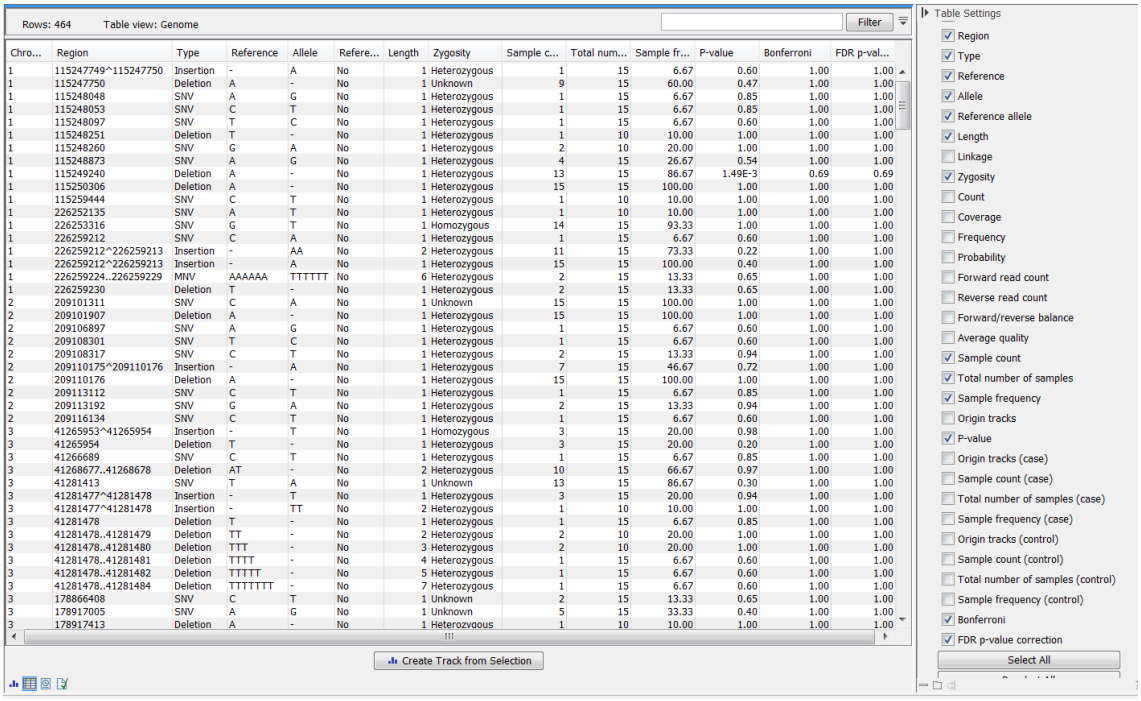



Figure 27.60: *In the output table, you can view information about all significant variants, select which columns to view, and filter manually on certain criteria.*

'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.

### 27.14.4 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** (  ) | **Compare Variants** | **Trio Analysis**

In the first step of the dialog, select the variant track of the child. Clicking **Next** shows the dialog in figure 27.61.

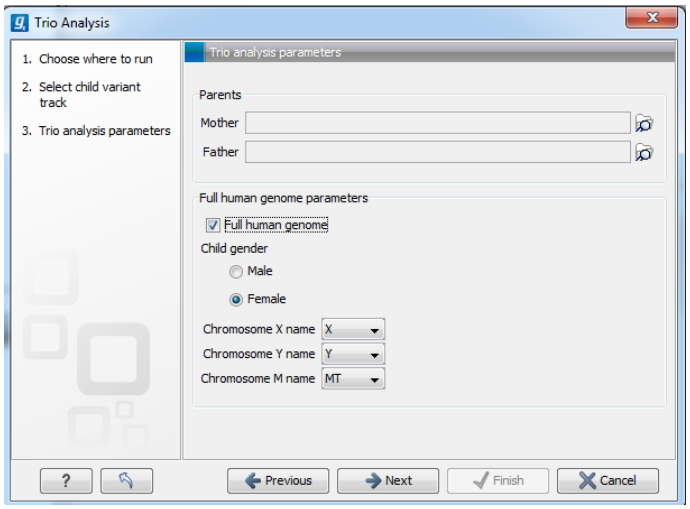



Figure 27.61: *Selecting variant tracks of the parents.*

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 27.62).

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'.

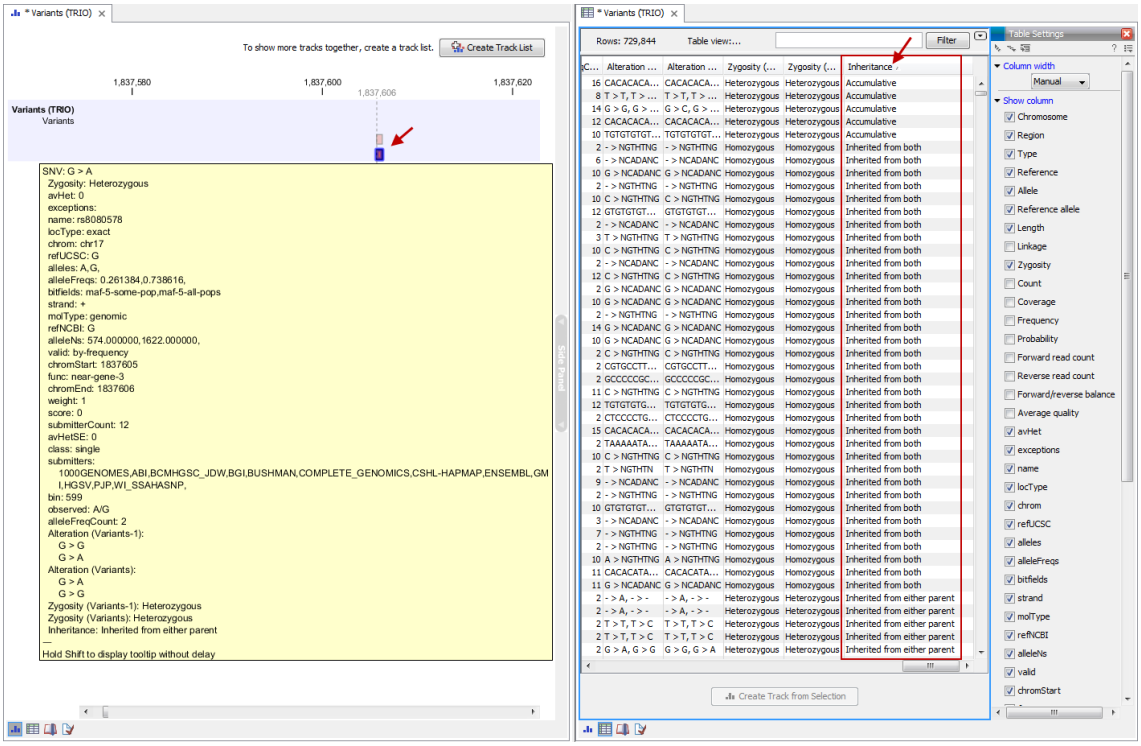



Figure 27.62: *Output from Trio Analysis showing the variants found in the child in track and table format.*

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 'accumulative' mutation. Accumulative mutations are present in a heterozygous state in each of the parents, but are homozygous in the child. To investigate potential disease relevant variants, 'accumulative' 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.

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.

Let's look at an example where these special rules apply - in this case 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 caller. 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 caller. 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 caller. 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: 'De novo', 'Accumulative', 'Inherited from both', 'Inherited from either', 'Inherited from (Name of parent track)'.

**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.

### 27.14.5 Filter Against Control Reads

Running the variant caller 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 Filter Against 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** () | **Compare Variants** | **Filter against 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 27.63). 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 27.63 the threshold is set at two. This means that if a variant is found in two or more of the control reads, it will be filtered away.

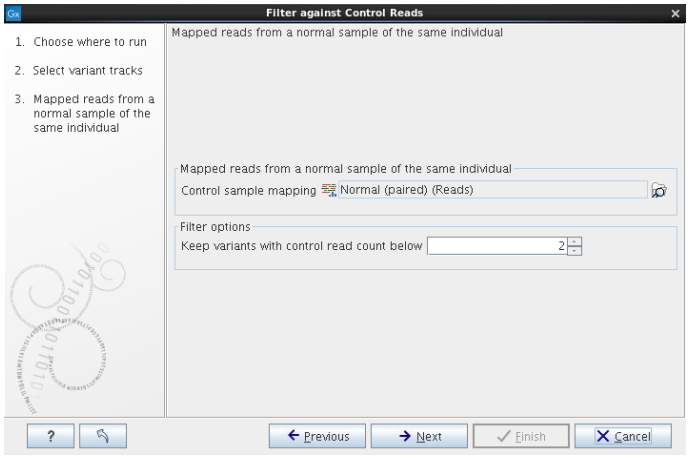



Figure 27.63: *The control reads data set.*

## 27.15 Predicting 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.

### 27.15.1 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 27.64).

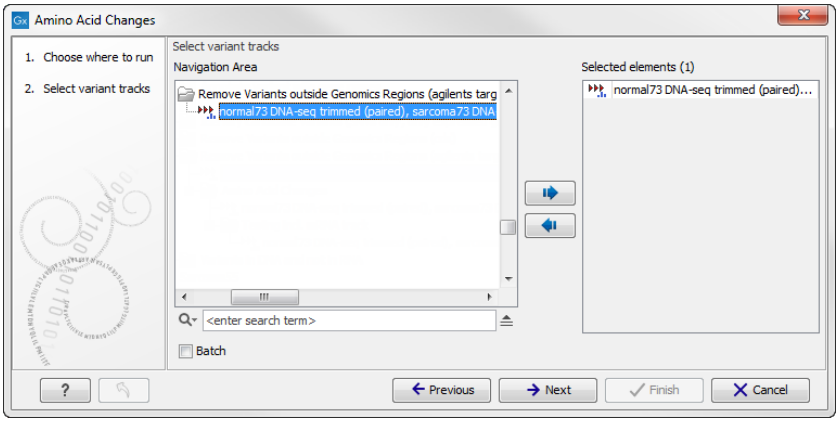



Figure 27.64: *The amino acid changes annotation tool takes variant tracks as input.*

Click on the button labeled **Next** to go to the next wizard step (figure 27.65).

In this step you get the following options:

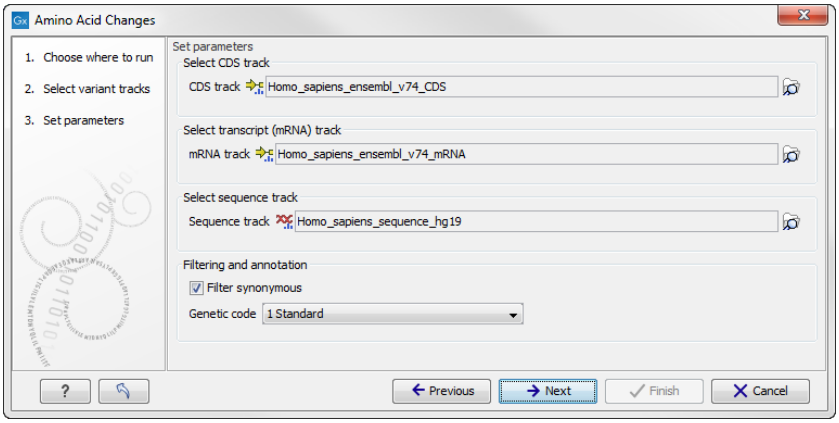



Figure 27.65: *Select CDS, mRNA, and sequence track and choose whether or not you would like to filter away synonymous variants.*

- **Select CDS track** Click on the folder icon ( ) in the right side of the wizard to select a 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 by clicking on the **Download** icon ( ) found in the upper right corner of the Workbench. Select **Download Reference Genome Data ( )** as described in section 11.4.

- **Select mRNA track** Click on the folder icon ( ) in the right side of the wizard to select the mRNA track. The mRNA track is used to determine whether the variant is inside or outside the region covered by the transcript. The mRNA track is optional. If you choose to provide an mRNA track you 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. If you choose not to provide an mRNA track, variants found outside the CDS region will not be annotated.

- **Select sequence track** Click on the folder icon ( ) in the right side of the wizard to select the reference sequence track.

- **Filtering and annotation** The "Filter synonymous" option allows you to filter away synonymous variants that does not cause any change to the encoded amino acid. 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.

- **Filter CDS regions with no variants** When choosing this option, "empty" CDS regions (CDS regions with no overlapping variants) will be filtered away from the track.

Click on the button labeled **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 27.66.

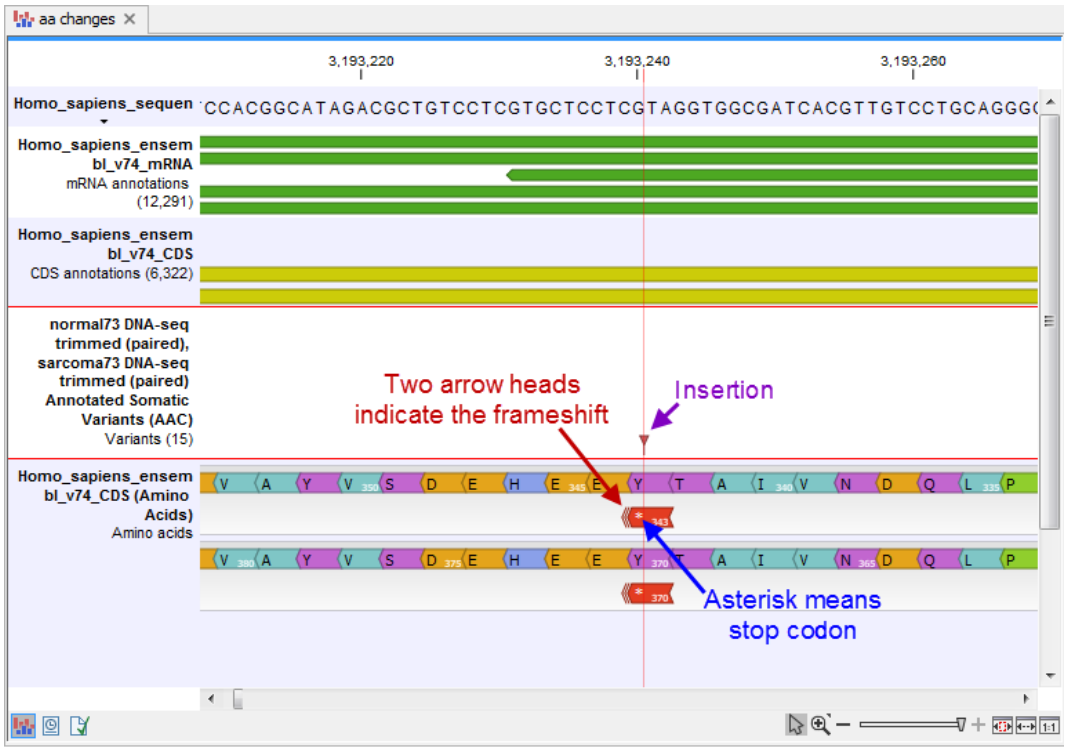



Figure 27.66: *The variant track and the amino acid track is here presented together with the reference sequence and the mRNA and CDS tracks. An insertion (purple arrow) has caused a frameshift (red arrow) that has changed an alanine to a stop codon (blue arrow).*

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://www.hgvs.org/mutnomen/. Variants inside 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 of the nomenclature see the "Recommendations for the description of protein sequence variants (v2.0)" at http://www.hgvs.org/mutnomen/.

- **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 and the amino acid change annotated by the amino acid changes may not be correct in this case.
- **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".

An example of the output is given in figure 27.67.

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 27.68.

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.

**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 27.69).

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




Figure 27.67: *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.*

  - **Turquoise** neutral, nonpolar
  - **Orange** acidic, polar
  - **Blue** basic ,polar
  - **Bright green** other (functional properties)

- **Polarity** Colors the amino acids according to the following categories:
  - **Green** neutral, polar
  - **Black** neutral, nonpolar
  - **Red** acidic, polar
  - **Blue** basic ,polar

- **Rasmol** Colors the amino acids according to the Rasmol color scheme (see http://www.openrasmol.org/doc/rasmol.html).

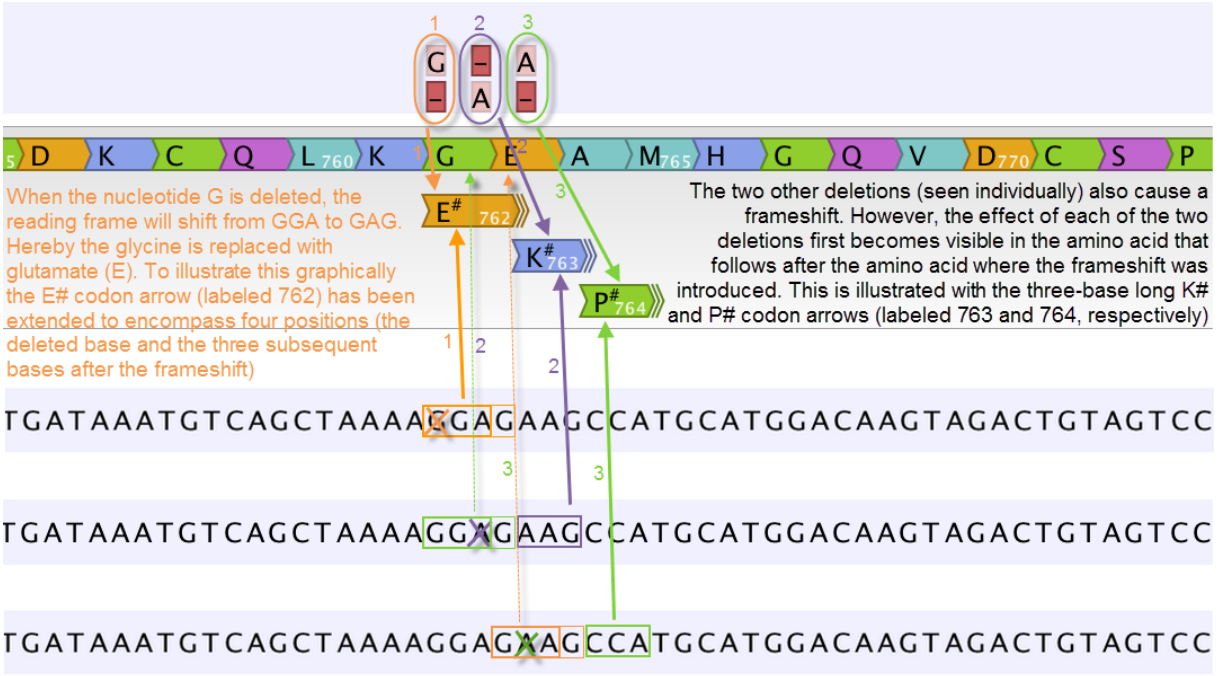



Figure 27.68: *Amino acids encoded from codons that potentially could have been affected by more than one variant are marked with a hash symbol (#) as the graphically presented amino acid changes always only include a single variant (a SNV, MNV, insertion, or deletion). Shown here are three different 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.*

### 27.15.2 Predict splice site effect

This tool will analyze a variant track to determine whether the variants fall within potential splice sites. A transcript track has to be selected as shown in figure 27.70.

If a variant falls within two base pairs of an intron-exon boundary, it will annotated as a possible splice site disruption. As part of the dialog you can choose to exclude all variants that do not fall within a splice site.

### 27.15.3 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.

For this, you need a GO association file, which includes gene names and associated Gene Ontology terms. You can download that from the Gene Ontology web site for different species (`http://www.geneontology.org/GO.downloads.annotations.shtml`). Find *Bos taurus* on the list and double-click on "annotations" (see figure 27.71). Import the downloaded annotations

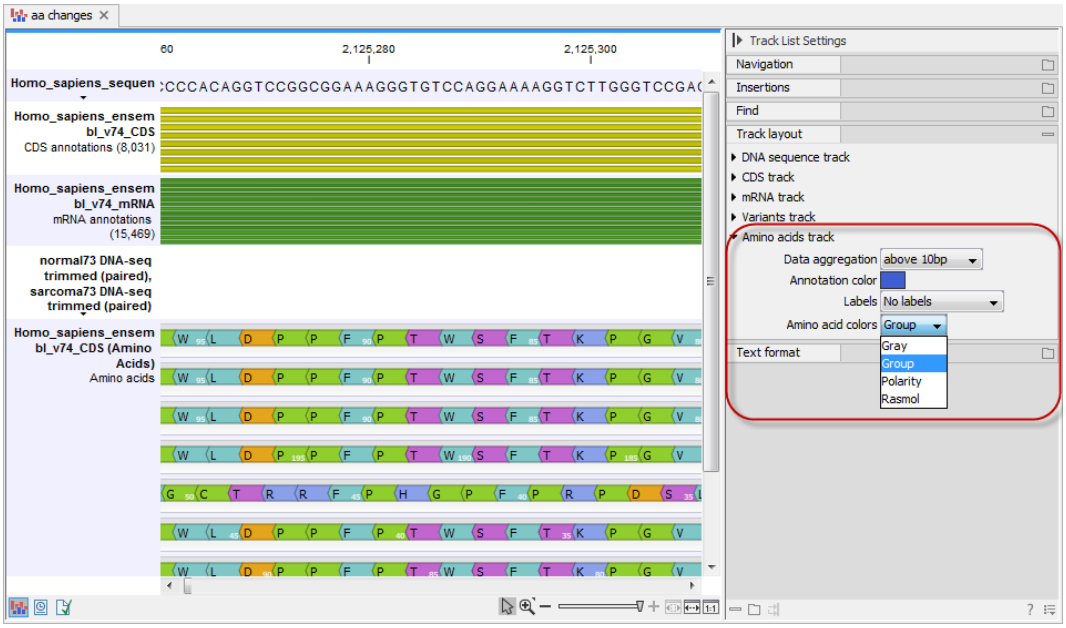



Figure 27.69: *The colors of the amino acids can be changed in the Side Panel under "Amino acids track".*

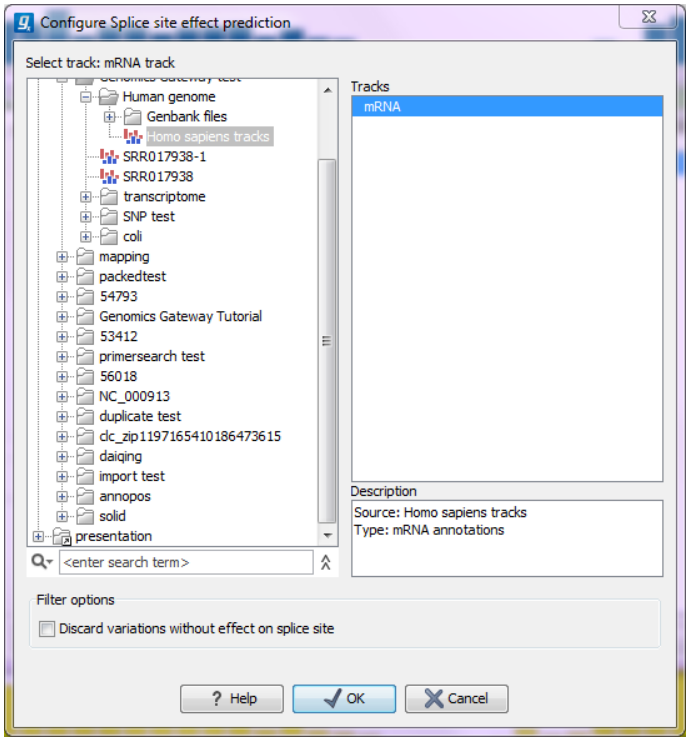


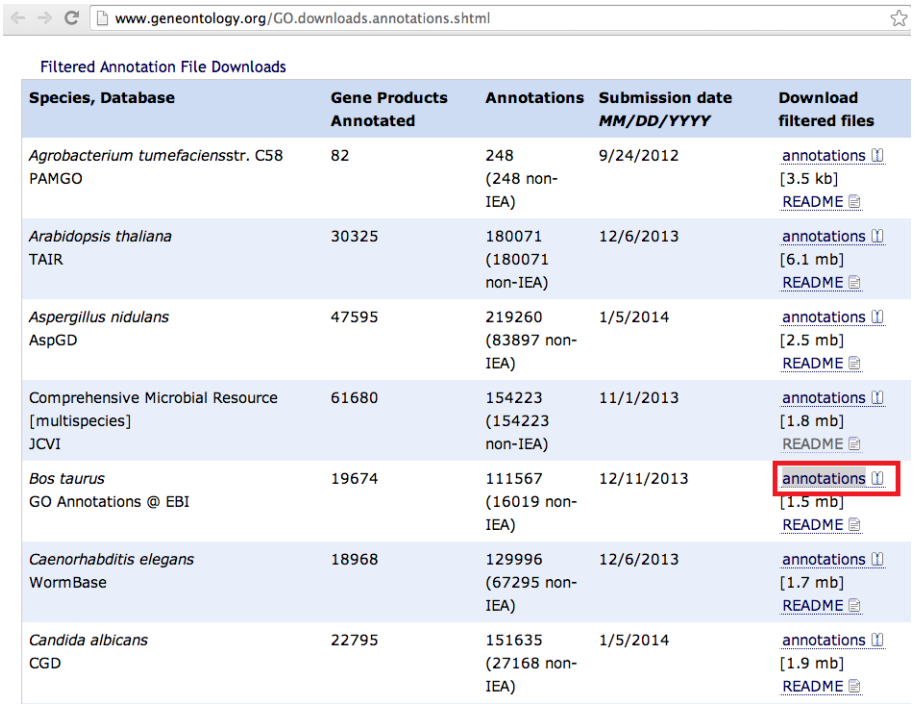

Figure 27.70: *The splice site annotation.*

into the *CLC Genomics Workbench* using "Standard Import".

To run the analysis go to the toolbox:

**Toolbox** | **Resequencing Analysis** ( ) | **Functional Consequences** | **GO Enrichment Analysis**

When you run the GO Enrichment Analysis, you have to specify both the annotation association



www.geneontology.org/GO.downloads.annotations.shtml

Filtered Annotation File Downloads

| Species, Database | Gene Products Annotated | Annotations | Submission date *MM/DD/YYYY* | Download filtered files |
|---|---|---|---|---|
| *Agrobacterium tumefaciens*str. C58 PAMGO | 82 | 248 (248 non-IEA) | 9/24/2012 | annotations [3.5 kb] README |
| *Arabidopsis thaliana* TAIR | 30325 | 180071 (180071 non-IEA) | 12/6/2013 | annotations [6.1 mb] README |
| *Aspergillus nidulans* AspGD | 47595 | 219260 (83897 non-IEA) | 1/5/2014 | annotations [2.5 mb] README |
| Comprehensive Microbial Resource [multispecies] JCVI | 61680 | 154223 (154223 non-IEA) | 11/1/2013 | annotations [1.8 mb] README |
| *Bos taurus* GO Annotations @ EBI | 19674 | 111567 (16019 non-IEA) | 12/11/2013 | annotations [1.5 mb] README |
| *Caenorhabditis elegans* WormBase | 18968 | 129996 (67295 non-IEA) | 12/6/2013 | annotations [1.7 mb] README |
| *Candida albicans* CGD | 22795 | 151635 (27168 non-IEA) | 1/5/2014 | annotations [1.9 mb] README |

Figure 27.71: *Download the GO Annotations from* Bos taurus *by double-clicking on "annotations" and import the downloaded annotations into the* **CLC Genomics Workbench** *using "Standard Import".*

file, a gene track and finally which ontology (cellular component, biological process or molecular function) you would like to test for (see figure 27.72).

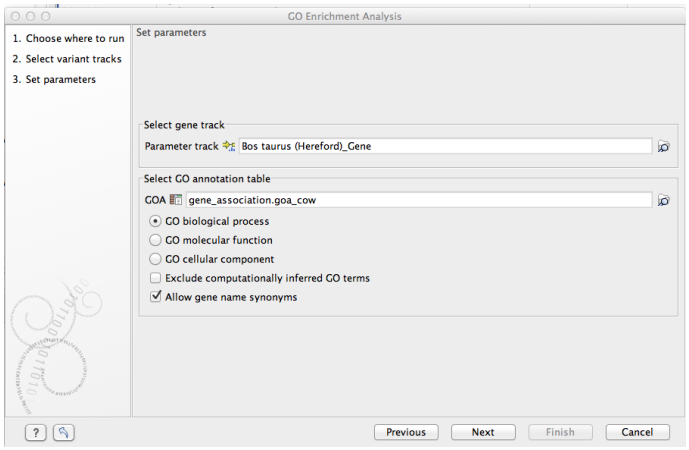



Figure 27.72: *The GO enrichment settings.*

The analysis starts by associating all of the variants from the input variant file with genes in the gene track, based on overlap with the gene annotations. A variant track can be created with the *CLC Genomics Workbench* variant callers (section 31.1, 27.10 and 31.2).

Next, the Workbench tries to match gene names from the gene (annotation) track with the gene names in the GO association file. A gene (annotation) track can be created (see section 24.4.1. Please be aware that the same gene name definition should be used in both files.

Based on this, the Workbench finds GO terms that are over-represented in the list. A 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, and a new variant file with annotated GO terms and the corresponding p-value (see figure 27.73). 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.

Rows: 7,037 GO enrichment analysis Filter

| GO term | Description | Occurrences in all g... | Occurrences in sample | P-values |
|---|---|---|---|---|
| 0002755 | MyD88-dependent toll-like receptor signaling pathway | 6 | 4 | 2.38E-6 |
| 0032755 | positive regulation of interleukin-6 production | 22 | 6 | 3.64E-6 |
| 0032757 | positive regulation of interleukin-8 production | 13 | 5 | 3.66E-6 |
| 0034123 | positive regulation of toll-like receptor signaling pathway | 4 | 3 | 3.23E-5 |
| 0042346 | positive regulation of NF-kappaB import into nucleus | 14 | 4 | 1.40E-4 |
| 0007252 | I-kappaB phosphorylation | 6 | 3 | 1.57E-4 |
| 0032722 | positive regulation of chemokine production | 7 | 3 | 2.70E-4 |
| 0050707 | regulation of cytokine secretion | 7 | 3 | 2.70E-4 |
| 0071224 | cellular response to peptidoglycan | 2 | 2 | 4.09E-4 |
| 0050729 | positive regulation of inflammatory response | 19 | 4 | 4.99E-4 |
| 0044130 | negative regulation of growth of symbiont in host | 11 | 3 | 1.20E-3 |
| 0002830 | positive regulation of type 2 immune response | 3 | 2 | 1.21E-3 |
| 0045356 | positive regulation of interferon-alpha biosynthetic process | 3 | 2 | 1.21E-3 |
| 0042773 | ATP synthesis coupled electron transport | 4 | 2 | 2.39E-3 |
| 0048016 | inositol phosphate-mediated signaling | 4 | 2 | 2.39E-3 |
| 0050766 | positive regulation of phagocytosis | 14 | 3 | 2.53E-3 |
| 0071260 | cellular response to mechanical stimulus | 30 | 4 | 2.97E-3 |
| 0001867 | complement activation, lectin pathway | 5 | 2 | 3.93E-3 |
| 0002024 | diet induced thermogenesis | 5 | 2 | 3.93E-3 |
| 0042116 | macrophage activation | 5 | 2 | 3.93E-3 |
| 0045080 | positive regulation of chemokine biosynthetic process | 5 | 2 | 3.93E-3 |
| 0045359 | positive regulation of interferon-beta biosynthetic process | 5 | 2 | 3.93E-3 |
| 0006955 | immune response | 101 | 7 | 4.35E-3 |
| 0051092 | positive regulation of NF-kappaB transcription factor activity | 55 | 5 | 4.97E-3 |
| 0002925 | positive regulation of humoral immune response mediated by circulating immunog... | 6 | 2 | 5.82E-3 |
| 0006120 | mitochondrial electron transport, NADH to ubiquinone | 6 | 2 | 5.82E-3 |
| 0051607 | defense response to virus | 58 | 5 | 6.24E-3 |
| 0019221 | cytokine-mediated signaling pathway | 19 | 3 | 6.26E-3 |
| 0032760 | positive regulation of tumor necrosis factor production | 19 | 3 | 6.26E-3 |
| 0050830 | defense response to Gram-positive bacterium | 19 | 3 | 6.26E-3 |
| 0006954 | inflammatory response | 86 | 6 | 7.88E-3 |
| 0002504 | antigen processing and presentation of peptide or polysaccharide antigen via MH... | 7 | 2 | 8.04E-3 |
| 0045078 | positive regulation of interferon-gamma biosynthetic process | 7 | 2 | 8.04E-3 |
| 0048246 | macrophage chemotaxis | 7 | 2 | 8.04E-3 |
| 0050718 | positive regulation of interleukin-1 beta secretion | 7 | 2 | 8.04E-3 |
| 0050730 | regulation of peptidyl-tyrosine phosphorylation | 7 | 2 | 8.04E-3 |
| 0006465 | signal peptide processing | 8 | 2 | 0.01 |
| 0008306 | associative learning | 8 | 2 | 0.01 |
| 0003214 | cardiac left ventricle morphogenesis | 9 | 2 | 0.01 |
| 0007338 | single fertilization | 9 | 2 | 0.01 |
| 0018149 | peptide cross-linking | 9 | 2 | 0.01 |
| 0051894 | positive regulation of focal adhesion assembly | 9 | 2 | 0.01 |
| 0008203 | cholesterol metabolic process | 25 | 3 | 0.01 |
| 0021987 | cerebral cortex development | 25 | 3 | 0.01 |
| 0032695 | negative regulation of interleukin-12 production | 10 | 2 | 0.02 |
| 0033198 | response to ATP | 10 | 2 | 0.02 |
| 0030198 | extracellular matrix organization | 51 | 4 | 0.02 |
| 0042523 | positive regulation of tyrosine phosphorylation of Stat5 protein | 11 | 2 | 0.02 |
| 0048013 | ephrin receptor signaling pathway | 11 | 2 | 0.02 |
| 0002023 | reduction of food intake in response to dietary excess | 1 | 1 | 0.02 |
| 0002238 | response to molecule of fungal origin | 1 | 1 | 0.02 |
| 0002369 | T cell cytokine production | 1 | 1 | 0.02 |

Figure 27.73: *The results of the analysis.*

### 27.15.4 Conservation score annotation

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** ( ) | **Functional Consequences** | **Annotate with Conservation Score**

Select the variant track as input and when you click **Next** you will need to provide the track with conservation scores (see figure 27.74).

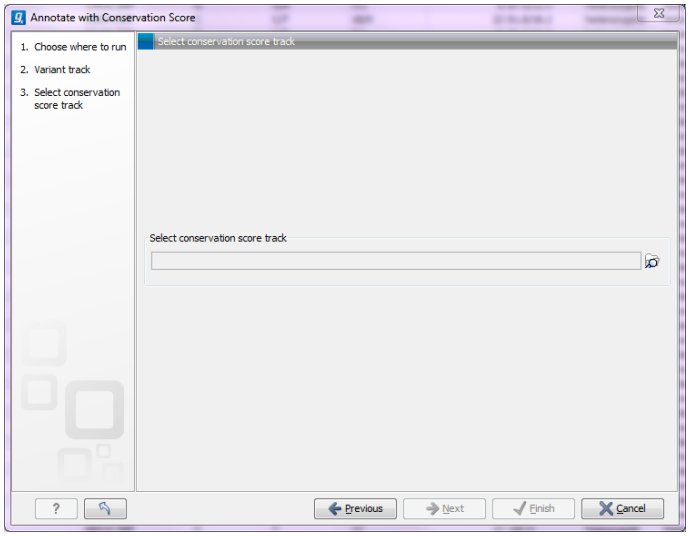



Figure 27.74: *The conservation score track.*

In the resulting track, all the variants will have quality scores annotated, and this can be used for sorting and filtering the track (see section 24.1.3).

### 27.15.5 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 an additional column ('Link to 3D Protein Structure') in the table view. The added column contains links that will invoke a menu with a one-click option to create a structure model of the affected protein, and open a visualization of the structural consequences of the variant in the workbench 3D viewer.

In section 27.15.2 it is described how the tool finds appropriate protein structures to use for the visualization, and in section 27.15.2 and onwards it is described how the visualization is 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 27.15.6).

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 27.75).

Click on the button labeled **Next**. In the next wizard step, you must provide a CDS track and the

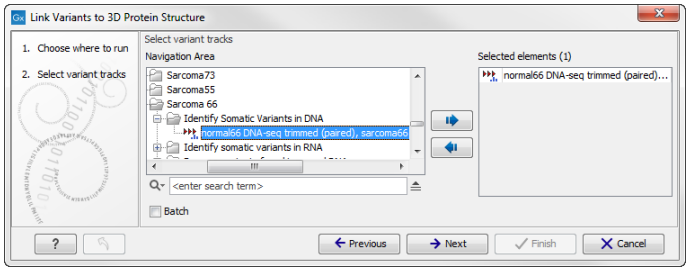



Figure 27.75: *Select the variant track holding the variants that you would like to visualize on 3D protein structures.*

reference sequence track (figure 27.76).

If you have not already downloaded a CDS and a reference sequence track, this can be done in the upper right corner of the Workbench by clicking on the button labeled **Download** and choosing **Download Reference Genome Data ( )** (see section 11.4).

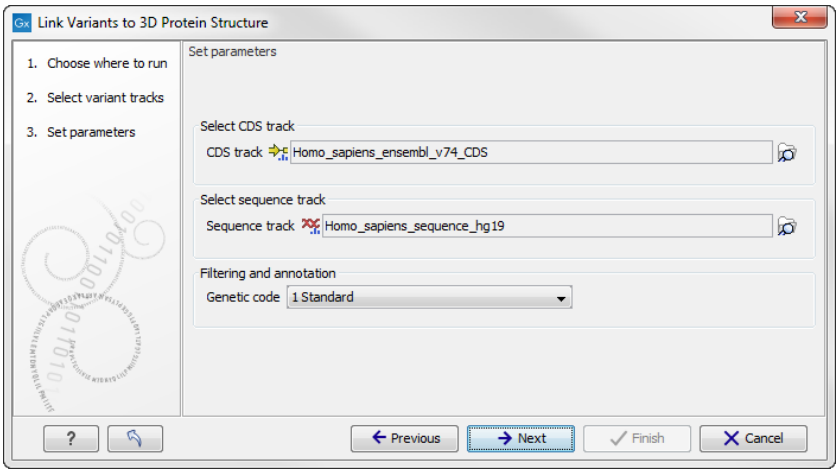



Figure 27.76: *Select CDS and reference sequence.*

Click on the button labeled **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 an additional column in the table view ('Link to 3D Protein Structure') containing links to the modeled structures (figure 27.77). 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.

### How the protein structures are found

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" column 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

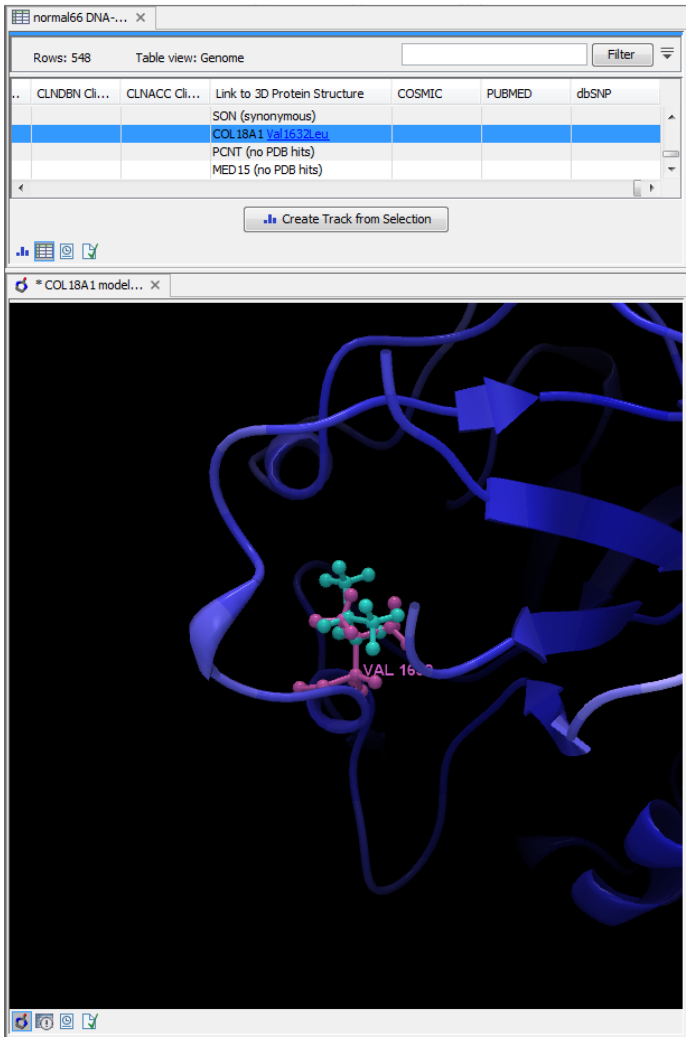



Figure 27.77: *The output shown in table format. The lower part of the figure shows a 3D view of one of the variants that have been opened using the link provided in the 'Link to 3D Protein Structure' column.*

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 27.15.6) 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 27.15.2).

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 (see section 27.15.2). Note that the amino acid numbering is based on the longest CDS annotation found.

#### 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 27.77).

- '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 27.78).

  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 27.15.2).

2. **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).

3. **Create a model structure for the reference** by mapping it onto the template structure based on the sequence alignment (see section 27.15.2).

4. **Create a model structure with variant(s)** by mapping the protein sequence with the variant consequences onto the reference structure model (see section 27.15.2).

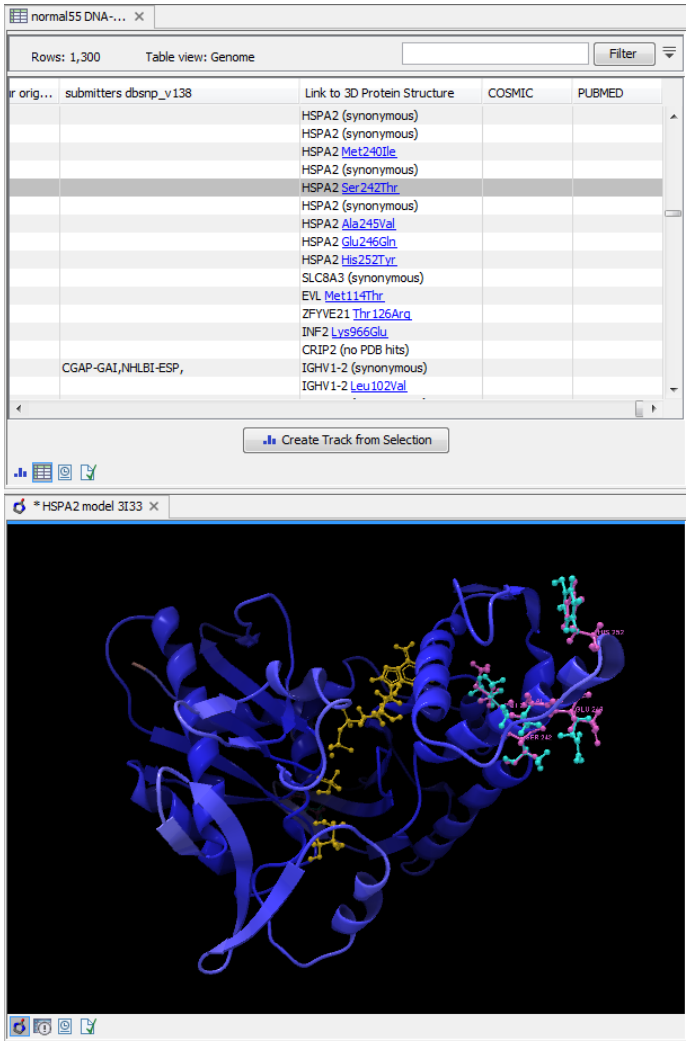



Figure 27.78: *Visualization of multiple variations on a protein structure.*

5. **Open a 3D view** (a Molecule Project) with the variant structure model shown in backbone representation. The model is colored by temperature (see figure 27.78), to indicate local model uncertainty (see section 27.15.2). 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.

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. 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 13.3.

#### 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 13.3.1).

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 27.79 and 27.80.

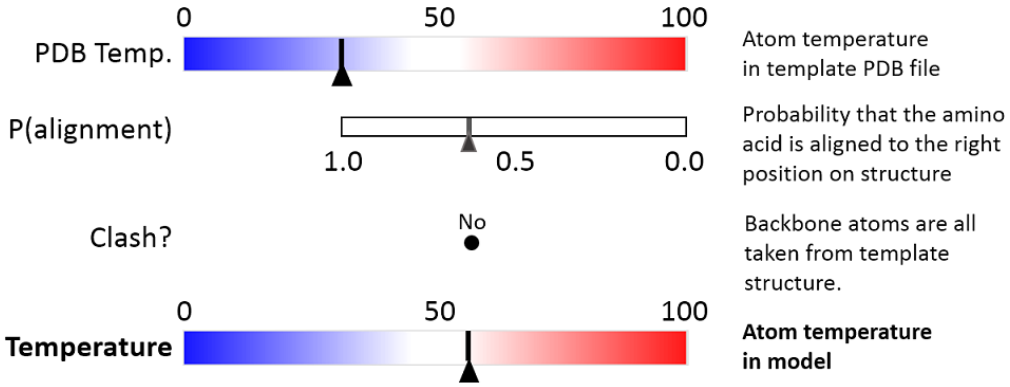



Figure 27.79: *Evaluation of temperature color for backbone atoms in structure models.*

When holding the mouse over an atom, the Property Viewer in the Side Panel will show various information about the atom. For atoms in structure models, the contributions to the assigned temperature are listed as seen in figure 27.81.

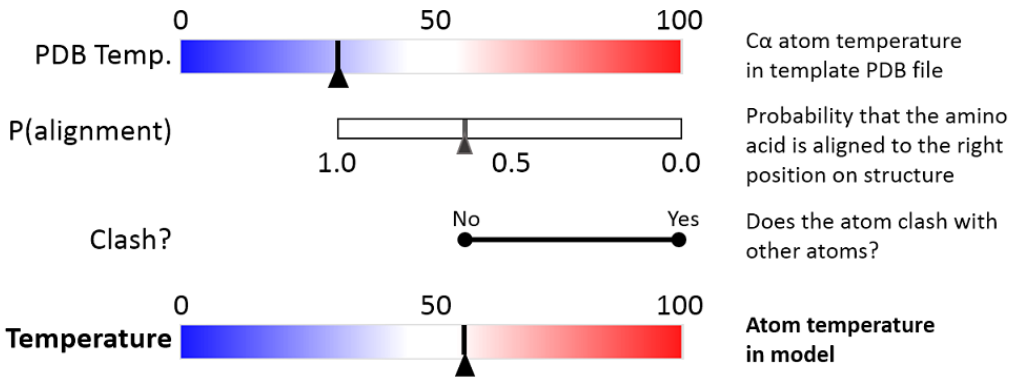



Figure 27.80: *Evaluation of temperature color for side chain atoms in structure models.*

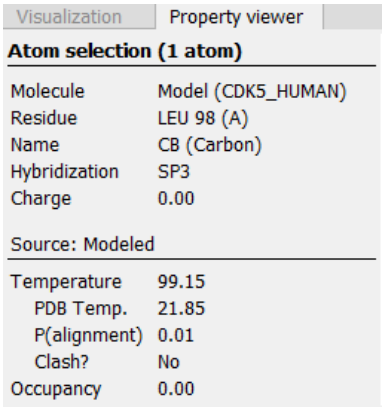



Figure 27.81: *Information displayed in the Side Panel Property viewer for a modeled atom.*

**Note:** For NMR structures, the temperature factor is set to zero in the PDB file, and the "Color by Temperature" will therefore suggest that the structure is more well determined than is actually the case.

**P(alignment)**

Alignment error is one of the largest causes of model inaccuracy, particularly when the model is built from a template sharing low sequence identity (e.g. lower than 60%). Misaligning a single amino acid by one position will cause a ca. 3.5 Å shift of its atoms from their true positions.

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. The scoring function used to evaluate protein-ligand interactions for docking and ligand optimization in *CLC Drug Discovery Workbench* is also used to evaluate the interaction between a given side chain atom and its surroundings.

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.

#### Method details

#### Ranking structures

The protein sequence of the gene affected by the variant (the query sequence) is BLASTed against the protein structure sequence database (section 27.15.6).

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 27.83. 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 12.5.

**% Match identity** is the identity between the query sequence and the BLAST hit in the matched region. It is evaluated as

$$\text{\% Match identity} = 100\% \cdot (\text{Identity in BLAST alignment})/L_{\text{B}}$$

where $L_{\text{B}}$ is the length of the BLAST alignment of the matched region, as indicated in figure 27.82, 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 27.82). 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.

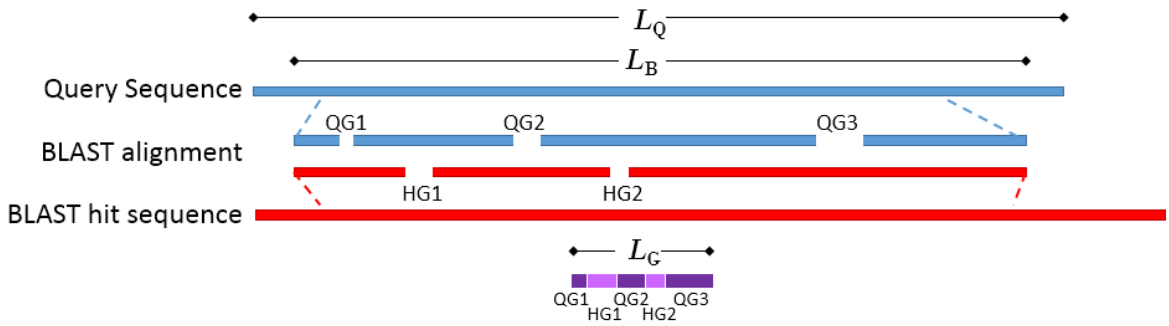



Figure 27.82: *Schematic of a query sequence matched to a BLAST hit. $L_Q$ is the length of the query sequence, $L_B$ is the length of the BLAST alignment of the matched region, QG1-3 are gaps in the matched region of the query sequence, HG1-2 are gaps in the matched region of the BLAST hit sequence, $L_G$ 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 27.84):

- For identical amino acids (example 1 in figure 27.84) => Copy atom positions from the PDB file. If the side chain is missing atoms in the PDB file, the side chain is rebuilt (section 27.15.2).
- For amino acid changes (example 2 in figure 27.84) => Copy backbone atom positions from the PDB file. Model side chain atom positions to match the query sequence (section 27.15.2).
- For amino acids in the query sequence not aligned to a position on the template structure (example 3 in figure 27.84) => No atoms are modeled. The model backbone will have a gap at this position and a "Structure modeling" issue is raised (see section 13.1.4).
- For amino acids on the template structure, not aligned to the query sequence (example 4 in figure 27.84) => The residues are deleted from the structure and a "Structure modeling" issue is raised (see section 13.1.4).

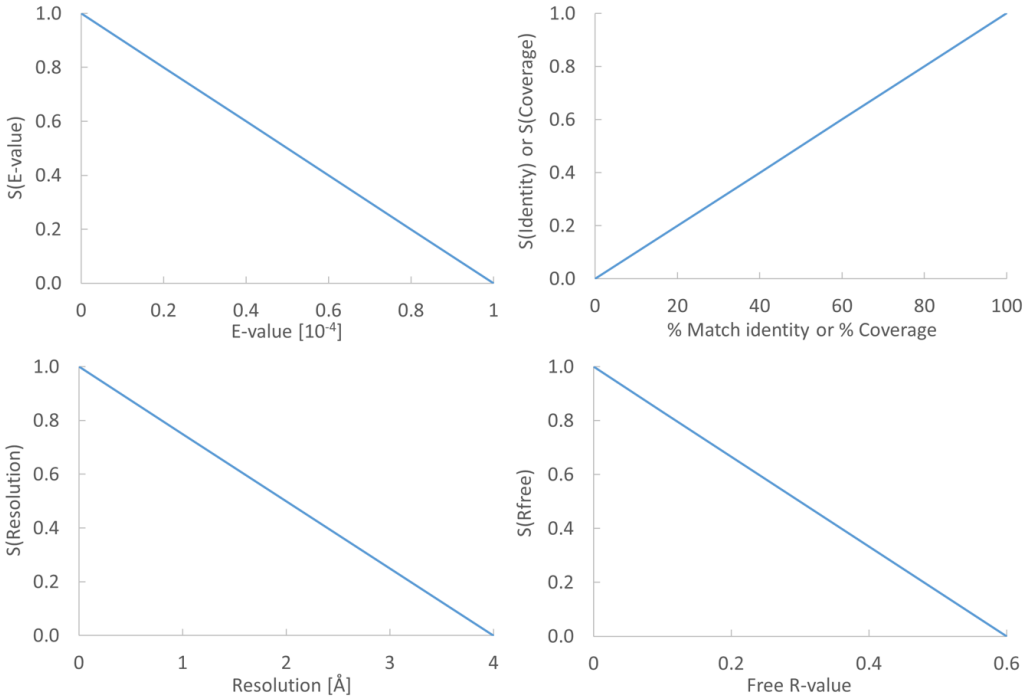



Figure 27.83: *From the E-value, % Match identity, % Coverage, Resolution, and Free R-value, the contributions to the "Template quality score" are determined from the linear functions shown in the graphs.*

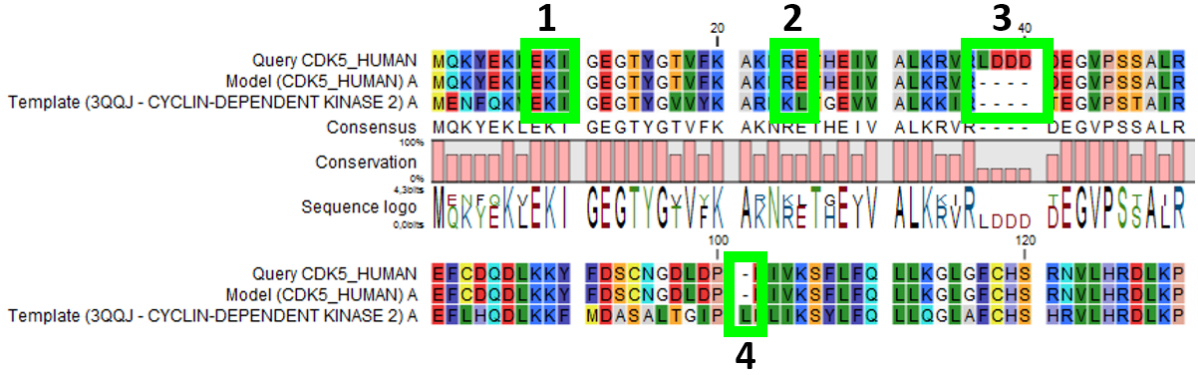



Figure 27.84: *Sequence alignment mapping query sequence (Query CDK5_HUMAN) to the structure with sequence "Template(3QQJ - CYCLIN-DEPENDENT KINASE 2)", producing a structure with sequence "Model(CDK5_HUMAN)". Examples are highlighted: 1. Identical amino acids, 2. Amino acid changes, 3. Amino acids in query sequence not aligned to a position on the template structure, and 4. Amino acids on the template structure, not aligned to query sequence.*

### 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 $\phi$ and $\psi$.
- Atom interaction potential: This score is used to evaluate the interaction between a given side chain atom and its surroundings. It is the same score that is used to evaluate protein-ligand interactions for docking and ligand optimization in *CLC Drug Discovery Workbench*.
- 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.

### 27.15.6 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, select:

**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 27.85).

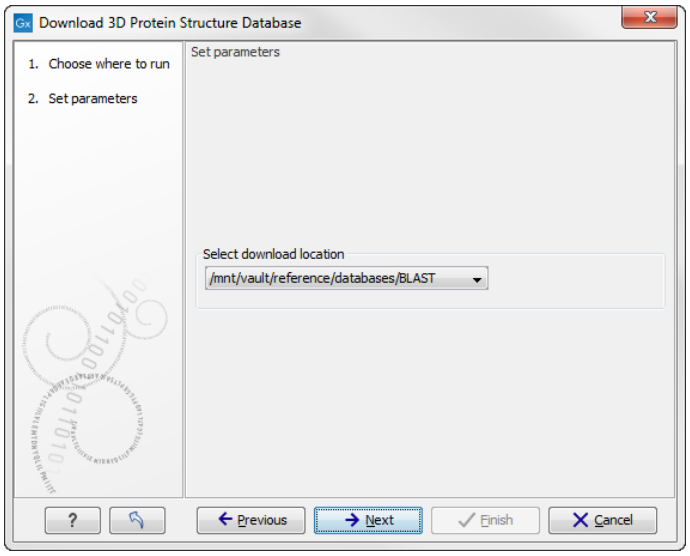



Figure 27.85: *Select the download location.*

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** ( )

# Chapter 28

# Transcriptomics

## Contents


**28.1 RNA-Seq analysis . . . 713**
28.1.1 Specifying reads and reference . . . 714
28.1.2 Tightly packed genes and genes in operons . . . 716
28.1.3 Defining mapping options for RNA-Seq . . . 716
28.1.4 Calculating expression values from RNA-Seq . . . 719
28.1.5 Specifying RNA-Seq outputs . . . 721
28.1.6 Interpreting the RNA-Seq analysis result . . . 722
**28.2 Small RNA analysis . . . 730**
28.2.1 Extract and count . . . 731
28.2.2 Downloading miRBase . . . 735
28.2.3 Annotating and merging small RNA samples . . . 736
28.2.4 Working with the small RNA sample . . . 746
28.2.5 Exploring novel miRNAs . . . 748
**28.3 Expression profiling by tags . . . 749**
28.3.1 Extract and count tags . . . 749
28.3.2 Create virtual tag list . . . 753
28.3.3 Annotate tag experiment . . . 757
**28.4 Experimental design . . . 760**
28.4.1 Setting up an experiment . . . 760
28.4.2 Organization of the experiment table . . . 763
28.4.3 Visualizing RNA-Seq read tracks for the experiment . . . 768
28.4.4 Adding annotations to an experiment . . . 769
28.4.5 Scatter plot view of an experiment . . . 770
28.4.6 Cross-view selections . . . 772
**28.5 Working with tracks and experiments . . . 774**
28.5.1 Data structures for transcriptomics . . . 774
28.5.2 From Experiments to Tracks . . . 775
28.5.3 Running the Create Track from Experiment tool . . . 775
28.5.4 Interpreting the results of the Create Track from Experiment tool . . . 779
**28.6 Transformation and normalization . . . 780**

28.6.1 Selecting transformed and normalized values for analysis . . . . . 781
28.6.2 Transformation . . . . . 781
28.6.3 Normalization . . . . . 782
**28.7 Quality control . . . . . 784**
28.7.1 Creating box plots - analyzing distributions . . . . . 784
28.7.2 Hierarchical clustering of samples . . . . . 788
28.7.3 Principal component analysis . . . . . 793
**28.8 Statistical analysis - identifying differential expression . . . . . 798**
28.8.1 Empirical analysis of DGE . . . . . 798
28.8.2 Tests on proportions . . . . . 802
28.8.3 Gaussian-based tests . . . . . 803
28.8.4 Corrected p-values . . . . . 806
28.8.5 Volcano plots - inspecting the result of the statistical analysis . . . . . 807
**28.9 Feature clustering . . . . . 809**
28.9.1 Hierarchical clustering of features . . . . . 809
28.9.2 K-means/medoids clustering . . . . . 813
**28.10 Annotation tests . . . . . 817**
28.10.1 Hypergeometric tests on annotations . . . . . 817
28.10.2 Gene set enrichment analysis . . . . . 819
**28.11 General plots . . . . . 823**
28.11.1 Histogram . . . . . 823
28.11.2 MA plot . . . . . 825
28.11.3 Scatter plot . . . . . 828

## 28.1 RNA-Seq analysis

Two tools are available for RNA-seq analysis, the tool **RNA-Seq Analysis** and the tool **Create Fold Change Track** Based on an annotated reference genome, the *CLC Genomics Workbench* supports RNA-Seq analysis by mapping next-generation sequencing reads and counting and distributing the reads across genes and transcripts. Subsequently, the results can be used for expression analysis using the tools in the **Transcriptomics Analysis** toolbox.

The tool for RNA-Seq analysis can be found here:

**Toolbox** | **Transcriptomics Analysis** ( ) | **RNA-Seq Analysis ( )**

The approach taken by the *CLC Genomics Workbench* is based on [Mortazavi et al., 2008].

The following describes the overall process of the RNA-Seq analysis when using an annotated eukaryote genome. See section 28.1.1 for more information on other types of reference data.

The RNA-Seq analysis is done in several steps: First, all genes are extracted from the reference genome (using a *gene* track). Next, 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 28.1.

This is a simple gene with three exons and two splice variants. The transcripts are extracted as shown in figure 28.2.

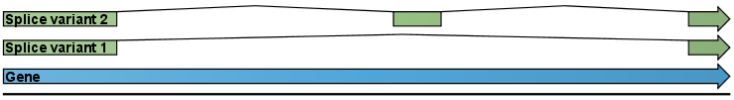



Figure 28.1: *A simple gene with three exons and two splice variants.*

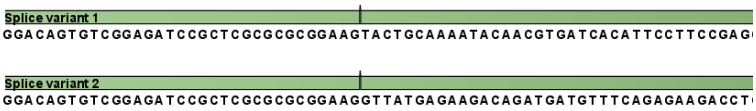



Figure 28.2: *All the exon-exon junctions are joined in the extracted transcript.*

Next, the reads are mapped against all the transcripts plus the entire gene (see figure 28.3) and optionally to the whole genome.

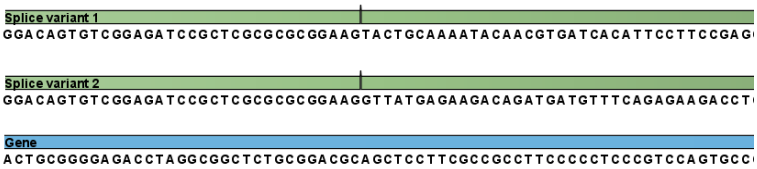



Figure 28.3: *The reference for mapping: all the exon-exon junctions and the gene.*

From this mapping, the reads are categorized and assigned to the genes (elaborated later in this section), and expression values for each gene and each transcript are calculated.

Details on the process are elaborated in the following sections, which describe how to run RNA-seq analyses.

### 28.1.1 Specifying reads and reference

To start the RNA-Seq analysis, go to:

**Toolbox** | **Transcriptomics 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 8.1).

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 28.4.

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. The mRNA annotations are used to define how the transcripts are spliced (as shown in figure 28.1). This option should be used for **Eukarotes** 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:

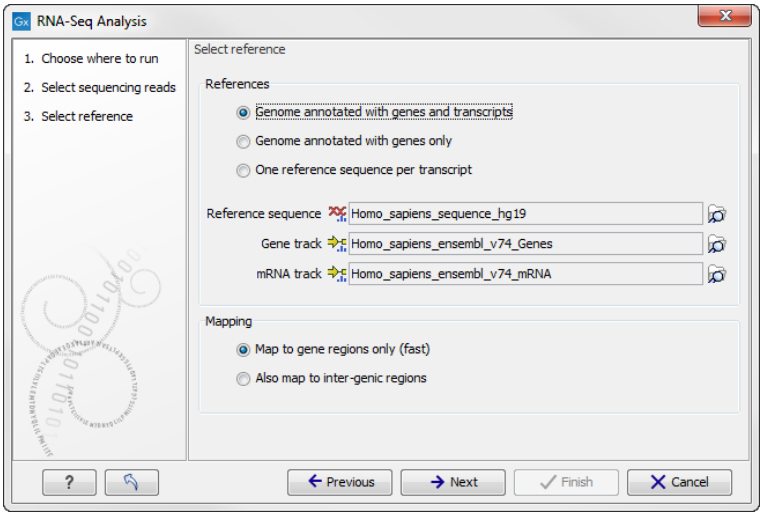



Figure 28.4: *Defining a reference genome for RNA-Seq.*

  - Directly downloaded as tracks using the **Download Reference Genome Data** tool (see section 11.4).
  - 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 24.4).
  - Downloaded from Genbank (see section 11.1) and converted to tracks (see section 24.4).

- **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.

- **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. When this option is selected, only the reference sequence should be provided, either as a sequence track or a sequence list.

At the bottom of the dialog you can choose the reference content to map to. Note that this is only relevant when using an annotated reference:

- **Map to gene regions only (fast)**. This option will ignore all inter-genic regions in the reference. Since only genes are considered, this options is also significantly faster than the alternative option. The effect of restricting the mapping to genes only is that any reads coming from genes or transcripts that are not part of the annotations will either be unmapped or map to another transcript with a similar sequence (e.g. a pseudo-gene). For poorly annotated references, it is possible to improve the annotations using the **Transcript Discovery** plugin which is freely available for download in the **Plugin Manager** (see section 1.7.1).

- **Also map to inter-genic regions**. This option will include the inter-genic regions as well. Please note that reads that map outside genes are counted as intergenic hits only and thus

do not contribute to the expression values[1]. If a read maps equally well to a gene and to an inter-genic region, the read will be placed in the gene.

### 28.1.2 Tightly packed genes and genes in operons

For annotated references containing genes located very close to each other, as commonly found in bacteria, including operon structures, we recommend that the option "Map genes to gene regions only (fast)" is chosen. In this case, reads that map over a gene boundary will be counted towards the expression value for the gene they best map to.

With the mapping option "Also map to inter-genic regions" selected, only reads mapping completely within a genes 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.

In both cases, reads are only mapped to, and thus counted towards the expression value of, one gene.

If differential expression is the main interest, then whether the intergenic mapping option is chosen or not may make little difference to the final results. However, 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 when the "Also map to inter-genic regions" is chosen.

Please also refer to the section entitled "Genes in Operons", which is found in section 28.1.4.

### 28.1.3 Defining mapping options for RNA-Seq

When the reference has been defined, click **Next** and you are presented with the dialog shown in figure 28.5.

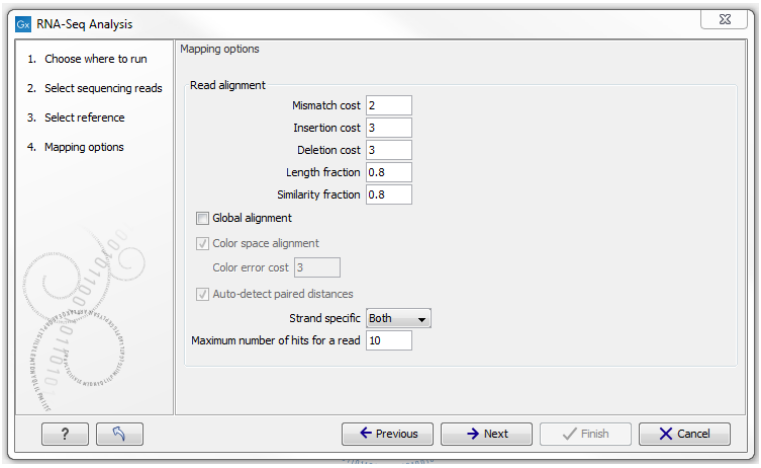



Figure 28.5: *Defining mapping parameters for RNA-Seq.*

The mapping parameters are identical to those applying to **Map Reads to Reference**, as the

[1]The reads will indirectly impact the RPKM expression values as they will be counted in the total number of mapped reads which is used to calculate RPKM (section 28.1.6)

underlying mapping is performed in the same way. For a description of the parameters, please see section 25.1.3.

For the estimation of paired reads distances, RNA-Seq uses the transcript level reference sequence information. This means that intronic regions are not included during this step - reflecting the true nature of the transcript on based 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 references than the 'Maximum number of hits for a read' specified will not be mapped (the notion of *distinct* places is elaborated below). If a read matches to multiple distinct places, but less than the specified maximum number, it will be randomly assigned to one of these places. The random distribution is done proportionally to the number of unique matches that the genes to which it matches have, normalized by the exon length (to ensure that genes with no unique matches have a chance of having multi-matches assigned to them, 1 will be used instead of 0, for their count of unique matches). This means that if there are 10 reads that match two different genes with equal exon length, the 10 reads will be distributed according to the number of unique matches for these two genes. The gene that has the highest number of unique matches will thus get a greater proportion of the 10 reads.

  The definition of a *distinct* place in the references is complicated because each annotated transcript is extracted and used as reference for the read mapping (if the "Genome annotated with genes and transcripts" is selected in figure 28.4). To exemplify, consider a gene with 10 transcripts and 11 exons, where all transcripts have exon 1, and each of the 10 transcripts have only one of the exons 2 to 11. Exon 1 will be 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 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 distinct placement of the read, and it will *not* be discarded for exceeding the maximum number of hits limit. Similarly, when a multi-match read is randomly assigned to one of it's match places, each distinct place is considered only once.

  The limit for how many non-specific matches a read is allowed to have, is applied first to the set of gene matches (if any), and then second to the intergenic matches. As an example using the default value of 10, if a read matches equally well 8 places within genes and 50 places in intergenic regions, it is still considered a valid match. It will only be discarded if the number of matches within genes is above the limit, or if there are no gene matches at all and the number of intergenic matches exceeds the limit.

  Note that, although a read is mapped *distinctly* at the gene level, it does not necessarily map *uniquely* to a particular transcript of the gene. The above example with a gene with 10 transcripts and 11 exons, where all transcripts have exon 1, and each of the 10 transcripts have only one of the exons 2 to 11, is a good an easy to understand example of this: all reads that are mapped to exon 1 are *uniquely* mapped at the gene level but are *non-specific matches* at the transcript level. A more complicated example is that you may have a gene with transcript annotations where one transcript has a longer version of an exon than the other. 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. Such an example is provided by the gene 'Ftl1' in the RNA-seq

Mouse Chromosome 7 Tutorial data. The gene and mRNA annotations for that gene are shown in Figure 28.6, along with the reads mapping to the gene.

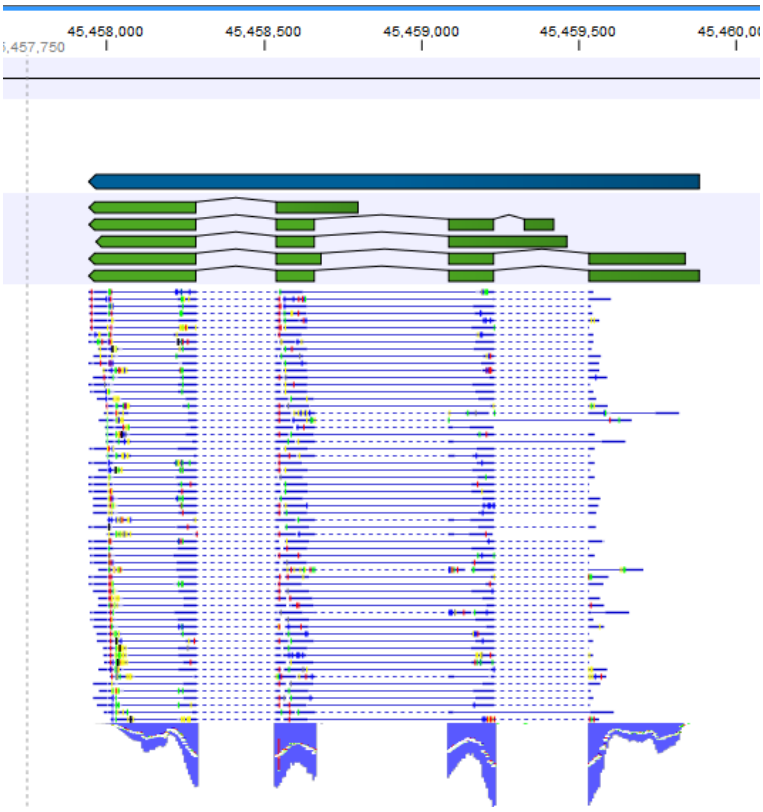



Figure 28.6: *The gene 'Ftl1' from the mouse chromosome 7 tutorial data.*

When you zoom in on the regions at the end of the second exons and the beginning of the third exons (Figure 28.7) you see that the reference sequence is identical in the start of the part of the second exons that is only present in the long version, and in the start of the third exons (they share the sequence 'CTGCACA'). So a read that is e.g. '...TCATCTTGAGATGGCTTCTGCACA' may be either mapped entirely within the long version of the second exons, or across the exon-exon boundary of the short version of the second exon and the third exon. When it comes to reporting expression levels at the *transcript* level, reads are randomly assigned among the transcripts to which they map. Note that this introduces some randomness in the numbers of total exon reads for the *transcripts* (but not for the *gene*), even when you require that only specific matches are used. Also, as there is the chance that a read may sometimes be assigned to a transcript for which it is an exon-exon read, and sometimes to a transcript for which it is mapped entirely within an exon, *even* for a run with the **maximum number of hits for a read** parameter set to 1, there is a random component to the number of exon-exon reads reported, but not to the total number of exon reads.

- **Strand-specific alignment**. When this option is checked, the user can specify whether the reads should be mapped only in their forward (or reverse) orientation. 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]). Also, applying the 'strand specific' 'reverse' option in an RNA-seq run could allow the user to assess the degree of antisense transcription.

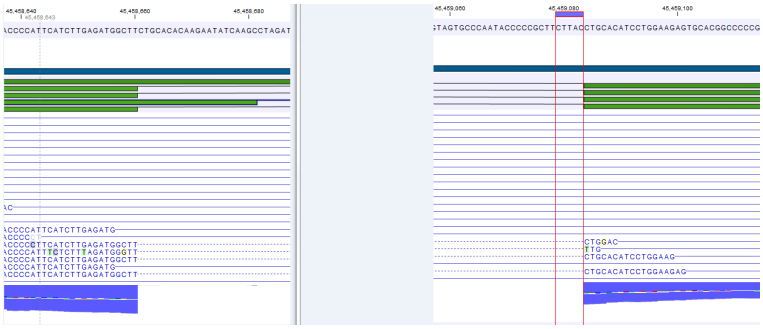


Figure 28.7: *The regions at the end of the second exons and the beginning of the third exons of the mRNA transcripts for the gene 'Ftl1'.*

### 28.1.4 Calculating expression values from RNA-Seq

When the reference has been defined, click **Next** and you are presented with the dialog shown in figure 28.8.

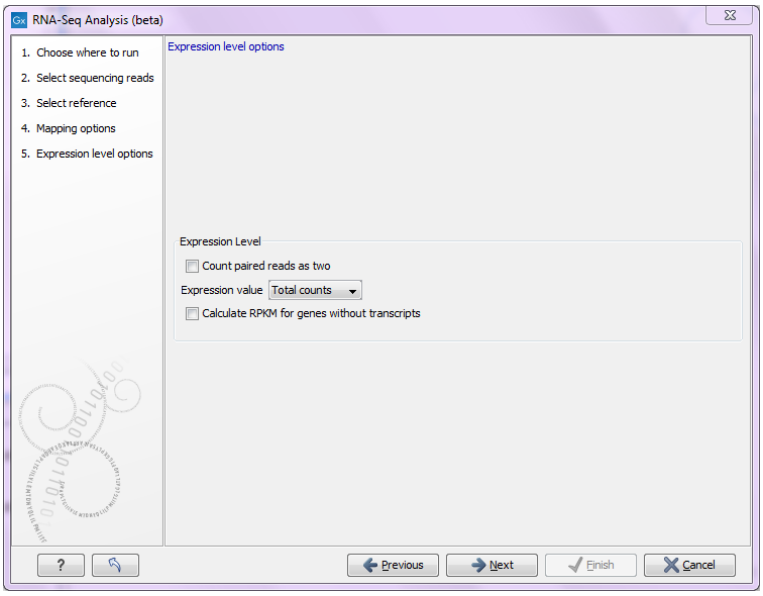



Figure 28.8: *Defining how expression values should be calculated.*

These parameters determine the way expression values are counted. Some background information on how paired reads are handled is useful before describing the parameters.

**Paired reads in RNA-Seq**

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.[2]

- 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 is supporting the presence of a fusion gene.

You can read more about how paired data are imported and handled in section 6.3.8.

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 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, either map in wrong orientation or only one of the reads of the pair map. Neither member of a broken pair are counted when the default counting scheme is used. The reasoning is that when reads map as a broken pair, it is in indication that something is not right. For example, perhaps the transcripts are not represented correctly on the reference or there are errors in the data. In general, more confidence can be placed on an intact pair representing transcription within the sample. If a combination of paired and single reads are input into the analysis, then single reads that map are given a count of one. This is different from reads input into the analysis as part of a pair, but where their partner did not map.

In some situations it may be too strict to disregard broken pairs 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' rather than mapped 'fragments. 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.

When looking at the mappings, reads from broken pairs have a darker color than reads that are intact pairs or originally single reads.

**Expression value**

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 28.4). The content of the result files is described in section 28.1.5.

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

[2] Note that the *CLC Genomics Workbench* only calculates the expression of the transcripts already annotated on the reference.

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 28.1.3).
- **RPKM**. This is a normalized form of the "Total counts" option (see more in section 28.1.6).

Please note that all values are present in the output. The **Expression value** in this dialog is solely used to inform the Workbench about which expression value should be applied when using the result in downstream analysis.

For genes without annotated transcripts, the RPKM cannot be calculated since the total length of all exons is needed. By checking the **Calculate RPKM 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.

**Genes in Operons**
In the case of operons, where several genes are transcribed in the same mRNA transcript and are therefore located directly alongside each other, it is likely that some RNA-seq reads will map across the boundary of two different genes. If the mapping option "Also map to inter-genic regions" has been turned on, then reads like this will not be counted towards expression of any gene. If the option "Map to gene regions only (fast)" has been chosen, then reads will be counted towards only one gene within the operon: the one it mapped to best, which will usually mean the gene which the longest segment of the read mapped to. This means that the value for the expression of a particular operon in the RNA-seq results will be divided across the component genes of that operon.

### 28.1.5 Specifying RNA-Seq outputs

Clicking **Next** will allow you to specify the output options as shown in figure 28.9.

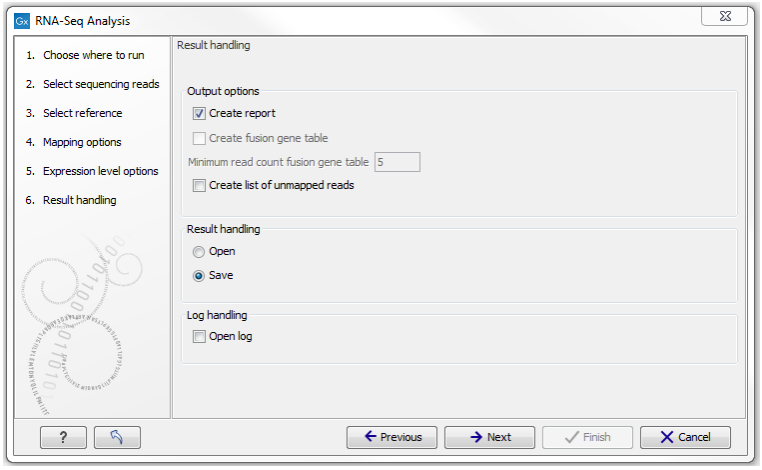



Figure 28.9: *Selecting the output of the RNA-Seq analysis.*

The main results of the RNA-Seq analysis are two expression tracks (one for gene-level and one for transcript-level expression) and a mapping track. In addition, the following optional results can be selected:

- **Create list of unmapped reads**. Creates a list of the reads that either did not map to the reference at all or that were non-specific matches with more placements than specified (see section 28.1.3).
- **Create report**. Creates a report of the results. See **RNA-Seq report** below for a description of the information contained in the report.
- **Create fusion gene table**. An option that is enabled when using paired data. Creates a table that lists potential fusion genes. This, along with the **Minimum read count**, is described further below in section **Gene fusion reporting**.

### 28.1.6 Interpreting the RNA-Seq analysis result

The main results of RNA-Seq analysis are:

- **Expression Tracks** One track summarizing expression at the gene level is produced. The track name ends in **(GE)**. If the "Genome annotated with genes and transcripts" option was selected, as shown in figure 28.4, then a second track summarizing expression at the transcript level is also produced. This track has a name ending with **(TE)**.
- **Reads track** This track contains the mapping of the reads to the references. This track has a name ending with **(Reads)**.

Other outputs, if these have been chosen when setting up the RNA-seq analysis are:

- **RNA-seq report**
- **Fusion gene table**
- **A sequence list of unmapped reads**

The rest of this page describes these output types in more detail.

#### Expression tracks

Both tracks can be shown in a **Table** ( ) and a **Graphical** ( ) 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 28.10).

Reads spanning two exons are shown with a dashed line between each end as shown in figure 28.10, and the thin solid line represents the connection between two reads in a pair.

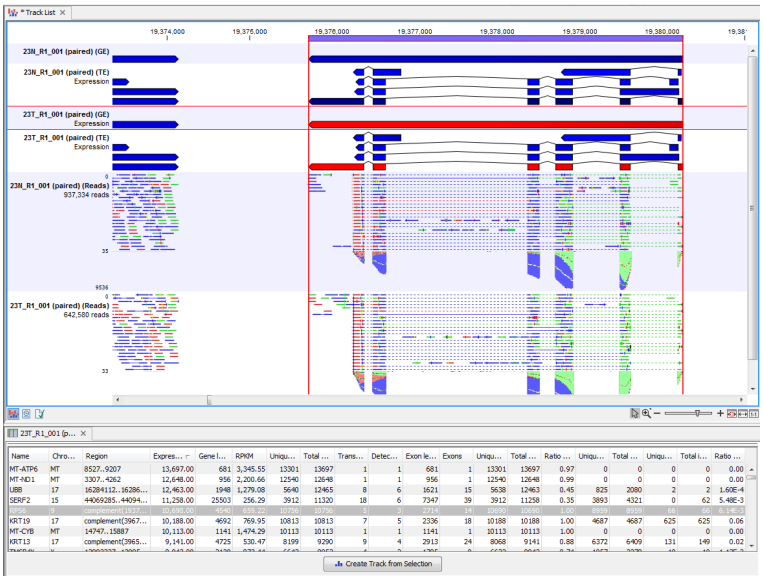


Figure 28.10: *RNA-Seq results shown in a split view with an expression track at the bottom and a track list with read mappings of two samples at the top.*

When doing comparative analysis and opening an experiment (see section 28.4) and a track list, clicking a row in the experiment will cause the track list to jump to the corresponding position, allowing for quick inspection of the reads underlying the counts in the experiment. Please note that at least one of the expression tracks used in the experiment have to be included in the track list in order for the link between the two to work.

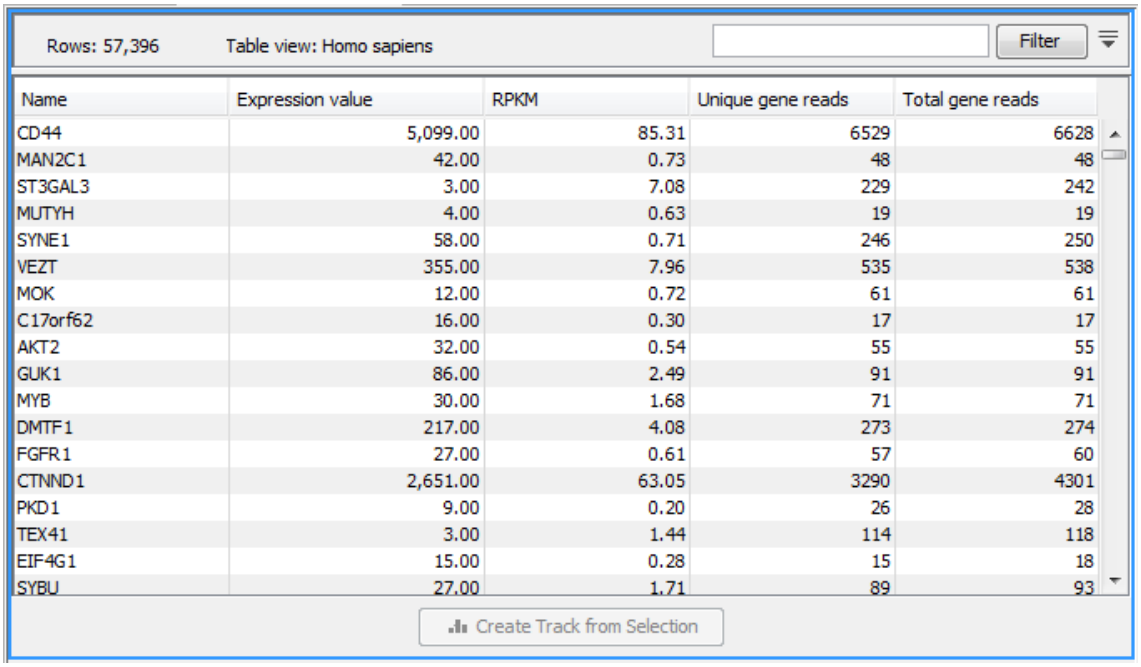
Expression tracks can also be used to annotate variants using the **Annotate with Overlap Information** tool. Select the variant track as input and annotate with the expression track. For variants inside genes or transcripts, information will be added about expression (counts, expression value etc) from the gene or transcript in the expression track. Read more about the annotation tool in section 24.5.1.

**Gene-level expression**

The gene-level expression track 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 28.11 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**. This is the name of the gene or the reference sequence, if the **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 28.1.4.
- **Gene length** The length of the gene as annotated.
- **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 exact definition at end of subsection.



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 28.11: *A subset of a result of an RNA-Seq analysis on the gene level. Not all columns are shown in this figure*

- **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 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).
- **Detected transcripts**. The number of annotated transcripts to which reads have been assigned (see the description of transcript-level expression below).
- **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**. 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.
- **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 28.10). 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 28.10). 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.
- **Unique intron-exon reads**. Reads that uniquely map across an intron-exon boundary of the gene.
- **Total intron-exon reads**. Reads that maps across an intron-exon boundary 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 28.4, a transcript-level expression track 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**. This is the name of the transcript, if the **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 28.1.4.
- **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 at end of subsection.
- **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).

- **Detected transcripts**. The number of annotated transcripts to which reads have been assigned.
- **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.
- **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.
- **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.

#### Reads track

A track containing the mapped reads is generated. Details about viewing and editing of reads-tracks are described in section 24, and the resequencing is detailed in section 27.

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 coloured green, and if it is opposite to the orientation of the gene it is coloured 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 28.12 for an example of forward and reverse reads mapped to a gene on the plus strand.

**Note:** Reads mapping to intergenic regions (if the **Also map to inter-genic regions** box is selected) will not be mapped in a strand specific way.

Although paired reads are coloured 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.

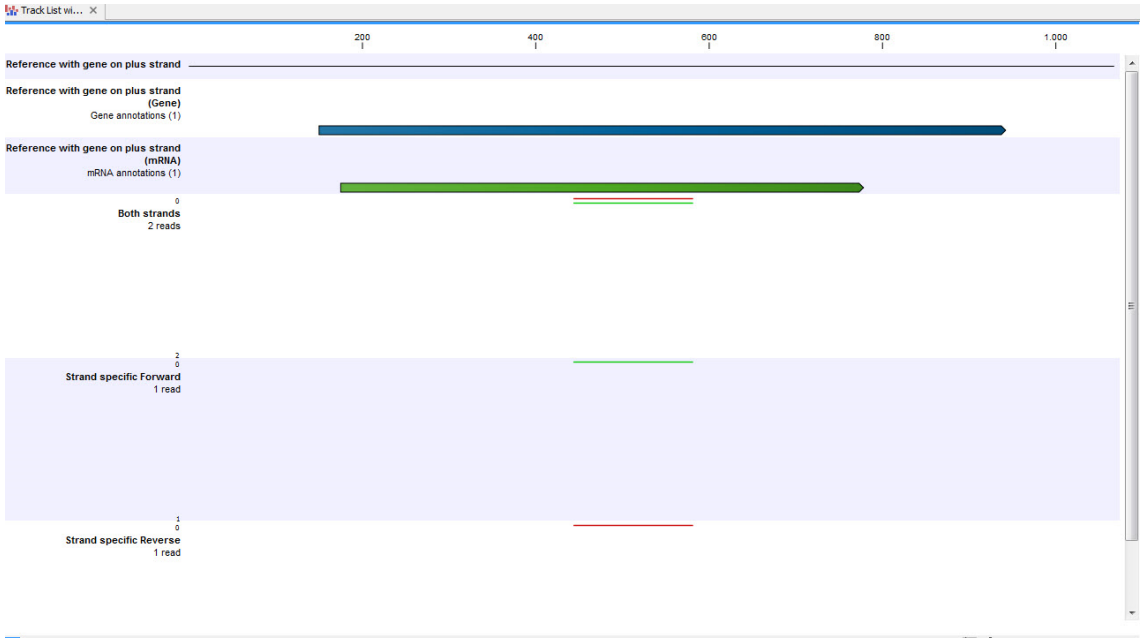



Figure 28.12: *A track list showing a gene and transcript on the plus strand, and various mapping results. The first reads track shows a mapping of two reads (one 'forward' and one 'reverse') using strand specific 'both' option. Both reads map successfully; the forward read coloured green (because it matches the direction of the gene), and the reverse read coloured 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.*

### RNA-Seq report

An example of an RNA-seq report that is created if you choose the **Create report** option is shown in figure 28.13.

The report contains the following information:

- **Sequence reads**. Information about the number of reads.
- **Reference sequences**. Information about the reference sequences used and their lengths.
- **Reference**. Information about the total number of genes and transcripts (for eukaryotes only) found in the reference.
- **Transcripts per gene**. A graph showing the number of transcripts per gene. For eukaryotes, this will be equivalent to the number of mRNA annotations per gene annotation.

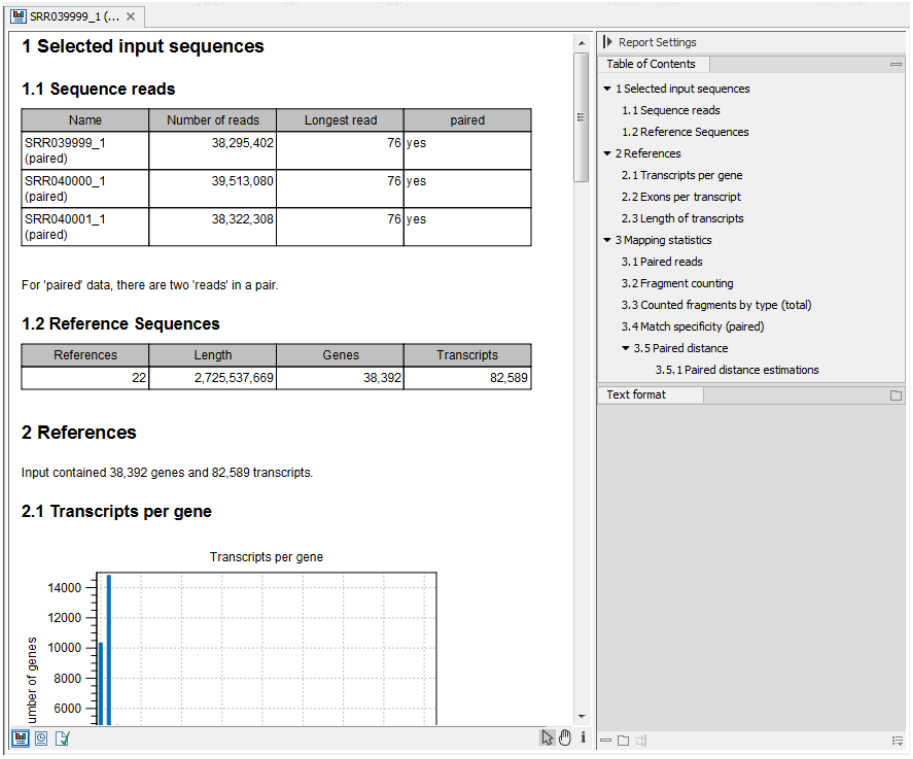



Figure 28.13: *Report of an RNA-Seq run.*

- **Exons per transcript**. A graph showing the number of exons per transcript.
- **Length of transcripts**. A graph showing the distribution of transcript lengths.
- **Mapping statistics**. Shows statistics on:
  - **Paired reads**. (Only included if paired reads are used). Shows the number of reads mapped in pairs, the number of reads in broken pairs and the number of unmapped reads.
  - **Fragment counting**. Lists the total number of fragments used for calculating expression, divided into uniquely and non-specifically mapped reads (see the point below on match specificity for details).
  - **Counted fragments by type**. Divides the fragments that are counted into different types
    * **Exon**. Reads that map completely within an exon
    * **Exon-exon reads**. Reads that map across an exon junction as specified in figure 28.10.
    * **Total exon reads**. Number of reads that fall entirely within an exon or in an exon-exon junction.
    * **Intron**. Reads that fall partly or entirely within an intron.
    * **Total gene reads**. All reads that map to the gene.
    * **Intergenic**. All reads that map partly or entirely between genes (will only be shown if the **Also map to inter-genic regions** option is used).
- **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 28.4. 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.

- **Paired distance**. (Only included if paired reads are used). Shows a graph of the distance between mapped reads in pairs.

Note that the report can be exported in pdf or Excel format.

### 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 28.14.

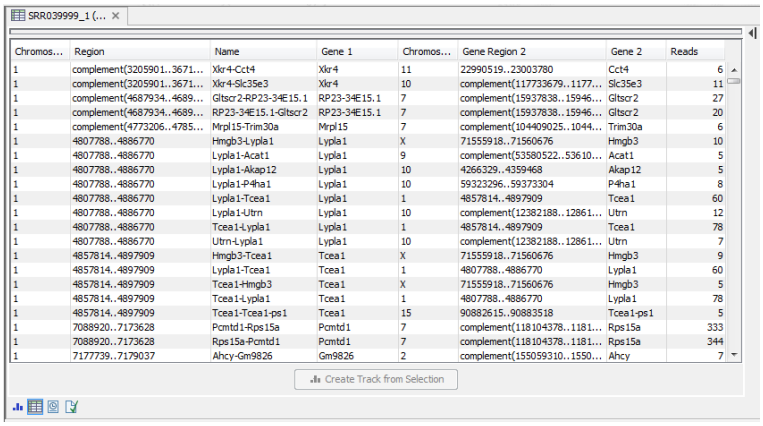



Figure 28.14: *An example of a gene fusion table.*

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 28.9 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 (**  **)**

#### Sequence list of unmapped reads

If you chose the option to **Create a list of unmapped reads**, then a sequence list of these will be produced. If you started with paired reads then more than one list of unmapped reads may be produced: paired reads are put in one list, with a name that ends in (paired), singe reads, including members of broken pairs, are put in a read list with a name than ends in (single).

#### 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 28.1.4. Please note that the option to **Map to gene regions only** will affect the number of mapped reads, since all intergenic reads will not be mapped if this option is selected. This means that comparison of RPKM values between samples should only be carried out if this parameter was set in the same way for all samples.

## 28.2 Small RNA analysis

The small RNA analysis tools in *CLC Genomics Workbench* 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. Illumina, 454 and SOLiD sequencing platforms are supported. For SOLiD, adapter trimming and annotation is done in color space.

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 has the advantage that it 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 (e.g. 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].

In the following, the tools for working with small RNAs are described in detail. Look at the tutorials on `http://www.clcbio.com/tutorials` to see examples of analyzing specific data sets.

### 28.2.1 Extract and count

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):

> **Toolbox** | **Transcriptomics Analysis** ( ) | **Small RNA Analysis** ( ) | **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.

This dialog (see figure 28.15) is where 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.

When you click **Next**, you will be able to specify how the trim should be performed as shown in figure 28.16.

If you have chosen not to trim the reads for adapter sequence, you will see figure 28.17 instead.

The trim options shown in figure 28.16 are the same as described under adapter trim in section 23.1.2. 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**.

Note that all reads will be trimmed for ambiguity symbols such as N before the adapter trim.

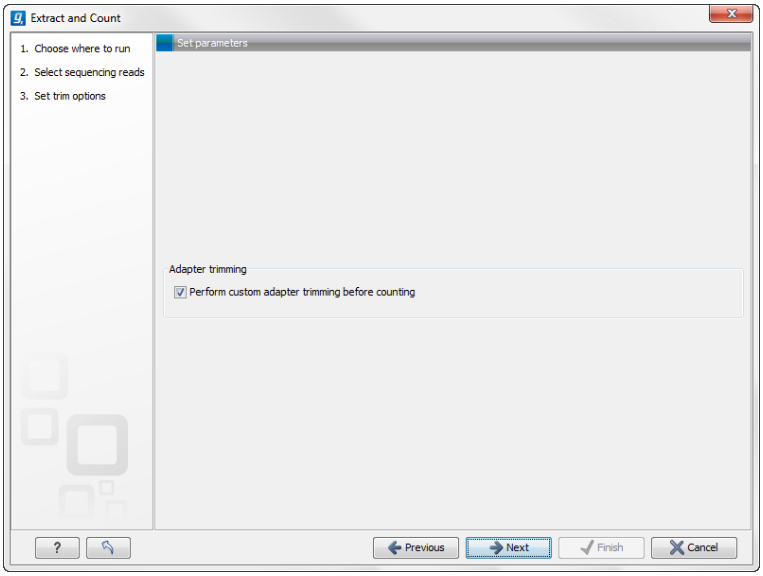



Figure 28.15: *Specifying whether adapter trimming is needed.*

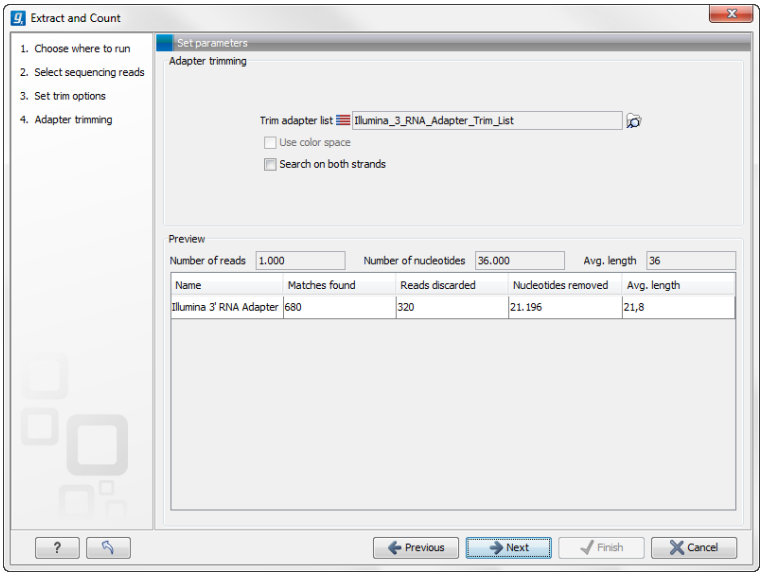



Figure 28.16: *Setting parameters for adapter trim.*

Clicking **Next** allows you to specify additional options regarding trimming and counting as shown in figure 28.17.

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[3]. 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

[3]Note that you can identify variants of the same miRNA when annotating the sample (see below).

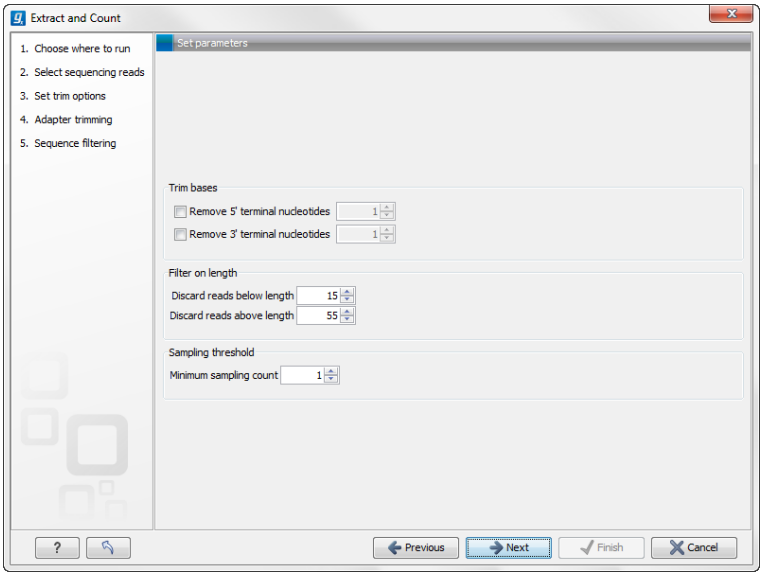



Figure 28.17: *Defining length interval and sampling threshold.*

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 28.18.

The options are:

**Create sample** This is the primary result showing all the tags and respective counts (an example is shown in figure 28.19). Each row represents a tag with the actual sequence as the feature ID and a column with **Length** and **Count**. The actual count is based on 100 % similarity[4]. The sample can be used in further analysis by the tools of the **Transcriptomics Analysis** toolbox 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 28.2.4.

**Create report** This will create a summary report as described below.

[4]Note that you can identify variants of the same miRNA when annotating the sample (see below).

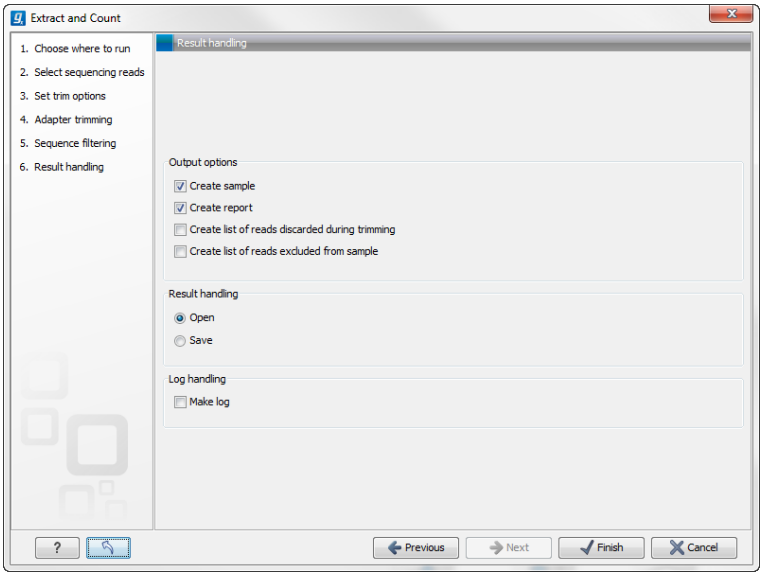



Figure 28.18: *Output options.*

**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.

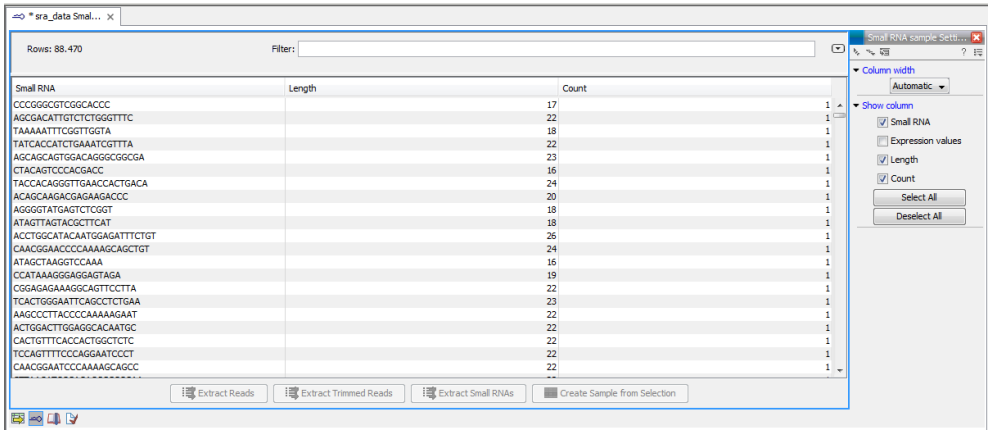



Figure 28.19: *The tags have been extracted and counted.*

The summary report includes the following information (an example is shown in figure 28.20):

**Trim summary** Shows the following information for each input file:

- 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 28.20 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 under adapter trim in section 23.1.2.

**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.

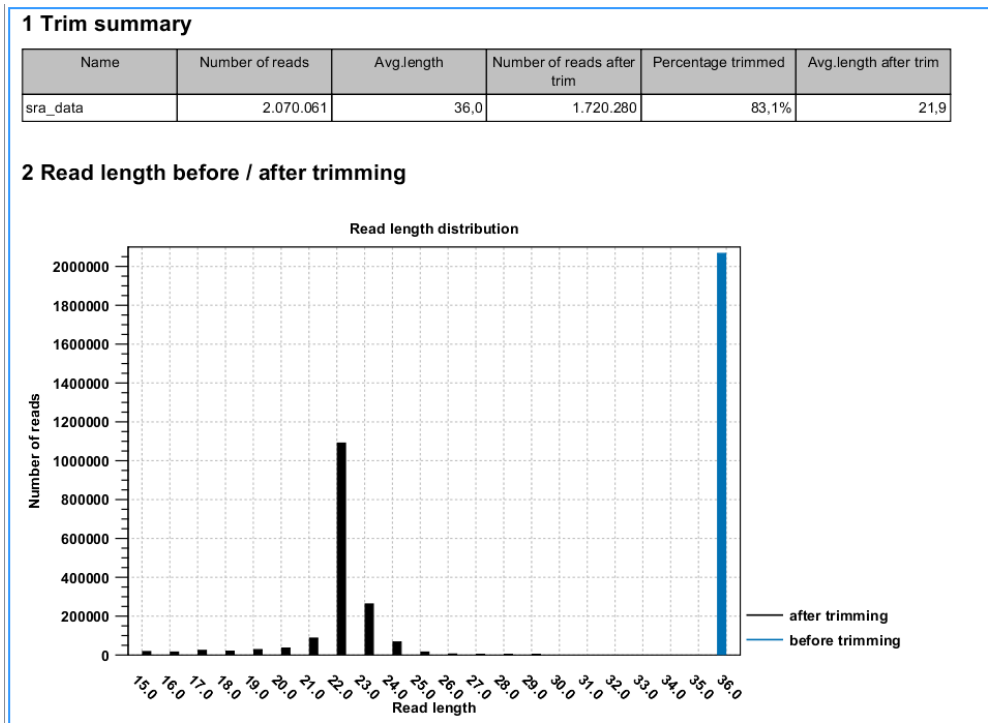



Figure 28.20: *A summary report of the counting.*

### 28.2.2 Downloading miRBase

In order to make use of the additional information about mature regions on the precursor miRNAs in miRBase, you need to use the integrated tool to download miRBase rather than downloading it from `http://www.mirbase.org/`:

**Toolbox** | **Transcriptomics Analysis** ( ) | **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 28.2.3).

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`.

In order 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 import dialog.

#### Creating your own miRBase file

If you wish to construct a file yourself to be used as a miRBase file for annotation, this is also possible if you format the file in the same way as the miRBase data file. In particular, the following needs to be in place:

- The sequences needs "miRNA" annotation on the precursor sequences. In the Workbench, you can add a miRNA annotation by selecting a region and right click to **Add Annotation**. You should have a max 2 miRNA annotations per precursor sequences. 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 form multiple species, the **Latin name** of the sequences should be set. This is used in the annotation dialog (see section 28.2.3) where you can select the species. If the Latin name is not set, the dialog will show "N/A".

Once you have created the file, it has to be imported as described above.

### 28.2.3 Annotating and merging small RNA samples

The small RNA sample produced when counting the tags (see section 28.2.1) 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 28.4.1).

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.

**Toolbox | Transcriptomics Analysis ( ) | 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 28.4.1) created from small RNA samples. Click **Next** when the data is listed in the right-hand side of the dialog.

This dialog (figure 28.21) is where you define the annotation resources to be used.

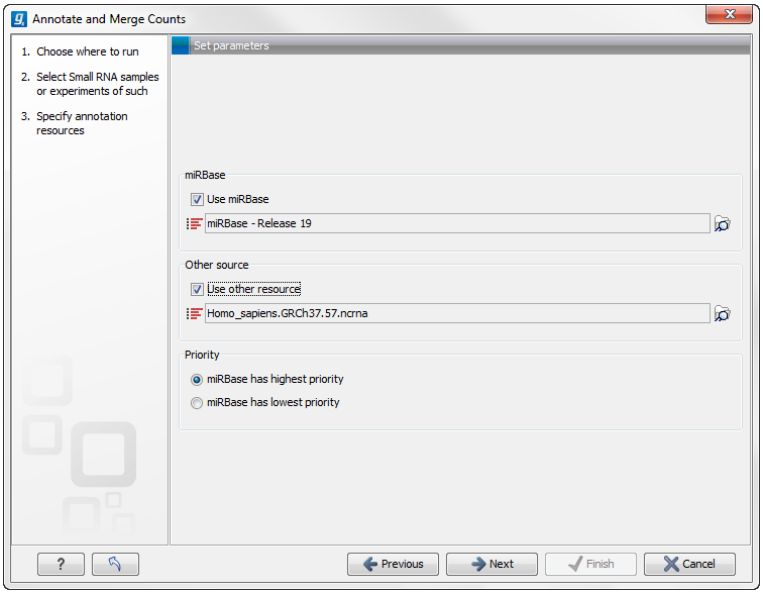



Figure 28.21: *Defining annotation resources.*

There are two ways of providing annotation sources:

- Downloading miRBase using the integrated download tool (explained in section 28.2.2).
- Importing a list of sequences, e.g. from a fasta file. This could be from Ensembl, e.g. `ftp://ftp.ensembl.org/pub/release-57/fasta/homo_sapiens/ncrna/Homo_sapiens.GRCh37.57.ncrna.fa.gz` or from ncRNA.org: `http://www.ncrna.org/frnadb`.

**Note:** 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 28.22).

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 28.21.

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

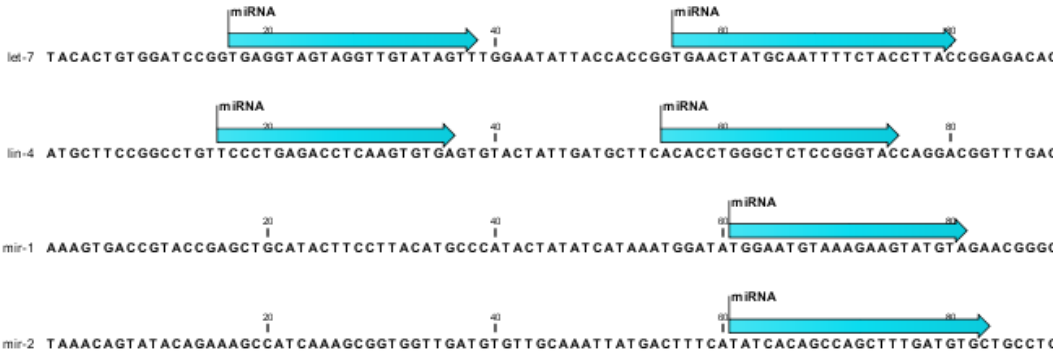



Figure 28.22: *Some of the precursor miRNAs from miRBase have both 3' and 5' mature regions (previously referred to as mature and mature*) annotated (as the two first in this list).*

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 28.23). Note that if you have not selected a miRBase annotation source, you will go directly to the next step shown in figure 28.24.

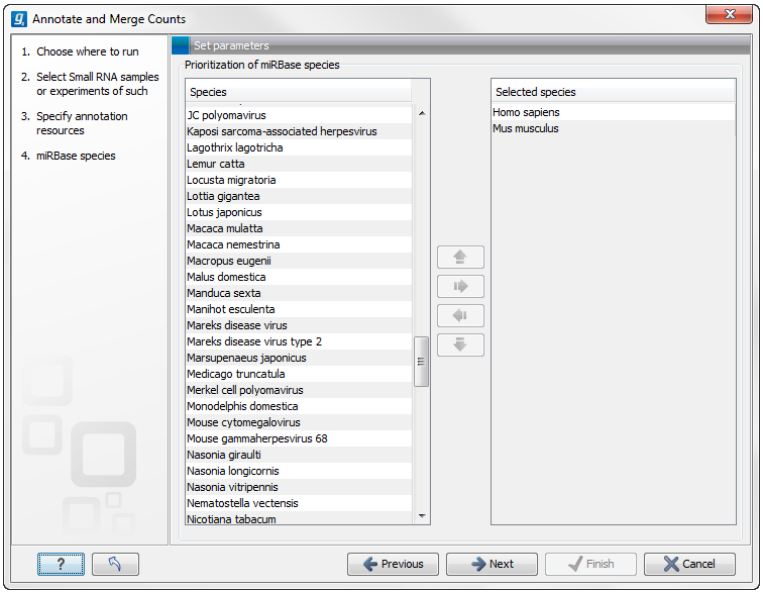



Figure 28.23: *Defining and prioritizing species in miRBase.*

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 28.23, 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 28.24).

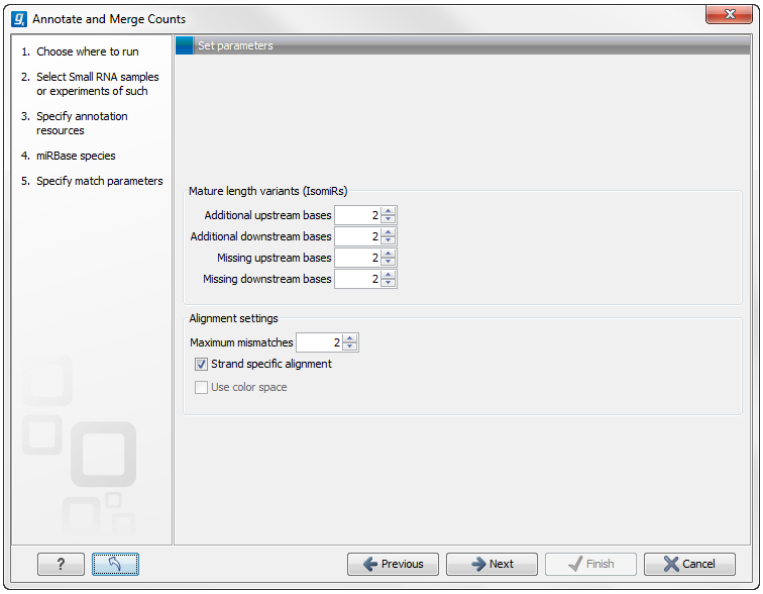



Figure 28.24: *Setting parameters for aligning.*

The panel at the top is active only if you have chosen to annotate with miRBase. It is used to define the requirements to the alignment of a read for it to be counted as a 3' or 5' mature region (previously referred to as mature and mature*) 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). Furthermore, you can specify if the alignment and annotation should be performed in **color space** which is available when your small RNA sample is based on SOLiD data. [5] 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

[5] Note that this option is only going to make a difference for tags with low counts. Since the actual tag counting in the first place is done based on perfect matches, the highly abundant tags are not likely to have sequencing errors, and aligning in color space does not add extra benefit for these.

etc. No gaps are allowed. The number of rounds depend on the number of mismatches allowed[6] (default is two which means three rounds of read mapping, see figure 28.24).

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.

Following the mapping, the tags are classified into the following categories according to where they match.

- Mature 5' exact
- Mature 5' super
- Mature 5' sub
- Mature 5' sub/super
- Mature 3' exact
- Mature 3' super
- Mature 3' sub
- Mature 3' sub/super
- Precursor
- Other

All these categories except *Other* refer to hits in miRBase. For hits on mirBase sequences we distinguish between where on the sequences the tags match. The mirBase sequences may have up to two mature micro RNAs annotated. We refer to a mature miRNA that is located closer (or equally close) to the 5' end than to the 3' end as 'Mature 5''. A mature miRNA that is located closer to the 3' end is referred to as 'Mature 3''. *Exact* means that the tag matches exactly to the annotated mature 5' or 3' region; *Sub* means that the observed tag is shorter than the annotated mature 5' or mature 3'; *super* means that the observed tag is longer than the annotated mature 5' or mature 3'. The combination *sub/super* means that the observed tag extends the annotation in one end and is shorter at the other end. Precursor means that 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). The Other category is for hits in the other resources (the information about resource is also shown in the output).

An example of an alignment is shown in figure 28.25 using the same alignment settings as in figure 28.24.

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'.

[6] For color space, the maximum number of mismatches is 2.

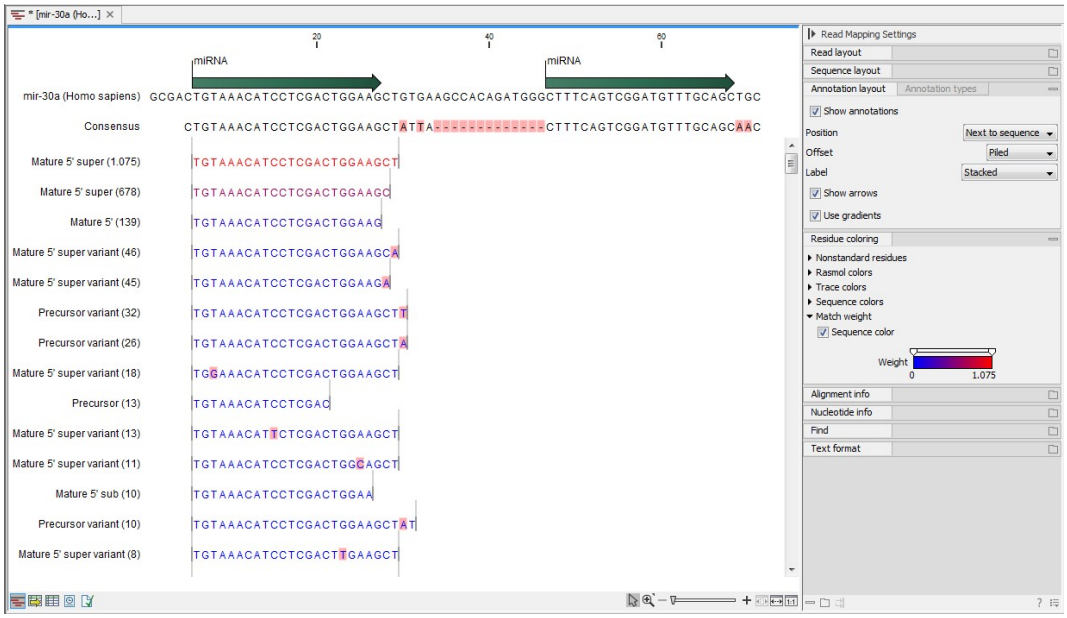



Figure 28.25: *Alignment of length variants of mir-30a.*

If a tag has several hits, the list above is used for prioritization. This means that e.g. a *Mature 5' sub* is preferred over a *Mature 3' exact*. Note that if miRBase was chosen as lowest priority (figure 28.21), 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* (e.g. 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) the following rules for adding annotations apply:

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 28.26.

The options are:

**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 28.2.5. 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 28.2.4. 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,

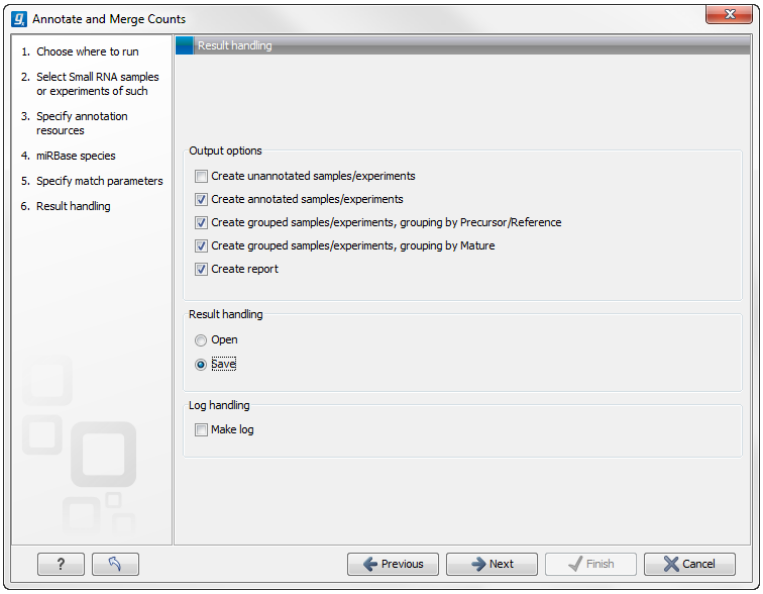



Figure 28.26: *Output options.*

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.

**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 e.g. 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 28.2.4. 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 28.2.4. 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. 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.** A summary report described below.

The summary report includes the following information (an example is shown in figure 28.27):

**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 28.27: *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 28.28.

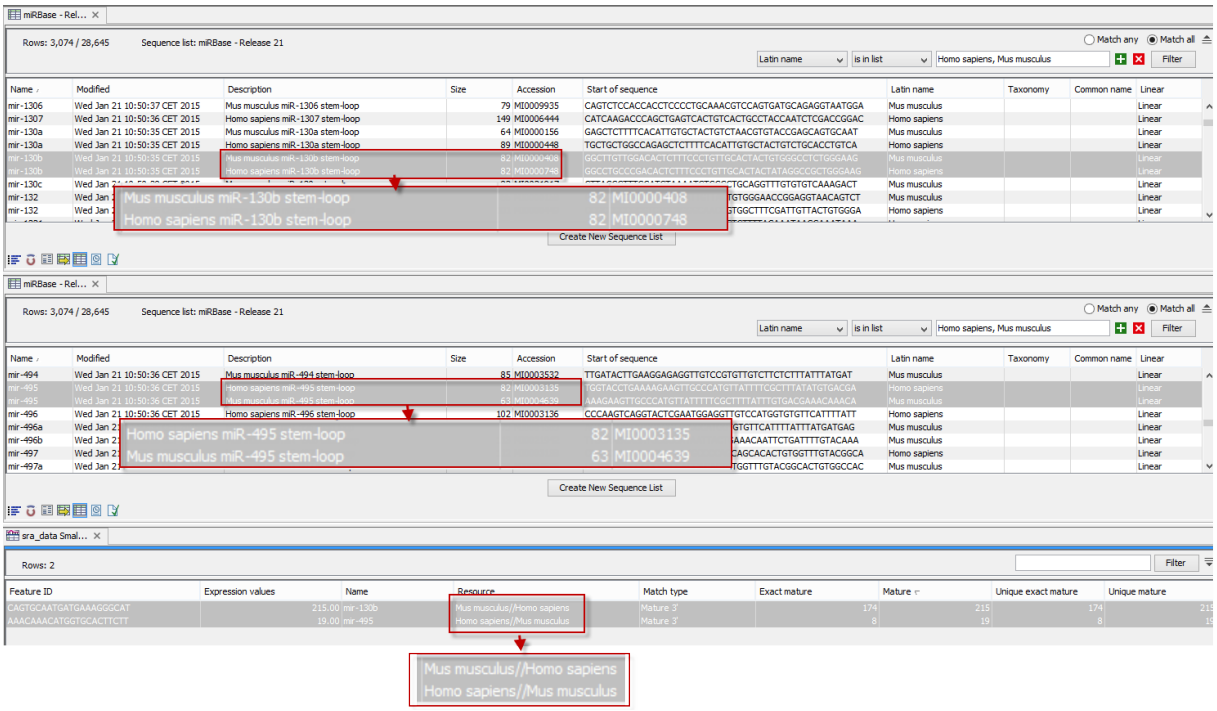



Figure 28.28: *The results table (at the bottom of the figure) shown in split view with the imported miRBase data shown twice to illustrate that the order in which the resource is listed is determined by the miRBase accession number. Due to the very small fonts in the table, the most important information has been magnified (red box).*

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.

### 28.2.4 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 8.3.

#### The un-grouped sample

An example of an un-grouped annotated sample is shown in figure 28.29.

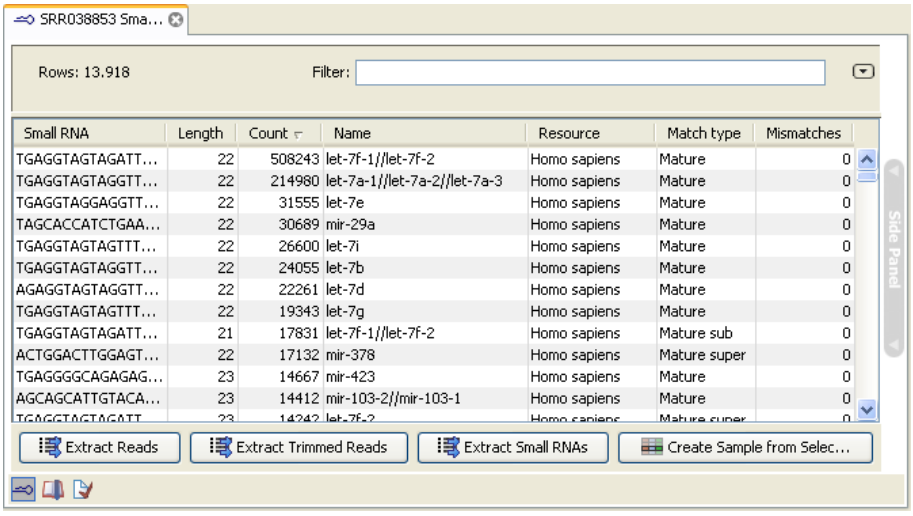



Figure 28.29: *An ungrouped annotated sample.*

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 28.30 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).

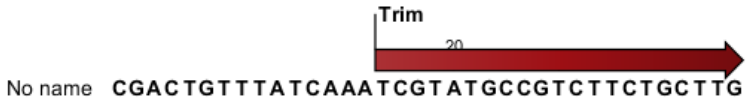



Figure 28.30: *Extracting reads from a sample.*

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 28.31.

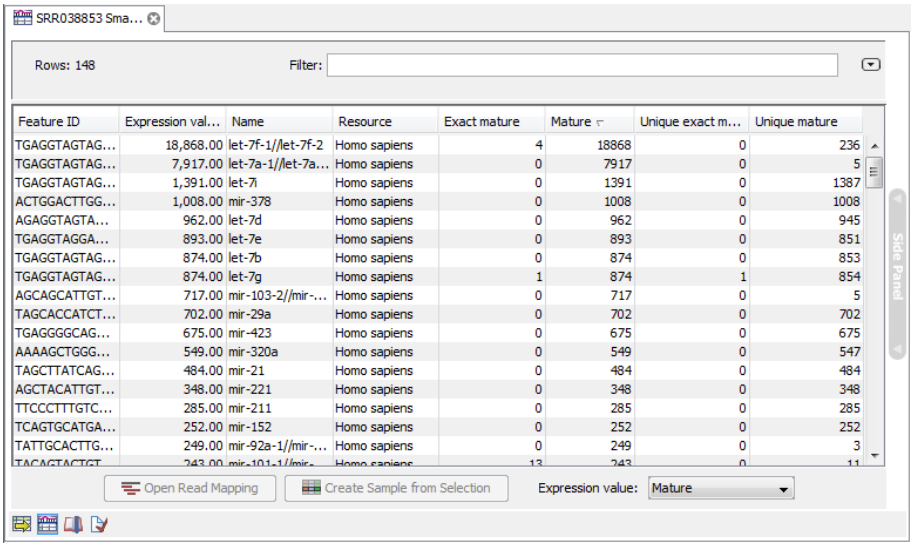



Figure 28.31: *A sample grouped on mature 5' miRNAs.*

The contents of the table are explained in section 28.2.3. 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 28.32. 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.

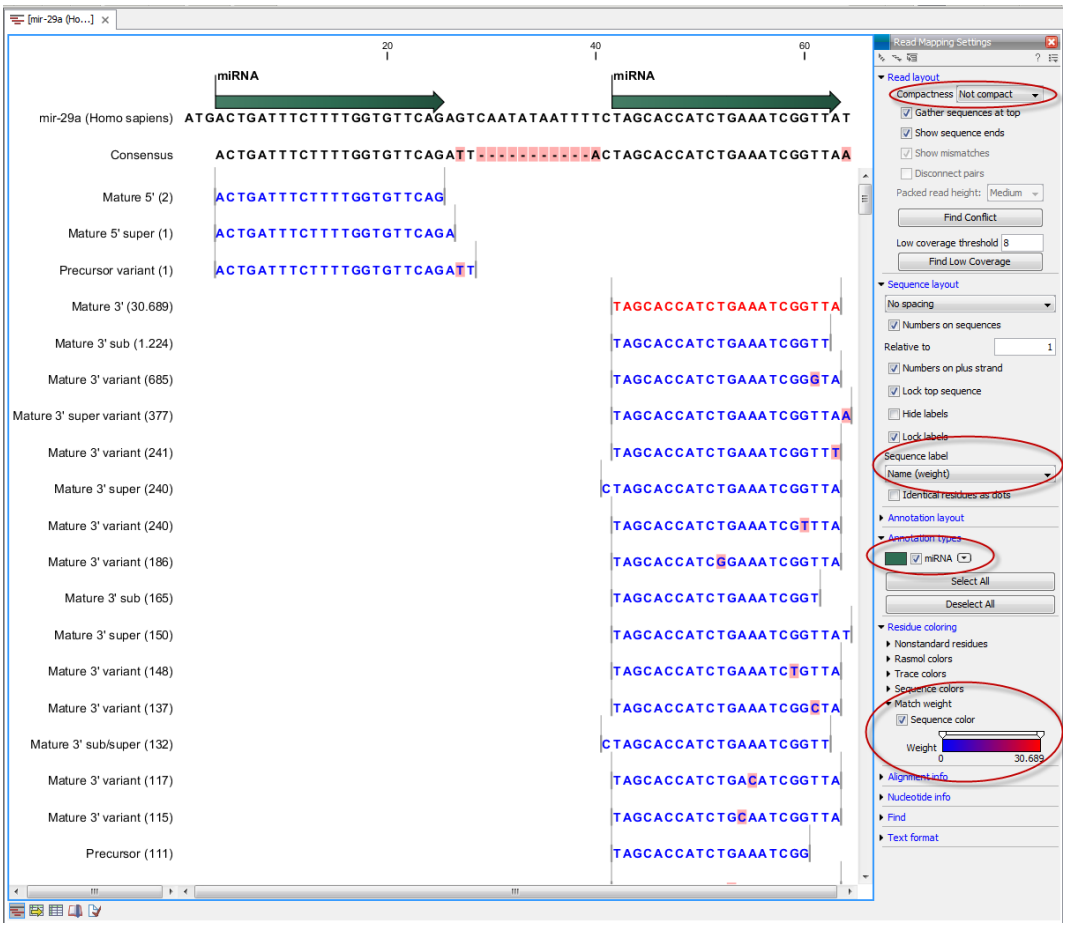



Figure 28.32: *Aligning all the variants of this miRNA from miRBase, providing a visual overview of the distribution of tags along the precursor sequence.*

### 28.2.5 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.

## 28.3 Expression profiling by tags

Expression profiling by tags, also known as *tag profiling* or *tag-based transcriptomics*, is an extension of Serial analysis of gene expression (SAGE) using next-generation sequencing technologies. With respect to sequencing technology it is similar to RNA-seq (see section 28.1), but with tag profiling, you do not sequence the mRNA in full length. Instead, small tags are extracted from each transcript, and these tags are then sequenced and counted as a measure of the abundance of each transcript. In order to tell which gene's expression a given tag is measuring, the tags are often compared to a virtual tag library. This consists of the 'virtual' tags that would have been extracted from an annotated genome or a set of ESTs, had the same protocol been applied to these. For a good introduction to tag profiling including comparisons with different micro array platforms, we refer to ['t Hoen et al., 2008]. For more in-depth information, we refer to [Nielsen, 2007].

Figure 28.33 shows an example of the basic principle behind tag profiling. There are variations of this concept and additional details, but this figure captures the essence of tag profiling, namely the extraction of a tag from the mRNA based on restriction cut sites.

The *CLC Genomics Workbench* supports the entire tag profiling data analysis work flow following the sequencing:

- Extraction of tags from the raw sequencing reads (tags from different samples are often barcoded and sequenced in one pool).
- Counting tags including a sequencing-error correction algorithm.
- Creating a virtual tag list based on an annotated reference genome or an EST-library.
- Annotating the tag counts with gene names from the virtual tag list.

Each of the steps in the work flow are described in details below.

### 28.3.1 Extract and count tags

First step in the analysis is to import the data (see section 6.3).

The next step is to extract the tags and count them:

**Toolbox** | **Transcriptomics Analysis ( )** | **Expression Profiling by Tags ( )** | **Extract and Count Tags ( )**

This will open a dialog where you select the reads that you have imported. Click **Next** when the sequencing data is listed in the right-hand side of the dialog.

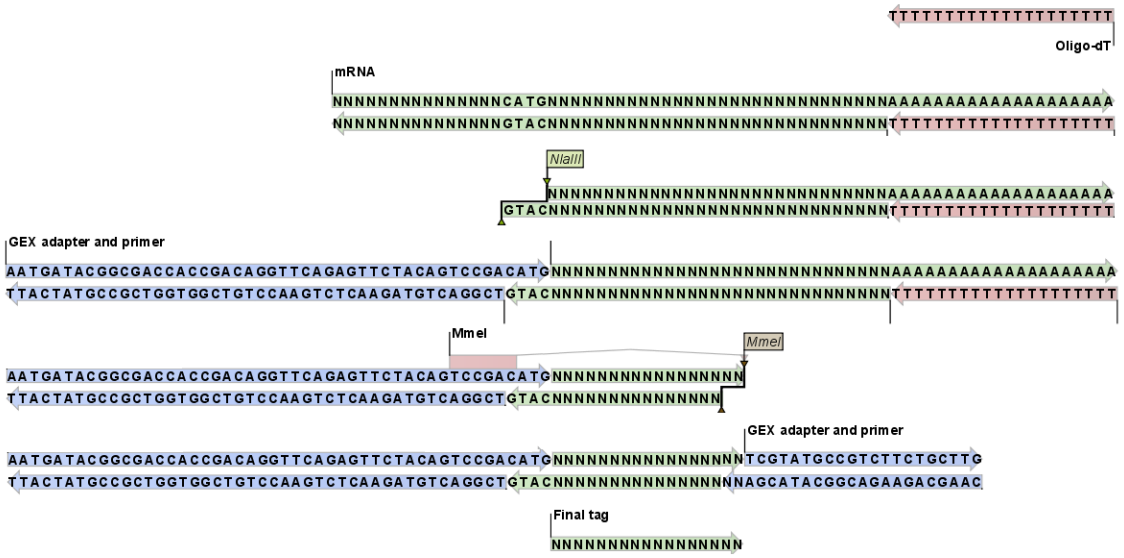



Figure 28.33: *An example of the tag extraction process. 1+2. Oligo-dT attached to a magnetic bead is used to trap mRNA. 3. The enzyme* NlaIII *cuts at CATG sites and the fragments not attached to the magnetic bead are removed. 4. An adapter is ligated to the GTAC overang. 5. The adapter includes a recognition site for* MmeI *which cuts 17 bases downstream. 6. Another adapter is added and the sequence is now ready for amplification and sequencing. 7. The final tag is 17 bp. The example is inspired by ['t Hoen et al., 2008].*

This dialog is where you define the elements in your reads. An example is shown in figure 28.34.

By defining the order and size of each element, the Workbench is now able to both separate samples based on bar codes and extract the tag sequence (i.e. removing linkers, bar codes etc). The elements available are:

**Sequence** This is the part of the read that you want to use as your final tag for counting and annotating. If you have tags of varying lengths, add a spacer afterwards (see below).

**Sample keys** Here you input a comma-separated list of the sample keys used for identifying the samples (also referred to as "bar codes"). If you have not pooled and bar coded your data, simply omit this element.

**Linker** This is a known sequence that you know should be present and do not want to be included in your final tag.

**Spacer** This is also a sequence that you do not want to include in your final tag, but whereas the linker is defined by its sequence, the spacer is defined by its length. Note that the length defines the maximum length of the spacer. Often not all tags will be exactly the same length, and you can use this spacer as a buffer for those tags that are longer than what you have defined as your sequence. In the example in figure 28.34, the tag length is 17 bp, but a spacer is added to allow tags up to 19 bp. Note that the part of the read that

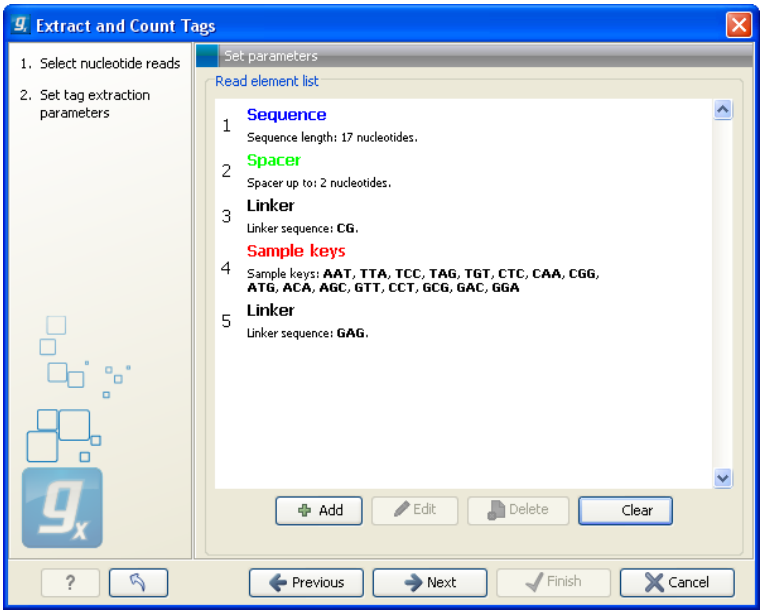



Figure 28.34: *Defining the elements that make up your reads.*

is extracted and used as the final tag does not include the spacer sequence. In this way you homogenize the tag lengths which is usually desirable because you want to count short and long tags together.

When you have set up the right order of your elements, click **Next** to set parameters for counting tags as shown in figure 28.35.

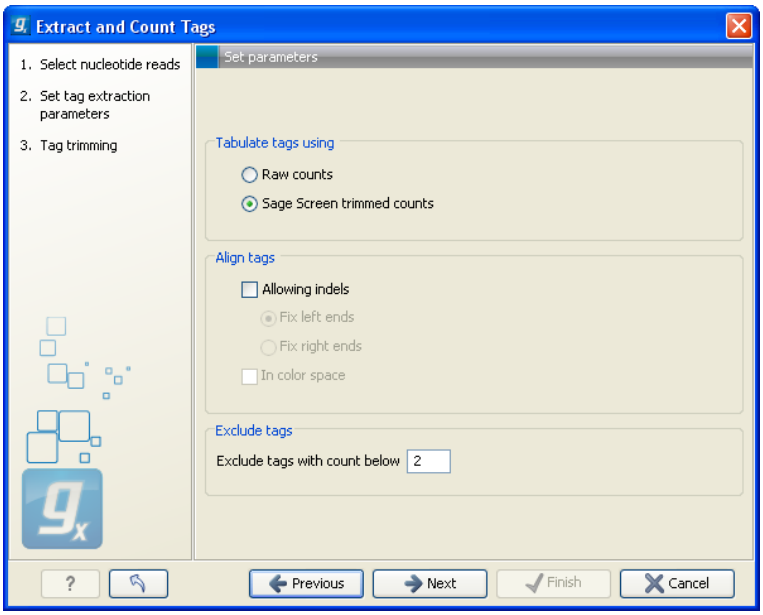



Figure 28.35: *Setting parameters for counting tags.*

At the top, you can specify how to tabulate (i.e. count) the tags:

**Raw counts** This will produce the count for each tag in the data.

**Sage Screen trimmed counts** This will produce trimmed tag counts. The trimmed tag counts

are obtained by applying an implementation of the SAGEscreen method ( [Akmaev and Wang, 2004]) to the raw tag counts. In this procedure, raw counts are trimmed using probabilistic reasoning. In this procedure, if a tag with low count has a neighboring tag with high count, and it is likely, based on the estimated mutation rate, that the low count tags have arisen through sequencing errors of the tags with higher count, the count of the less abundant tag will be attributed to the higher abundant neighboring tag. The implementation of the SAGEscreen method is highly efficient and provides considerable speed and memory improvements.

Next, you can specify additional parameters for the alignment that takes place when the tags are tabulated:

**Allowing indels** Ticking this box means that, when SAGEscreen is applied, neighboring tags will, in addition to tags which differ by nucleotide substitutions, also include tags with insertion or deletion differences.

**Color space** This option is only available if you use data generated on the SOLiD platform. Checking this option will perform the alignment in color space which is desirable because sequencing errors can be corrected. Learn more about color space in section 25.4.

At the bottom you can set a minimum threshold for tags to be reported. Although the SAGEscreen trimming procedure will reduce the number of erroneous tags reported, the procedure only handles tags that are neighbors of more abundant tags. Because of sequencing errors, there will be some tags that show extensive variation. There will by chance only be a few copies of these tags, and you can use the minimum threshold option to simply discard tags. The default value is two which means that tags only occurring once are discarded. This setting is a trade-off between removing bad-quality tags and still keeping tags with very low expression (the ability to measure low levels of mRNA is one of the advantages of tag profiling over for example micro arrays [’t Hoen et al., 2008]).

**Note!** If more samples are created, SAGEscreen and the minimum threshold cut-offs will be applied to the cumulated counts (i.e. all tags for all samples).

Clicking **Next** allows you to specify the output of the analysis as shown in figure 28.36.

The options are:

**Create expression samples with tag counts** This is the primary result showing all the tags and respective counts (an example is shown in figure 28.37). For each sample defined via the bar codes, there will be an expression sample like this. Note that all samples have the same list of tags, even if the tag is not present in the given sample (i.e. there will be tags with count 0 as shown in figure 28.37). The expression samples can be used in further analysis by the expression analysis tools for statistical analyses etc.

**Create sequence lists of extracted tags** This is a simple sequence list of all the tags that were extracted. The list is simple with no counts or additional information.

**Create list of reads which have no tags** This list contains the reads from which a tag could not be extracted. This is most likely bad quality reads with sequencing errors that make them impossible to group by their bar codes. It can be useful for troubleshooting if the amount of real tags is smaller than expected.

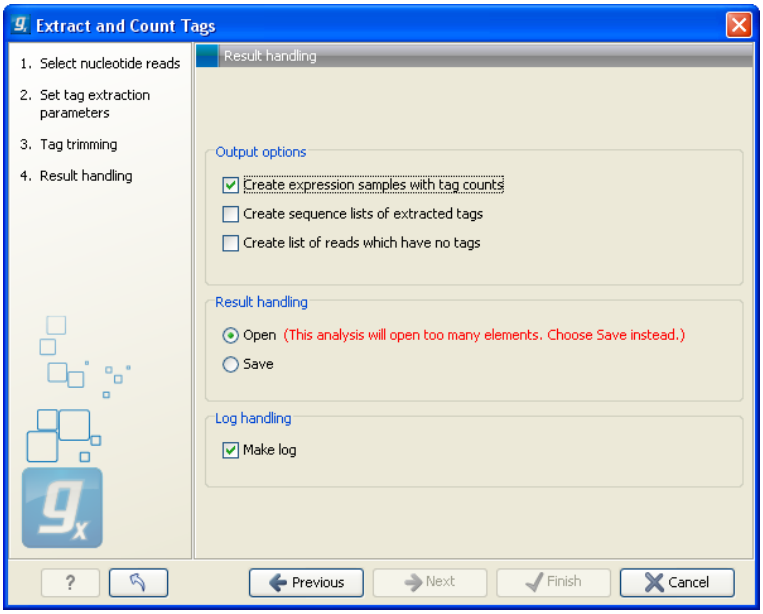



Figure 28.36: *Output options.*

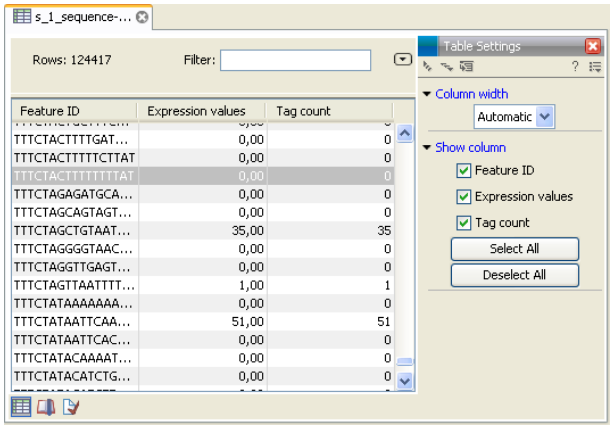



Figure 28.37: *The tags have been extracted and counted.*

Finally, a log can be shown of the extraction and count process. The log gives useful information such as the number of tags in each sample and the number of reads without tags.

### 28.3.2 Create virtual tag list

Before annotating the tag sample ( ) created above, you need to create a so-called *virtual tag list*. The list is created based on a DNA sequence or sequence list holding, an annotated genome or a list of ESTs. It represents the tags that you would expect to find in your experimental data (given the reference genome or EST list reflects your sample). To create the list, you specify the restriction enzyme and tag length to be used for creating the virtual list.

The virtual tag list can be saved and used to annotate experiments made from tag-based expression samples as shown in section 28.3.3.

To create the list:

**Toolbox** | **Transcriptomics Analysis ( )** | **Expression Profiling by Tags ( )** | **Create Virtual Tag List ( )**

This will open a dialog where you select one or more annotated genomic sequences or a list of ESTs. Click **Next** when the sequences are listed in the right-hand side of the dialog.

This dialog is where you specify the basis for extracting the virtual tags (see figure 28.38).

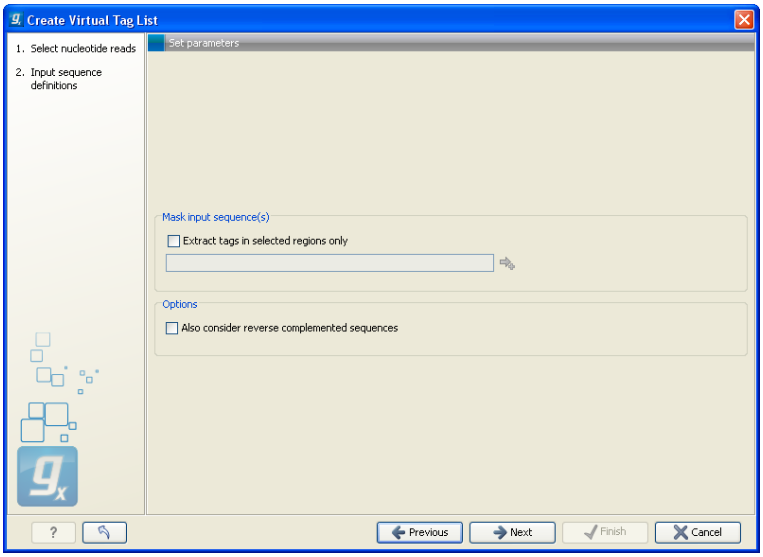



Figure 28.38: *The basis for the extraction of reads.*

At the top you can choose to extract tags based on annotations on your sequences by checking the **Extract tags in selected areas only** option. This option is applicable if you are using annotated genomes (e.g. Refseq genomes). Click the small button ( ) to the right to display a dialog showing all the annotation types in your sequences. Select the annotation type representing your transcripts (usually *mRNA* or *Gene*). The sequence fragments covered by the selected annotations will then be extracted from the genomic sequence and used as basis for creating the virtual tag list.

If you use a sequence list where each sequence represents your transcript (e.g. an EST library), you should not check the **Extract tags in selected areas only** option.

Below, you can choose to include the reverse complement for creating virtual tags. This is mainly used if there is uncertainty about the orientation of sequences in an EST library.

Clicking **Next** allows you to specify enzymes and tag length as shown in figure 28.39.

At the top, find the enzyme used to define your tag and double-click to add it to the panel on the right (as it has been done with *NlaIII* in figure 28.39). You can use the filter text box so search for the enzyme name.

Below, there are further options for the tag extraction:

**Extract tags** When extracting the virtual tags, you have to decide how to handle the situation where one transcript has several cut sites. In that case there would be several potential tags. Most tag profiling protocols extract the 3'-most tag (as shown in the introduction in figure 28.33), so that would be one way of defining the tags in the virtual tag list. However, due to non-specific cleavage, new alternative splicing or alternative polyadenylation ['t Hoen et al., 2008], tags produced from internal cut sites of the transcript are also quite frequent. This means that it is often not enough to consider the 3'-most restriction site only. The list lets you select either **All**, **External 3'** which is the 3'-most tag or **External 5'** which is the

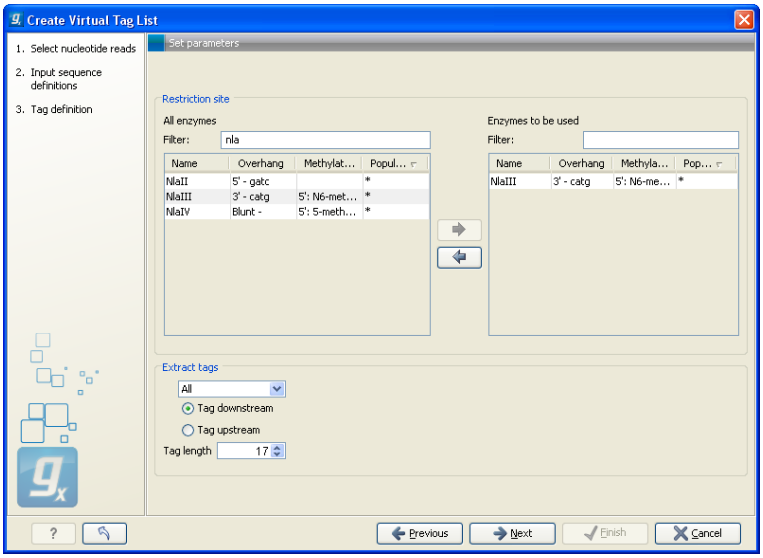



Figure 28.39: *Defining restriction enzyme and tag length.*

5' most tag (used by some protocols, for example CAGE - cap analysis of gene expression - see [Maeda et al., 2008]). The result of the analysis displays whether the tag is found at the 3' end or if it is an internal tag (see more below).

**Tag downstream/upstream** When the cut site is found, you can specify whether the tag is then found downstream or upstream of the site. In figure 28.33, the tag is found downstream.

**Tag length** The length of the tag to be extracted. This should correspond to the sequence length defined in figure 28.34.

Clicking **Next** allows you to specify the output of the analysis as shown in figure 28.40.

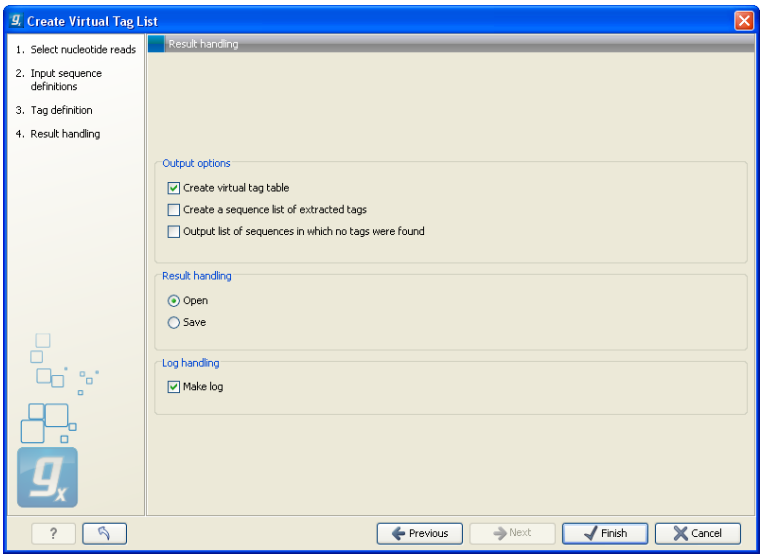



Figure 28.40: *Output options.*

The output options are:

**Create virtual tag table** This is the primary result listing all the virtual tags. The table is explained in detail below.

**Create a sequence list of extracted tags** All the extracted tags can be represented in a raw sequence list with no additional information except the name of the transcript. You can e.g. **Export** (  ) this list to a fasta file.

**Output list of sequences in which no tags were found** The transcripts that do not have a cut site or where the cut site is so close to the end that no tag could be extracted are presented in this list. The list can be used to inspect which transcripts you could potentially fail to measure using this protocol. If there are tags for all transcripts, this list will not be produced.

In figure 28.41 you see an example of a table of virtual tags that have been produced using the **3' external** option described above.

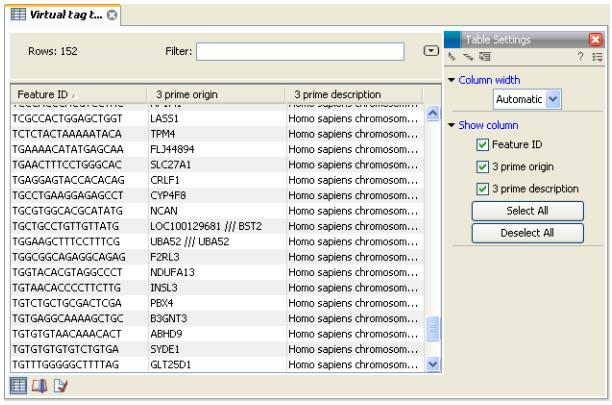



Figure 28.41: *A virtual tag table of 3' external tags.*

The first column lists the tag itself. This is the column used when you annotate your tag count samples or experiments (see section 28.3.3). Next follows the name of the tag's origin transcript. Sometimes the same tag is seen in more than one transcript. In that case, the different origins are separated by /// as it is the case for the tag of *LOC100129681 /// BST2* in figure 28.41. The row just below, *UBA52*, has the same name listed twice. This is because the analysis was based on mRNA annotations from a Refseq genome where each splice variant has its own mRNA annotation, and in this case the *UBA52* gene has two mRNA annotations including the same tag.

The last column is the description of the transcript (which is either the sequence description if you use a list of un-annotated sequences or all the information in the annotation if you use annotated sequences).

The example shown in figure 28.41 is the simplest case where only the 3' external tags are listed. If you choose to list **All** tags, the table will look like figure 28.42.

In addition to the information about the 3' tags, there are additional columns for 5' and internal tags. For the internal tags there is also a numbering, see for example the top row in figure 28.42 where the *TMEM16H* tag is tag number 3 out of 16. This information can be used to judge how close to the 3' end of the transcript the tag is. As mentioned above, you would often expect to sequence more tags from cut sites near the 3' end of the transcript.

If you have chosen to include reverse complemented sequences in the analysis, there will be an additional set of columns for the tags of the other strand, denoted with a (-).

You can use the advanced table filtering (see section 8.3) to interrogate the number of tags with specific origins (e.g. define a filter where 3' origin != and then leave the text field blank).

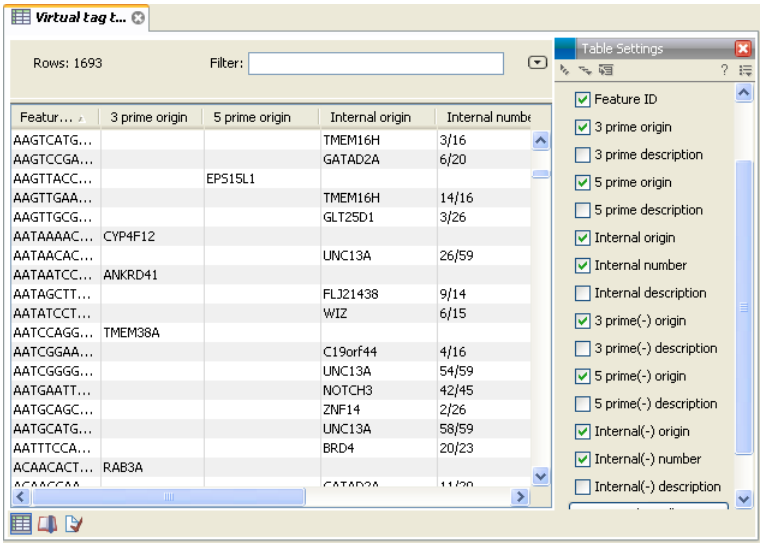



Figure 28.42: *A virtual tag table where all tags have been extracted. Note that some of the columns have been ticked off in the Side Panel.*

### 28.3.3 Annotate tag experiment

Combining the tag counts ( ) from the experimental data (see section 28.3.1) with the virtual tag list ( ) (see above) makes it possible to put gene or transcript names on the tag counts. The Workbench simply compares the tags in the experimental data with the virtual tags and transfers the annotations from the virtual tag list to the experimental data.

This is done on an experiment level (experiments are collections of samples with defined groupings, see section 28.4):

> **Toolbox** | **Transcriptomics Analysis ( )** | **Expression Profiling by Tags ( )** | **Annotate Tag Experiment ( )**

You can also access this functionality at the bottom of the **Experiment table** ( ) as shown in figure 28.43.

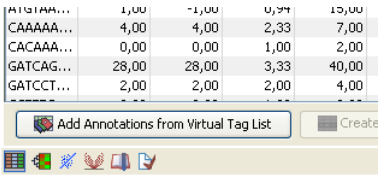



Figure 28.43: *You can annotate an experiment directly from the experiment table.*

This will open a dialog where you select a virtual tag list ( ) and an experiment ( ) of tag-based samples. Click **Next** when the elements are listed in the right-hand side of the dialog.

This dialog lets you choose how you want to annotate your experiment (see figure 28.44).

If a tag in the virtual tag list has more than one origin (as shown in the example in figure 28.42) you can decide how you want your experimental data to be annotated. There are basically two options:

**Annotate all** This will transfer all annotations from the virtual tag. The type of origin is still preserved so that you can see if it is a 3' external, 5' external or internal tag.

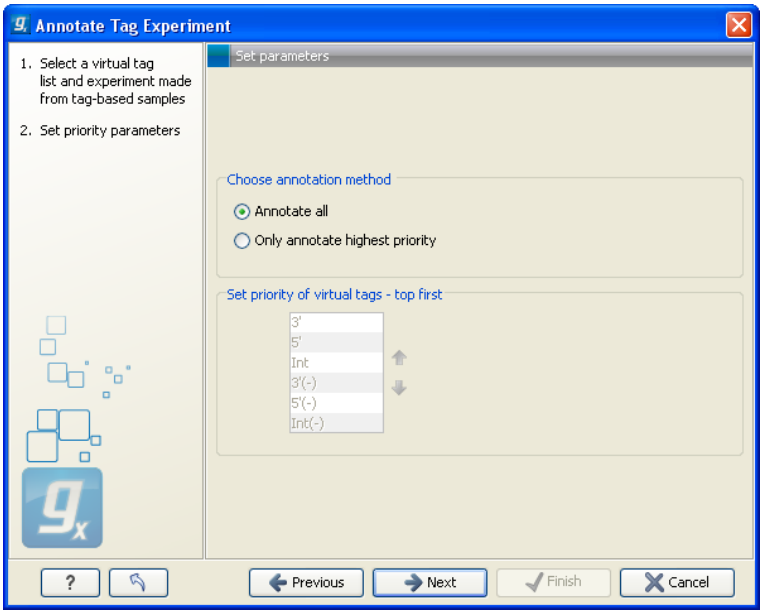



Figure 28.44: *Defining the annotation method.*

**Only annotate highest priority** This will look for the highest priority annotation and only add this to the experiment. This means that if you have a virtual tag with a 3' external and an internal tag, only the 3' external tag will be annotated (using the default prioritization). You can define the prioritization yourself in the table below: simply select a type and press the **up** ( ) and **down** ( ) arrows to move it up and down in the list. Note that the priority table is only active when you have selected **Only annotate highest priority**.

Click **Next** to choose how you want to tags to be aligned (see figure 28.45). When the tags

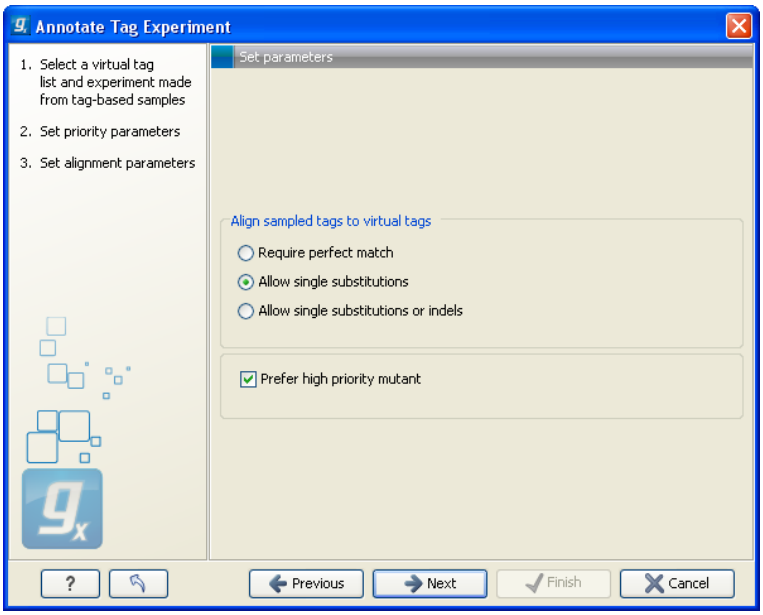



Figure 28.45: *Settings for aligning the tags.*

from the virtual tag list are compared to your experiment, the tags are matched using one of the following options:

```
Tag from experiment:                          CGTATCAATCGATTAC
                                              ||||||||||||||||
Tag1 from virtual tag list (internal):        CGTATCAATCGATTAC
                                              | ||||||||||||||
Tag1 from virtual tag list (3' external):     CCTATCAATCGATTAC
```

**Require perfect match** The tags need to be identical to be matched.

**Allow single substitutions** If there is up to one mismatch in the alignment, the tags will still be matched. If there is a perfect match, single substitutions will not be considered.

**Allow single substitutions or indels** Similar to the previous option, but now single-base insertions and deletions are also allowed. Perfect matches are preferred to single-base substitutions which are preferred to insertions, which are again preferred to deletions. [7]

If you select any of the two options allowing mismatches or mismatches and indels, you can also choose to **Prefer high priority mutant**. This option is only available if you have chosen to annotate highest priority only in the previous step (see figure 28.44). The option is best explained through an example: In this case, you have a tag that matches perfectly to an internal tag from the virtual tag list. Imagine that in this example, you have prioritized the annotation so that 3' external tags are of higher priority than internal tags. The question is now if you want to accept the perfect match (of a low priority virtual tag) or the high-priority virtual tag with one mismatch? If you check the **Prefer high priority mutant**, the 3' external tag in the example above will be used rather than the perfect match.

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**. . This will add extra annotation columns to the experiment. The extra columns corresponds to the columns found in your virtual tag list. If you have chosen to annotate highest priority-only, there will only be information from one origin-column for each tag as shown in figure 28.46.

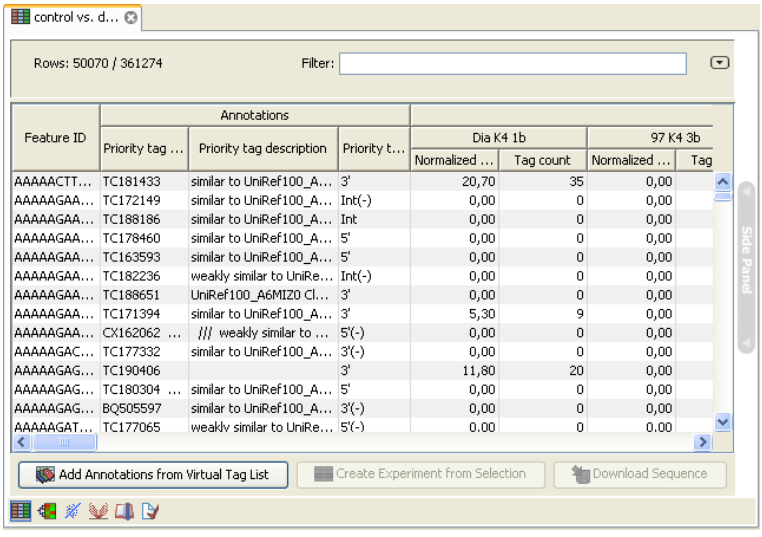



Figure 28.46: *An experiment annotated with prioritized tags.*

The *CLC Genomics Workbench* is able to analyze expression data produced on microarray platforms and high-throughput sequencing platforms (also known as Next-Generation Sequencing

[7] Note that if you use color space data, only color errors are allowed when choosing anything but perfect match.

platforms). The *CLC Genomics Workbench* provides 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 used to aid the interpretation of the results.

## 28.4 Experimental design

In order to make full use of the various tools for interpreting expression data, you need to know the central concepts behind the way the data is organized in the *CLC Genomics Workbench*.

The first piece of data you are faced with is the **sample**. In the Workbench, a sample contains the expression values from either one array or from sequencing data of one sample.

Note that the calculation of expression levels based on the raw sequence data is described in section 28.1, section 28.2 and section 28.3.

See more below on how to get your expression data into the Workbench as samples in section L.

In a sample, there is a number of **features**, usually genes, and their associated expression levels.

To analyze differential expression, you need to tell the workbench how the samples are related. This is done by setting up an **experiment**. An experiment is essentially a set of samples which are grouped. By creating an experiment defining the relationship between the samples, it becomes possible to do statistical analysis to investigate differential expression between the groups. The **Experiment** is also used to accumulate calculations like t-tests and clustering because this information is closely related to the grouping of the samples.

### 28.4.1 Setting up an experiment

To set up an experiment:

**Toolbox** | **Transcriptomics Analysis (**  **)**| **Set Up Experiment (**  **)**

Select the samples that you wish to use by double-clicking or selecting and pressing the **Add** (  ) button (see figure 28.47).

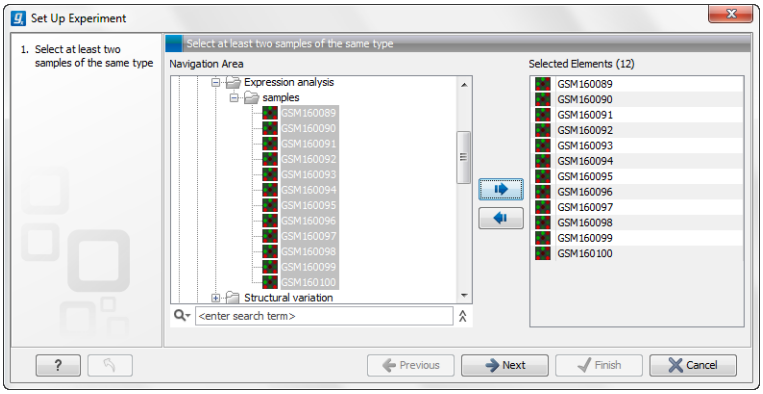



Figure 28.47: *Select the samples to use for setting up the experiment.*

Note that we use "samples" as the general term for both microarray-based sets of expression values and sequencing-based sets of expression values (e.g. an expression track from RNA-Seq).

Clicking **Next** shows the dialog in figure 28.48.

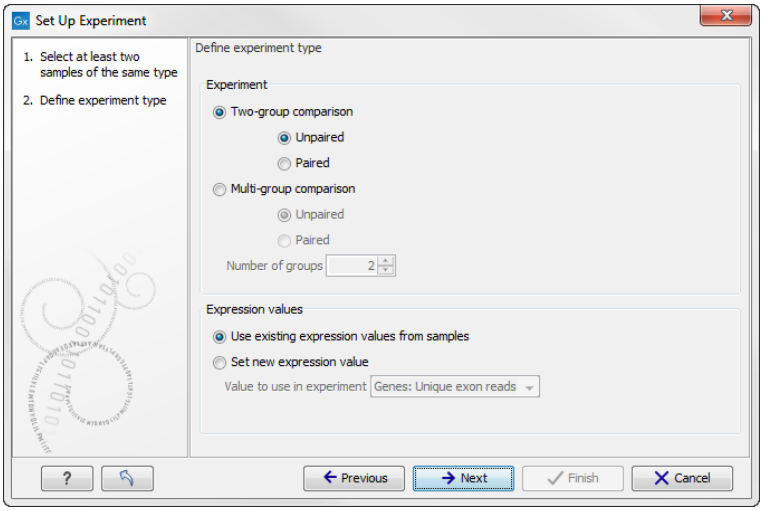



Figure 28.48: *Defining the number of groups and expression value type.*

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.

- **Expression values.** For RNA-Seq experiments, you can also choose which expression value to be used when setting up the experiment. This value will then be used for all subsequent analyses. 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 — 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 28.1.6.
- **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 (see the description of the "Maximum number of hits for a read" option, 28.1.1). 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 28.49.

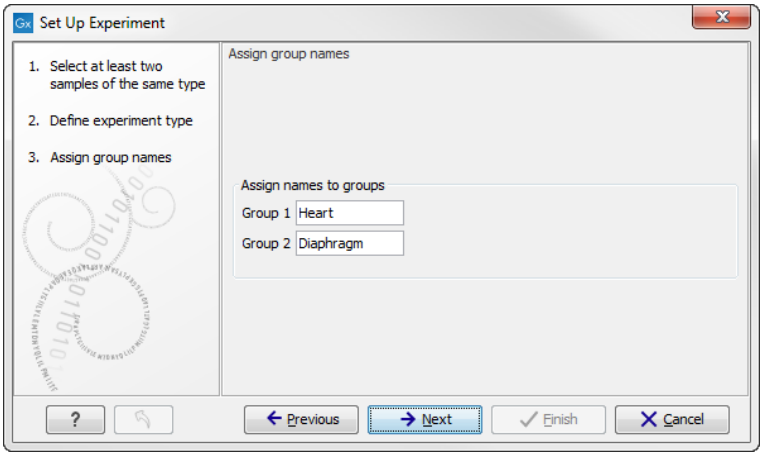



Figure 28.49: *Naming the groups.*

Depending on the number of groups selected in figure 28.48, 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 28.50.

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.

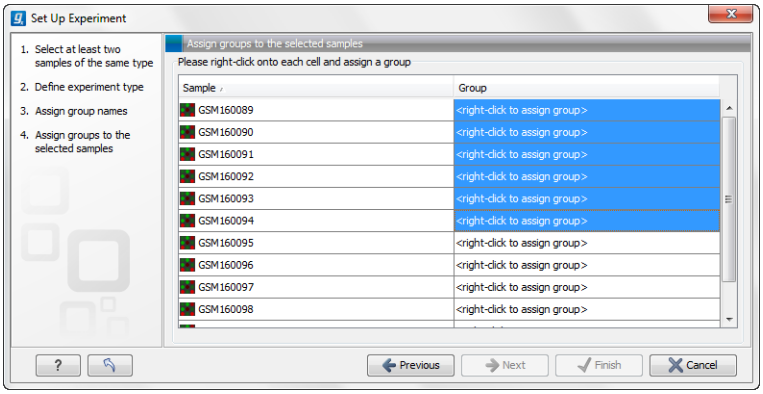



Figure 28.50: *Putting the samples into groups.*

Note that the samples are sorted alphabetically based on their names.

If you have chosen **Paired** in figure 28.48, 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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

### 28.4.2 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 28.51).

For a general introduction to table features like sorting and filtering, see section 8.3.

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 columns. This is useful if you have many columns. In

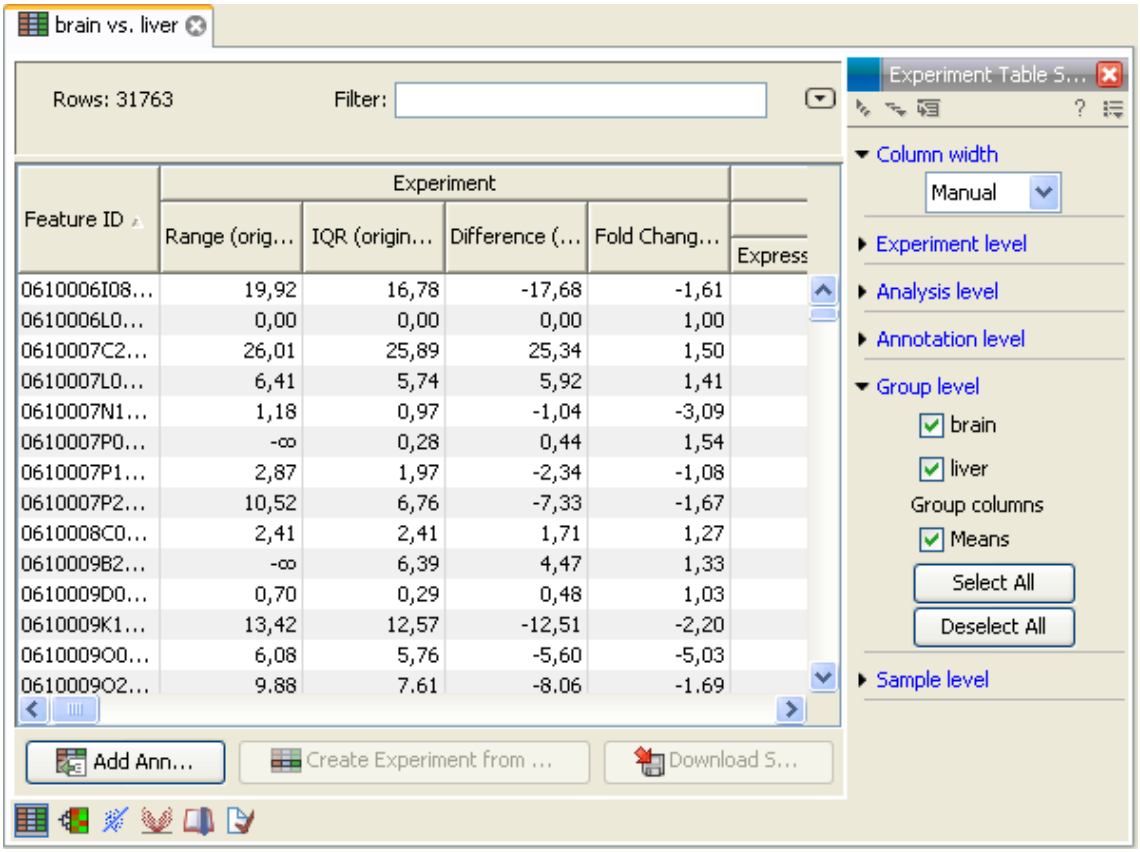



Figure 28.51: *Opening the experiment.*

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 28.52).

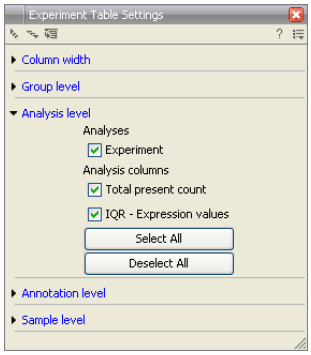



Figure 28.52: *The initial view of the experiment level for a two-group experiment.*

*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. 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 28.6.3 and 28.6.2, 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 28.8.5. 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

If you perform statistical analysis (see section 28.8), there will be a heading for each statistical analysis performed. Under each of these headings you find columns holding relevant values for the analysis (P-value, corrected P-value, test-statistic etc. - see more in section 28.8).

An example of a more elaborate analysis level is shown in figure 28.53.

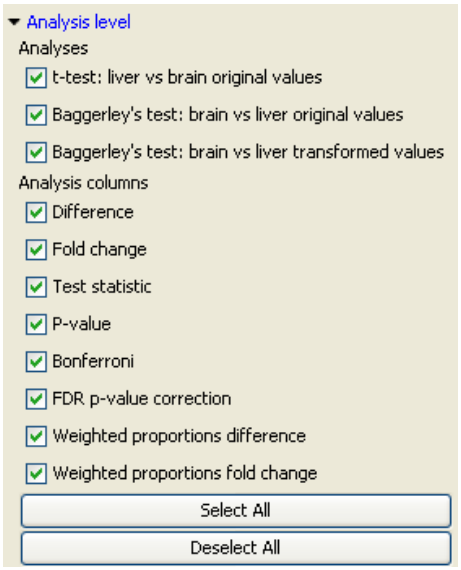



Figure 28.53: *Transformation, normalization and statistical analysis has been performed.*

### Annotation level

If your experiment is annotated (see section 28.4.4), the annotations will be listed in the **Annotation level** group as shown in figure 28.54.

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).

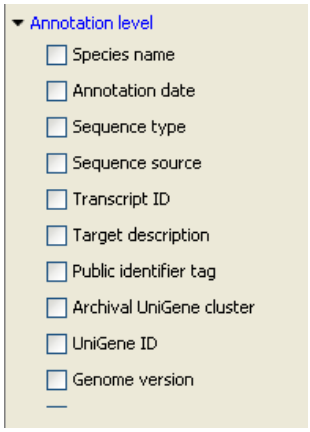



Figure 28.54: *An annotated experiment.*

### Group level

At the group level, you can show/hide entire groups (*Heart* and *Diaphragm* in figure 28.51). 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 28.6.3 and 28.6.2, respectively), the means of the normalized and transformed values will also appear.

### 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 28.6.3 and 28.6.2, respectively), the normalized and transformed values will also appear.

An example is shown in figure 28.55.

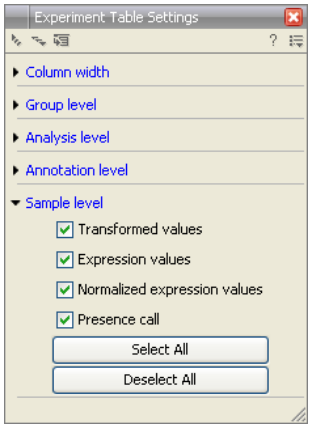



Figure 28.55: *Sample level when transformation and normalization has been performed.*

### 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 8.3).

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 28.56).

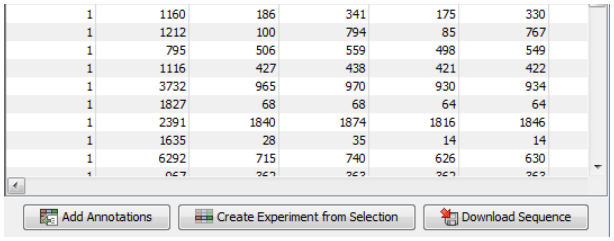



Figure 28.56: *Create a subset of the experiment by clicking the button at the bottom of the experiment table.*

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 28.57).

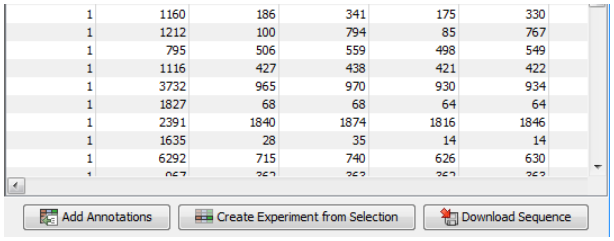



Figure 28.57: *Select sequences and press the download button.*

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 10. You can now use the downloaded sequences for further analysis in the Workbench, e.g. performing BLAST searches and designing primers for QPCR experiments.

### 28.4.3 Visualizing RNA-Seq read tracks for the experiment

When working with RNA-Seq data, the experiment can be used to browse the read mappings to investigate how the reads supporting each sample are mapped. This is done by creating a track list:

**File** | **New** | **Track List ( )**

Select the mapping and expression tracks of the samples you wish to visualize together and select any annotation tracks (e.g. gene and mRNA) to be included for visualization **Finish**.

Once the track list is shown, create a split view or drag the tab of the view on to a second screen (if you have two screens). 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 28.58. Note that the **Zoom to selection** ( ) button can be used to adjust the zoom level to fit the region selection.

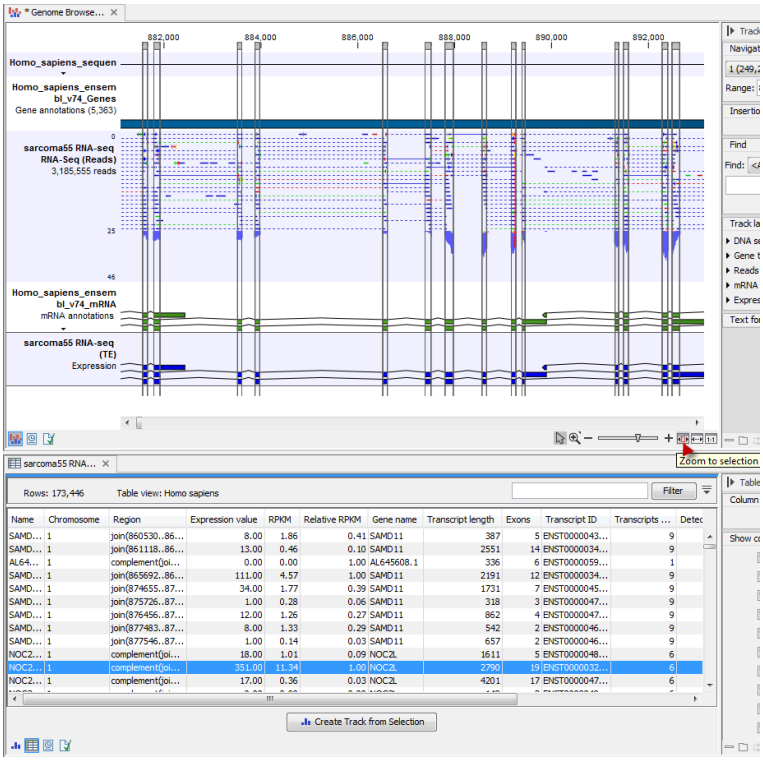



Figure 28.58: *RNA-Seq results shown in a split view with an experiment table at the bottom and a track list with read mappings of several samples at the top.*

Please note that at least one of the expression tracks used in the experiment have to be included in the track list in order for the link between the two to work.

### 28.4.4 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 L 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 L. In order to associate an annotation file with an experiment, either select the annotation file when you set up the experiment (see section 28.4.1), or click:

**Toolbox** | **Transcriptomics 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 28.4.2. You can also add annotations by pressing the **Add Annotations** ( ) button at the bottom of the table (see figure 28.59).

This will bring up a dialog where you can select the annotation file that you have imported

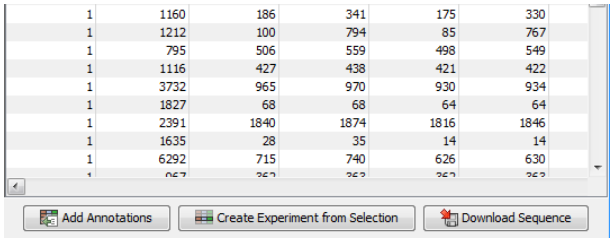



Figure 28.59: *Adding annotations by clicking the button at the bottom of the experiment table.*

together with the experiment you wish to annotate. Click **Next** to specify settings as shown in figure 28.60).

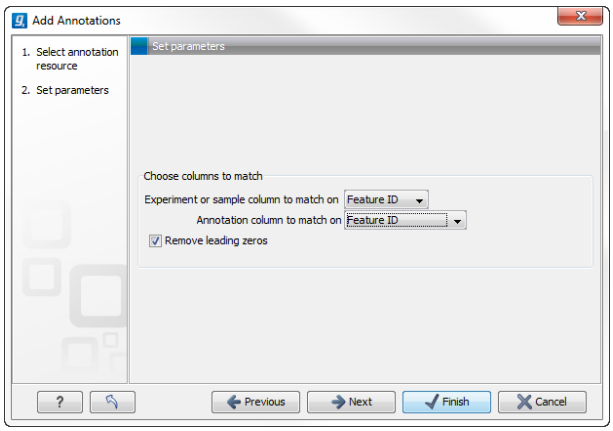



Figure 28.60: *Choosing how to match annotations with samples.*

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.

### 28.4.5 Scatter plot view of an experiment

At the bottom of the experiment table, you can switch between different views of the experiment (see figure 28.61).

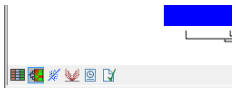


Figure 28.61: *An experiment can be viewed in several ways.*

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 28.62.

In the **Side Panel** to the left, there are a number of options to adjust this view. Under **Graph**

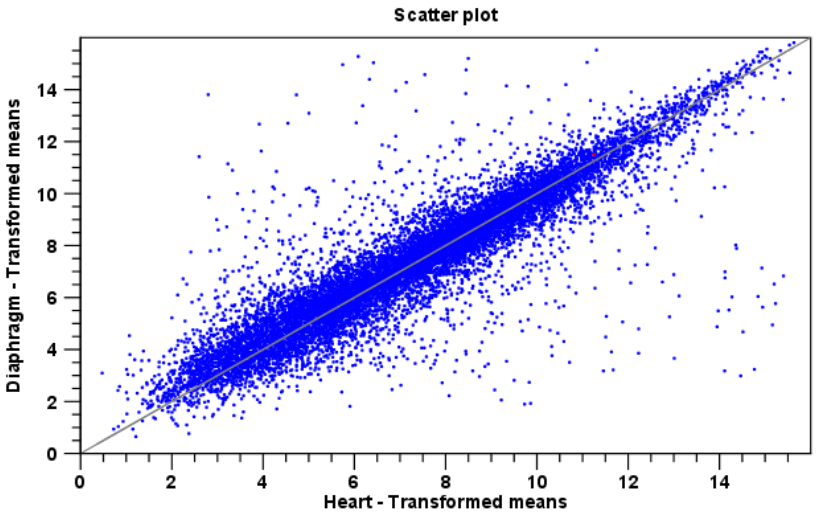



Figure 28.62: *A scatter plot of group means for two groups (transformed expression values).*

**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.
  - Outside
  - Inside
- **Tick lines at.** Choosing Major ticks will show a grid behind the graph.
  - None
  - Major ticks
- **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

  - Wide

- **Line type**
  - None
  - Line
  - Long dash
  - Short dash

- **Line color.** Allows you to choose between many different colors. 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**
  - None
  - Cross
  - Plus
  - Square
  - Diamond
  - Circle
  - Triangle
  - Reverse triangle
  - Dot

- **Dot color.** Allows you to choose between many different colors. 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).

### 28.4.6 Cross-view selections

There are a number of different ways of looking at an experiment as shown in figure 28.63).

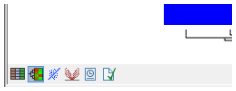


Figure 28.63: *An experiment can be viewed in several ways.*

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 28.63, 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 28.64.

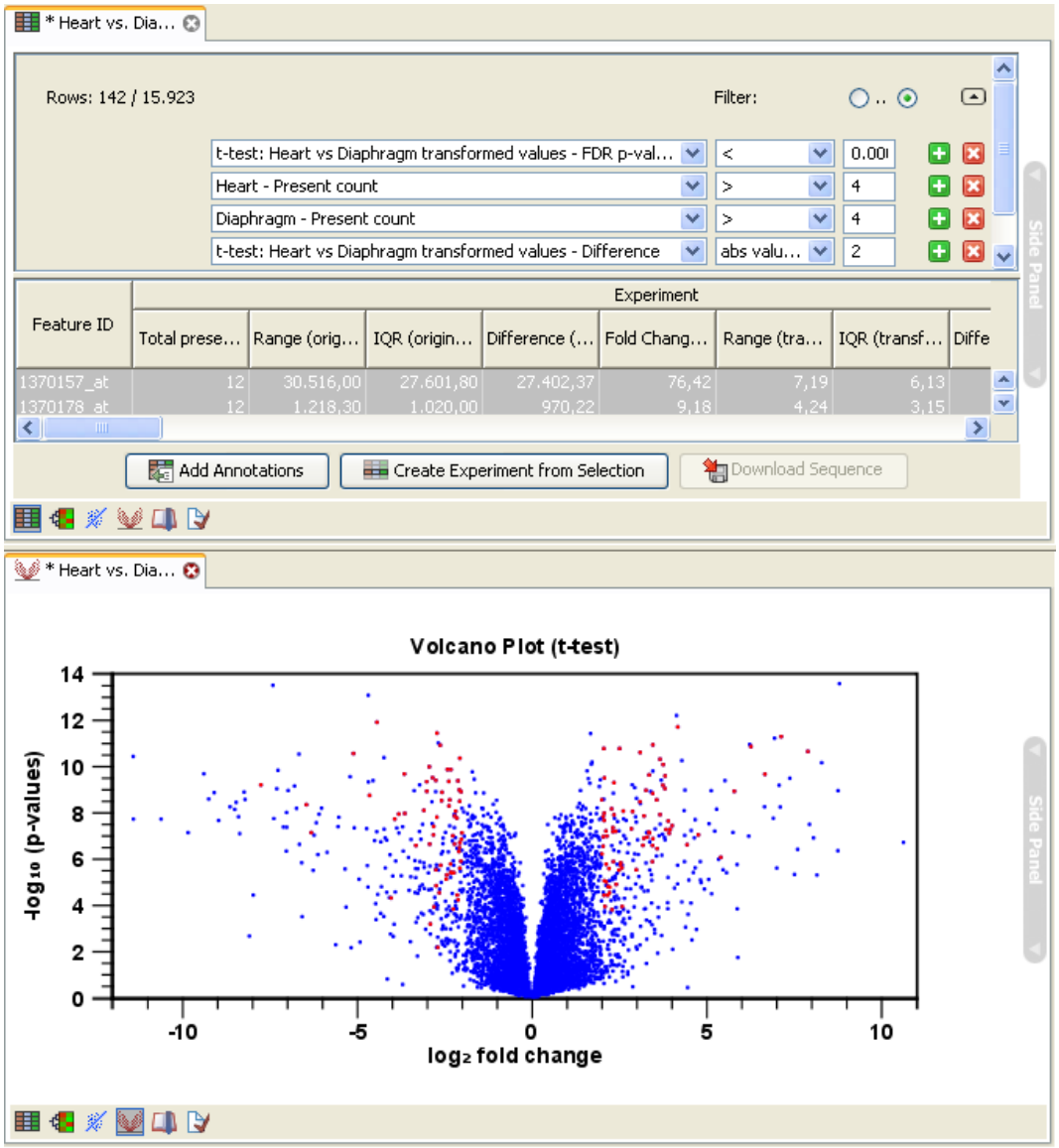



Figure 28.64: *A split view showing an experiment table at the top and a volcano plot at the bottom (note that you need to perform statistical analysis to show a volcano plot, see section 28.8).*

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 28.4.2) you typically want to choose 'Difference'.

## 28.5 Working with tracks and experiments

The *CLC Genomics Workbench* provides several tools for the analysis, organization, and visualization of expression data. In this section, we describe how Tracks and Experiments complement each other, and how they can be used together for the analysis of transcriptomics data using the tools found in the **Transcriptomics** toolbox.

### 28.5.1 Data structures for transcriptomics

The two main data structures used for transcriptomics data analysis in the *CLC Genomics Workbench* are tracks and experiments.

Tracks (see section 24) are the fundamental building blocks for data analysis in the *CLC Genomics Workbench*, where all information is tied to genomic positions. A central coordinate-system is provided by a reference genome, which allows that different types of data or results for different samples can be seen and analyzed together.

Experiments (see section 28.4), on the other hand, are used to represent complex relationships between expression samples, and to carry out statistical analysis (see section 28.8) of differential expression.

Tracks and experiments are intimately related, and it is possible in most cases to convert from one type to the other.

#### From Tracks to Experiments

When carrying out RNA-seq analysis using the **RNA-Seq Analysis** tool, the starting point is a set of reads from a sequencing study. As part of the RNA-Seq Analysis tool, these reads are mapped onto a reference genome. The RNA-Seq tool produces expression tracks, which are compatible with the reference genome, and can be visualized together with the genome in the **Track List View** (see section 24).

You can find more information about the RNA-Seq Analysis tool in section 28.1.

Once expression tracks have been obtained from the RNA-Seq Analysis tool, they can be used as sequencing-based sets of expression values in setting up an experiment. This can be done using the **Set up Experiment** tool, also found in the **Transcriptomics** toolbox. This is described in more detail in section 28.4.

1. Use the **Set up Experiment** tool on two or more expression tracks to set up an experiment, as described in section 28.4.

2. Save and open the resulting experiment, by double-clicking its name in the **Navigation Area**.
3. Use the **Create Track List** tool to create a track list from the expression tracks you used to set up the experiment, as described in section 24.
4. Save and open the resulting track list by double-clicking its name in the **Navigation Area**.
5. Drag the experiment tab downwards, until you see the blue shadow indicating the resulting placement (figure 28.65), and drop it in place. You should now have a divided view, with the experiment in the bottom half (figure 28.66).
6. Clicking on any line in the experiment will now automatically jump to the corresponding genomic location in the upper view. Use the **Zoom to Selection** ( ) button to zoom in to the desired genomic region.

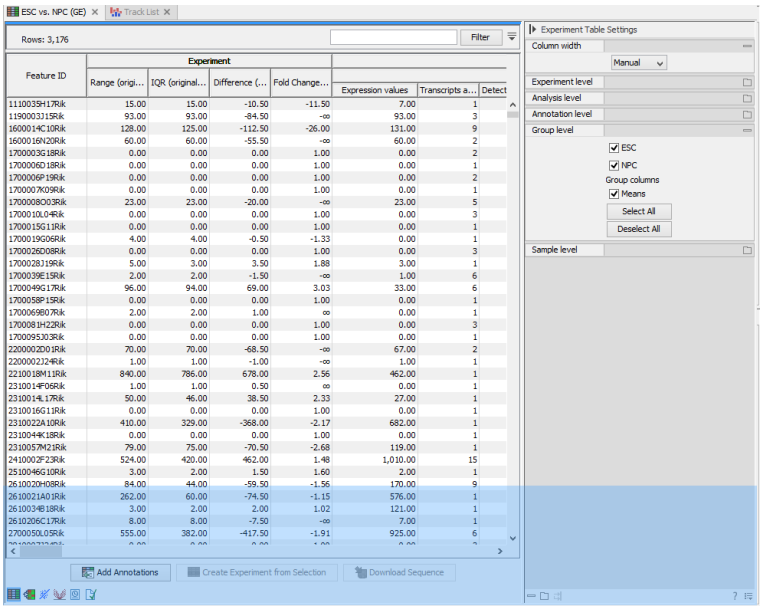



Figure 28.65: *Dragging a tab to the lower half of the view area.*

### 28.5.2 From Experiments to Tracks

Experiments can be used to carry out statistical analysis on the expression values obtained from RNA-seq analysis as described in section 28.8. The results of the statistical analysis are annotated on the experiment as additional columns.

It can be advantageous to visualize the results of the statistical analysis as tracks. The **Create Track from Experiment** tool in the *CLC Genomics Workbench* enables the conversion of experiments to tracks. You can find the **Create Track from Experiment** tool here:

**Toolbox** | **Transcriptomics Analysis** ( ) | **Create Track from Experiment** ( ) .

### 28.5.3 Running the Create Track from Experiment tool

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 two types of experiments:

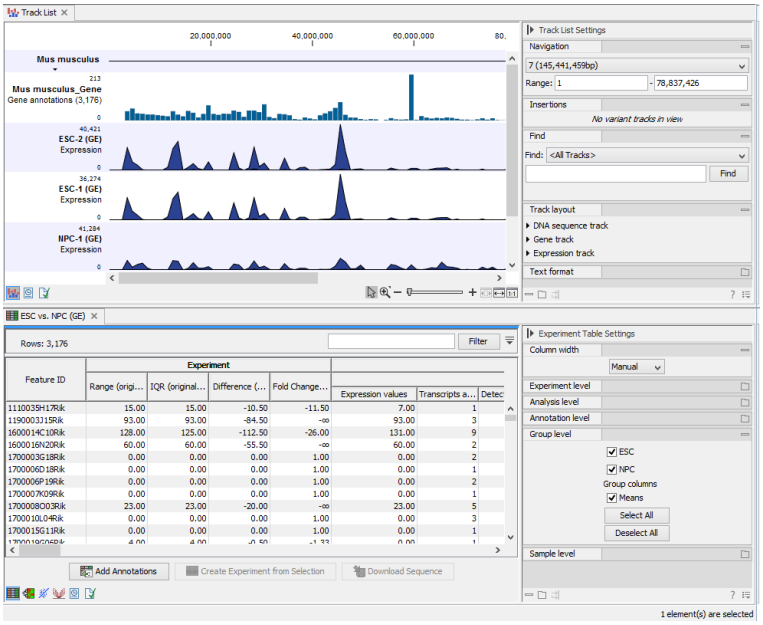


Figure 28.66: *After dropping a tab to the lower half othe view area.*

1. Experiments with associated genomic information, such as those created using expression tracks from the **RNA-Seq Analysis** tool.
2. Experiments without associated genomic information, such as those created using samples from the legacy 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 28.68.

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 28.67).

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.

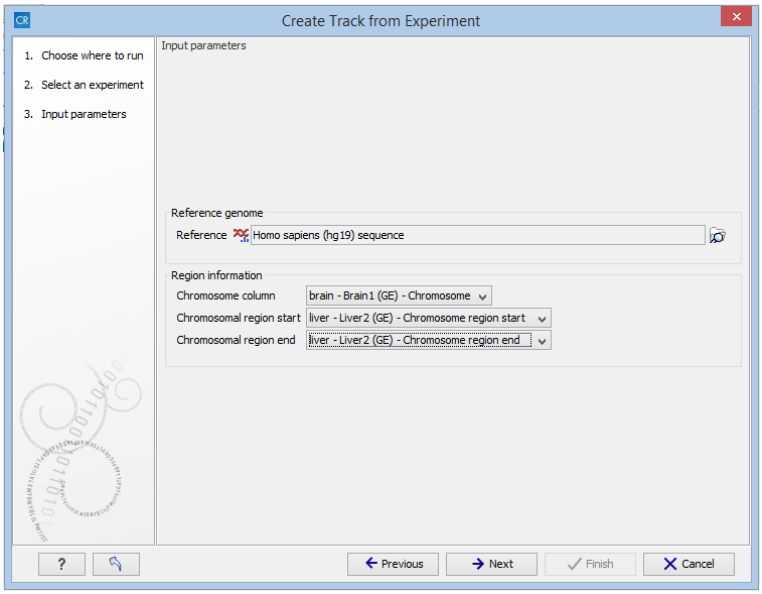



Figure 28.67: *The "Input parameters" step in the Create Track from Experiment tool.*

In the filtering step (figure 28.68), 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.
- **Type of p-value** This drop-down menu allows you to select between raw and corrected p-values (see section 28.8.4). 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 28.8.2): Proportions fold change.
- Baggerley's test(see section 28.8.2): Weighted proportions fold change.

- T-test (see section 28.8.3): Fold change.
- ANOVA (see section 28.8.3): Max fold change.
- Empirical analysis of DGE (see section 28.8.1): 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.

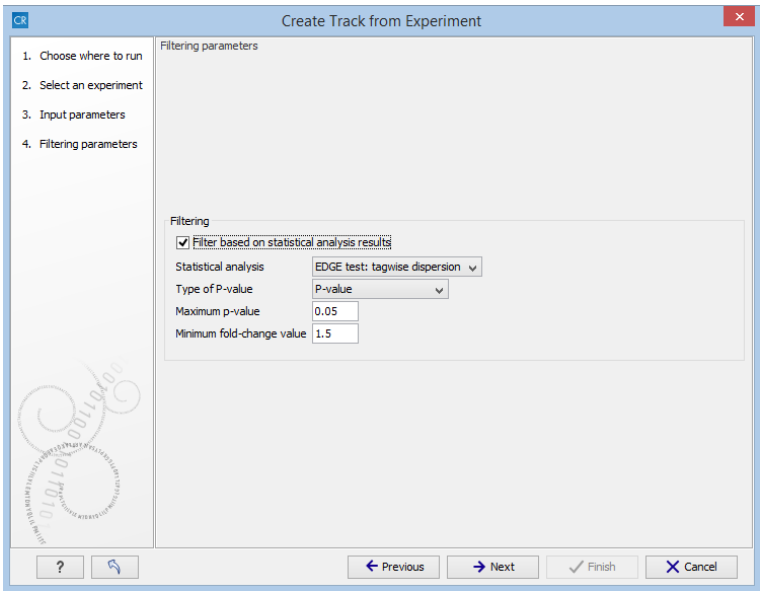



Figure 28.68: *The filtering step in the Create Track from Experiment tool.*

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 28.70).

- 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

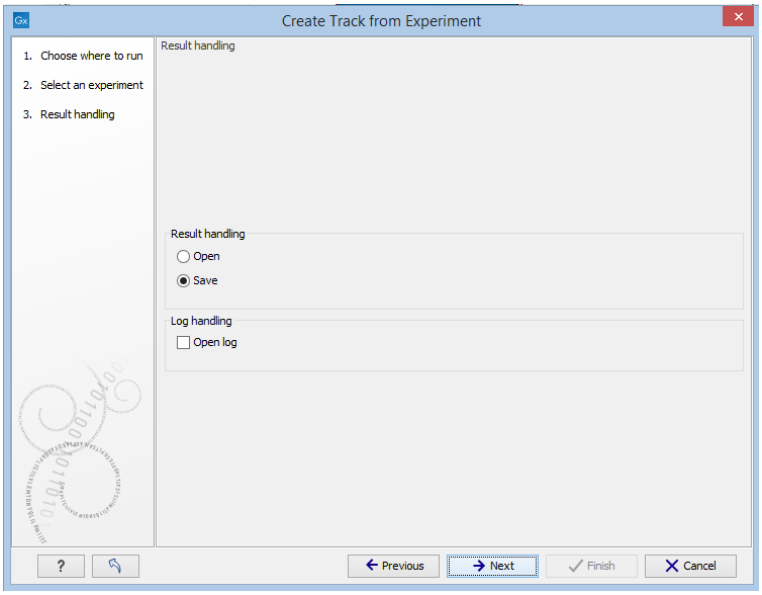



Figure 28.69: *The result handling step in the Create Track from Experiment tool.*

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'.

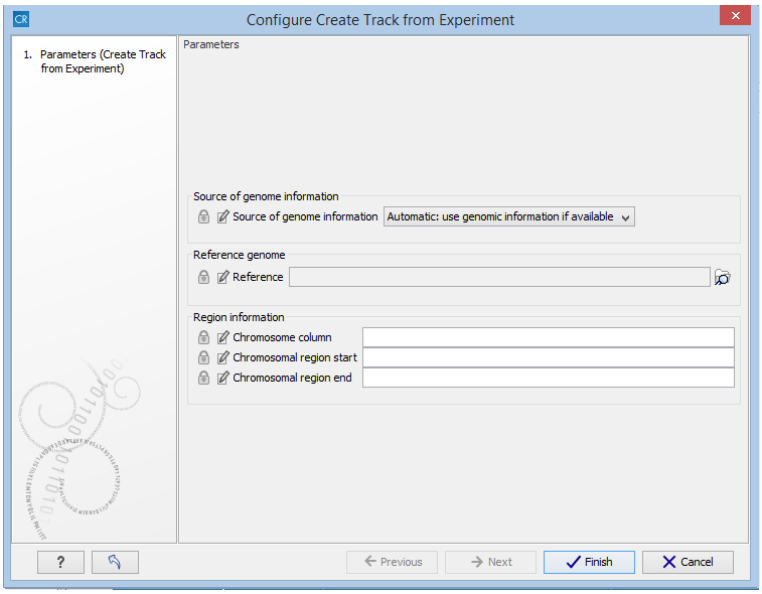



Figure 28.70: *Setting the parameters for the Create Track from Experiment tool in a workflow*

### 28.5.4 Interpreting the results of the Create Track from Experiment tool

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 the selected region. With the **Zoom to Selection** ( ) button it is possible to jump to and zoom in on the selected region (figure 28.71).

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 28.72).

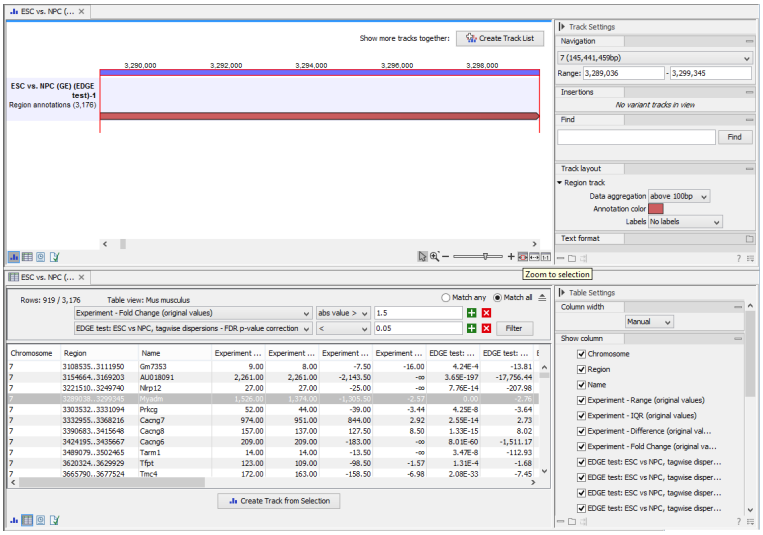



Figure 28.71: *Viewing the track produced by the Create Track from Experiment Tool*

## 28.6 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

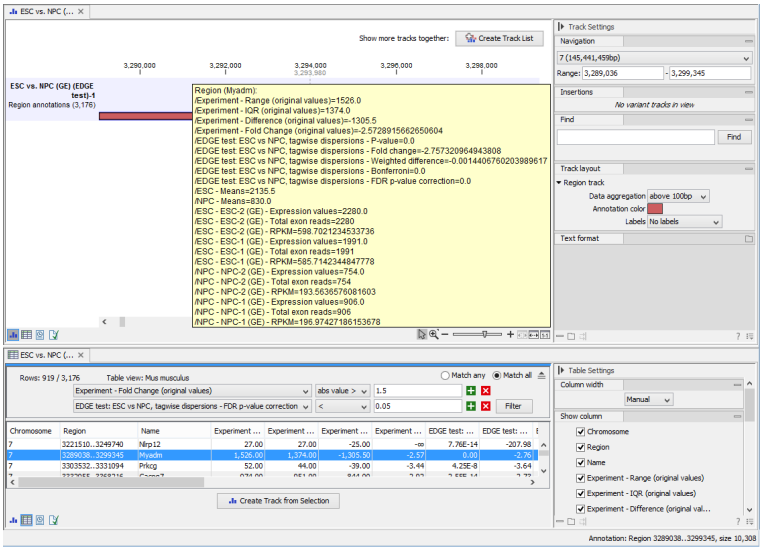


Figure 28.72: *The annotations on the track produced by the Create Track from Experiment Tool*

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).

### 28.6.1 Selecting transformed and normalized values for analysis

A number of the tools in the **Expression Analysis** ( ) folder use expression levels. All of these tools let you choose between *Original*, *Transformed* and *Normalized* expression values as shown in figure 28.73.

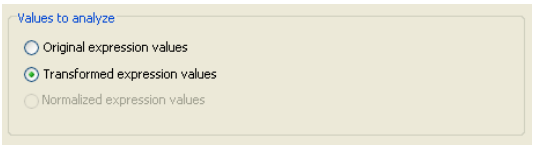



Figure 28.73: *Selecting which version of the expression values to analyze. In this case, the values have not been normalized, so it is not possible to select normalized values.*

In this case, the values have not been normalized, so it is not possible to select normalized values.

### 28.6.2 Transformation

The *CLC Genomics Workbench* lets you transform expression values based on logarithm and adding a constant:

> **Toolbox** | **Transcriptomics 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 28.74.

At the top, you can select which values to transform (see section 28.6.1).

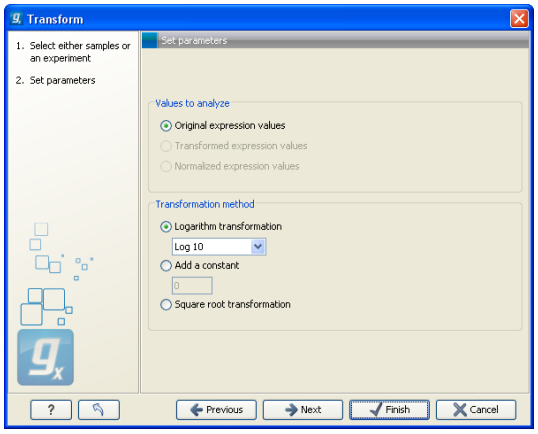



Figure 28.74: *Transforming expression values.*

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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

### 28.6.3 Normalization

The *CLC Genomics Workbench* lets you normalize expression values.

To start the normalization:

> **Toolbox** | **Transcriptomics 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 28.75.

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.

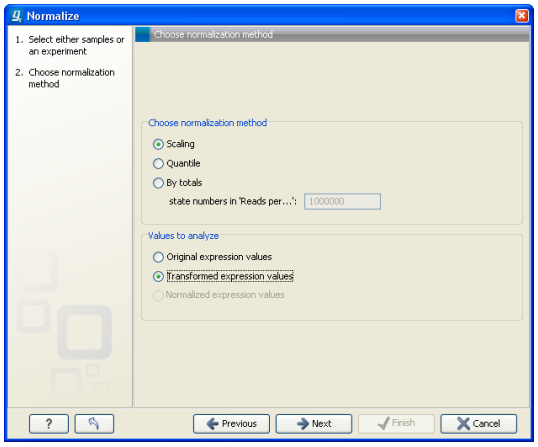



Figure 28.75: *Choosing normalization method.*

- **By totals**. This option is intended to be used with count-based data, i.e. data from RNA-seq, small RNA or expression profiling by tags. A sum is calculated for the expression values in a sample. The transformed value are generated by dividing the input values by the sample sum and multiplying by the factor (e.g. per '1,000,000').

Figures 28.76 and 28.77 show the effect on the distribution of expression values when using scaling or quantile normalization, respectively.

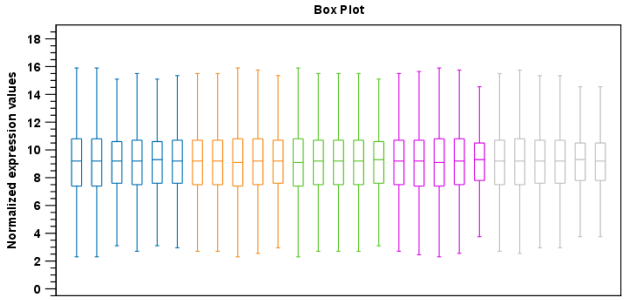



Figure 28.76: *Box plot after scaling normalization.*

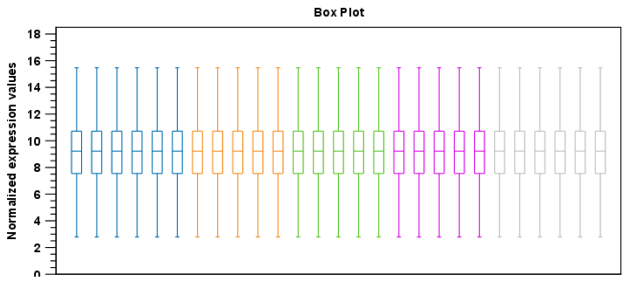



Figure 28.77: *Box plot after quantile normalization.*

At the bottom of the dialog in figure 28.75, you can select which values to normalize (see section 28.6.1).

Clicking **Next** will display a dialog as shown in figure 28.78.

The following parameters can be set:

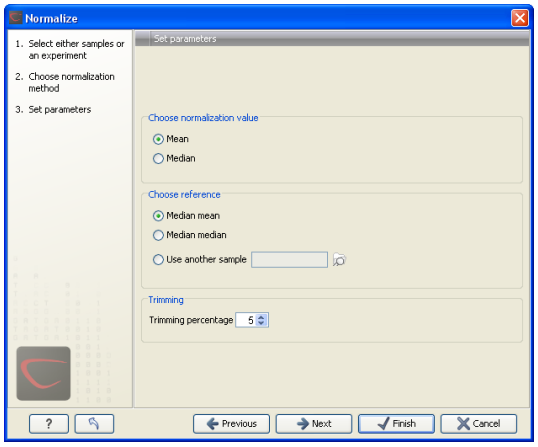



Figure 28.78: *Normalization settings.*

- **Normalization value**. The type of value of the samples which you want to ensure are equal for the normalized expression values
  - **Mean**.
  - **Median**.
- **Reference**. The specific value that you want the normalized value to be after normalization.
  - **Median mean**.
  - **Median median**.
  - **Use another sample**.
- **Trimming percentage**. Expression values that lie below the value of this percentile, or above 100 minus the value of this percentile, in the empirical distribution of the expression values in a sample will be excluded when calculating the normalization and reference values.

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

## 28.7 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.

### 28.7.1 Creating box plots - analyzing distributions

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** | **Transcriptomics 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 28.79.

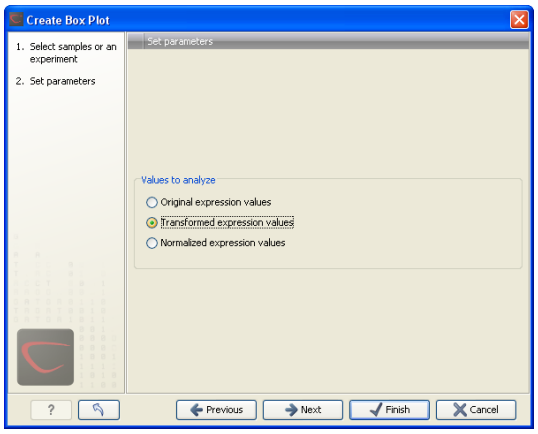



Figure 28.79: *Choosing values to analyze for the box plot.*

Here you select which values to use in the box plot (see section 28.6.1).

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

### Viewing box plots

An example of a box plot of a two-group experiment with 12 samples is shown in figure 28.80.

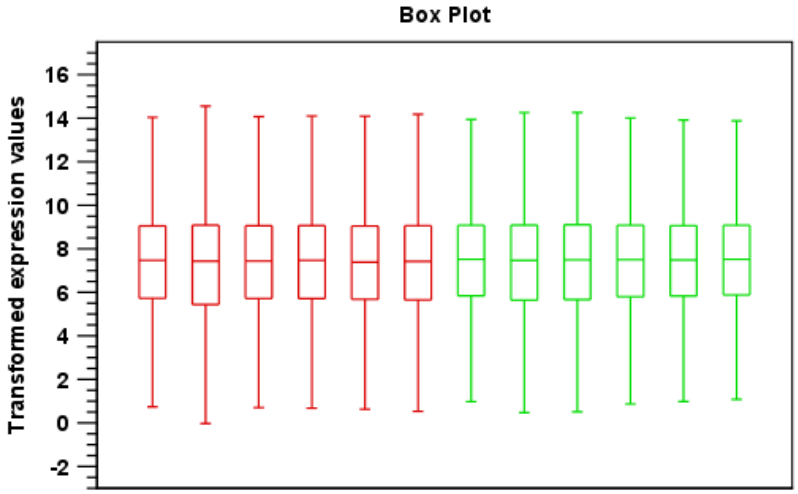



Figure 28.80: *A box plot of 12 samples in a two-group experiment, colored by group.*

Note that the boxes per default 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 (note that sample names are not shown in figure 28.80).

Per default the box includes the IQR values (from the lower to the upper quartile), the median is displayed as a line in the box, and the whiskers extend 1.5 times the height of the box.

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 28.81).

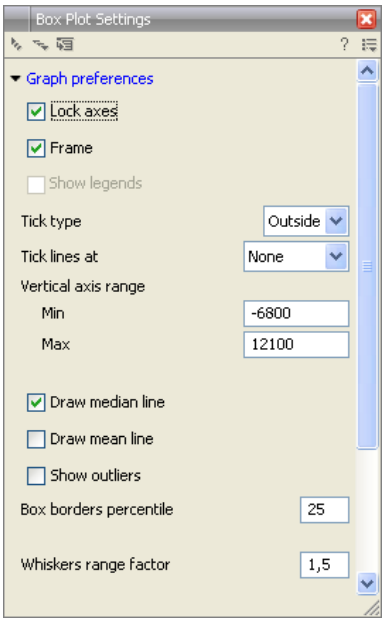



Figure 28.81: *Graph preferences for a box plot.*

- **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.
  - Outside
  - Inside
- **Tick lines at.** Choosing Major ticks will show a grid behind the graph.
  - None
  - Major ticks
- **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 28.82).

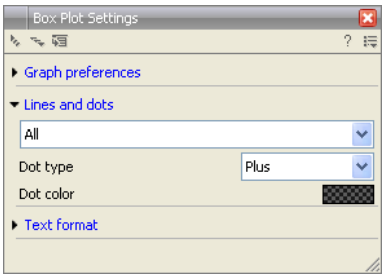



Figure 28.82: *Lines and dot preferences for a box plot.*

- **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**
  - None
  - Cross
  - Plus
  - Square
  - Diamond
  - Circle
  - Triangle
  - Reverse triangle
  - Dot
- **Dot 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 box plot, you need to save the settings of the **Side Panel** (see section 4.6).

#### Interpreting the box plot

This section will show how to interpret a box plot through a few examples.

First, if you look at figure 28.83, you can see a box plot for an experiment with 5 groups and 27 samples.

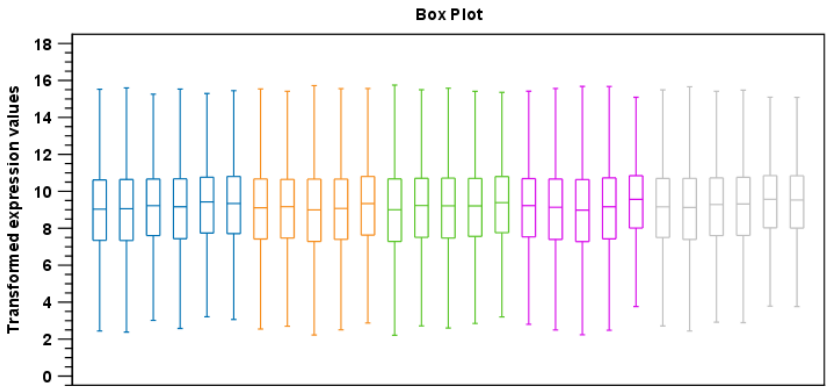



Figure 28.83: *Box plot for an experiment with 5 groups and 27 samples.*

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 28.84 shows a box plot for the same experiment after quantile normalization: the distributions have been brought into par.

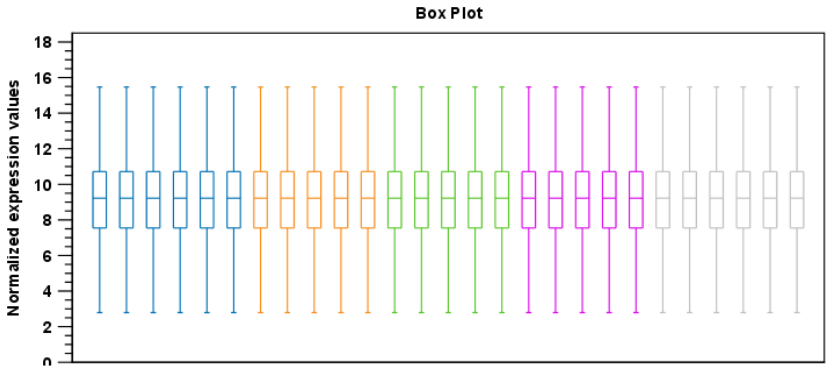



Figure 28.84: *Box plot after quantile normalization.*

In figure 28.85 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.

### 28.7.2 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 sampe be a cluster

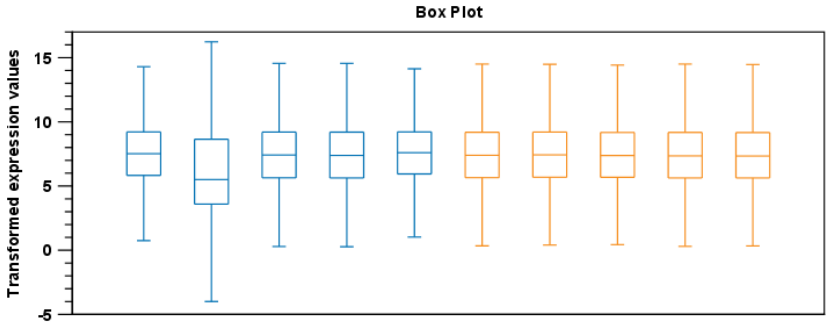



Figure 28.85: *Box plot for a two-group experiment with 5 samples.*

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** | **Transcriptomics 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 28.86. The hierarchical clustering algorithm requires that you specify a distance measure and a cluster linkage. The similarity measure is used to specify how distances between two samples should be calculated. The cluster distance metric specifies how you want the distance between two clusters, each consisting of a number of samples, to be calculated.

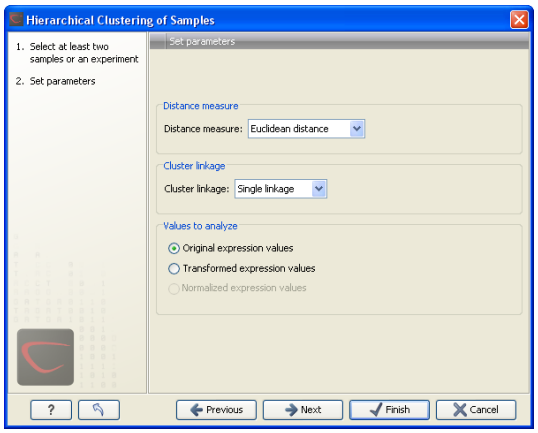



Figure 28.86: *Parameters for hierarchical clustering of samples.*

At the top, you can choose 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

$$|u - v| = \sum_{i=1}^{n}|u_i - v_i|.$$

Next, you can select the cluster linkage to be used:

- **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 28.6.1).

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

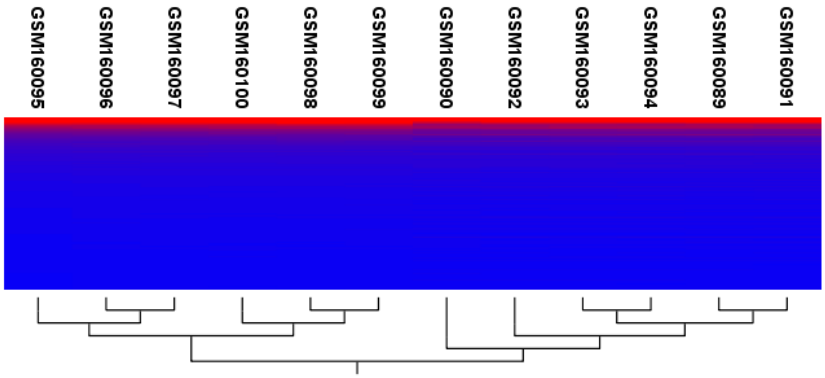



Figure 28.87: *Sample clustering.*

**Result of hierarchical clustering of samples**

The result of a sample clustering is shown in figure 28.87.

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 28.88).

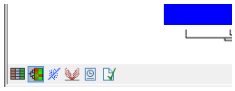


Figure 28.88: *Showing the hierarchical clustering of an experiment.*

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, the view of the clustering is the same. As you can see in figure 28.87, 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 28.9.1).

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 28.89).

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 28.103).

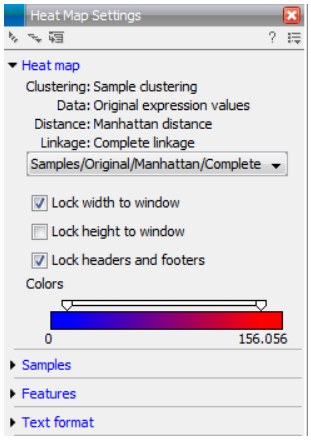



Figure 28.89: *Side Panel of heat map.*

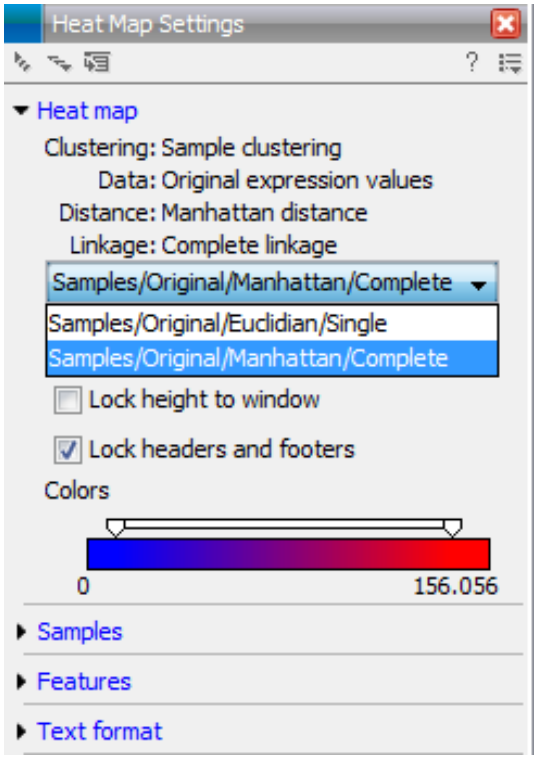



Figure 28.90: *When more than one clustering has been performed, there will be a list of heat maps to choose from.*

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.
- **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).

### 28.7.3 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** | **Transcriptomics 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 28.91.

In this dialog, you select the values to be used for the principal component analysis (see section 28.6.1).

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

#### Principal component analysis plot

This will create a principal component plot as shown in figure 28.92.

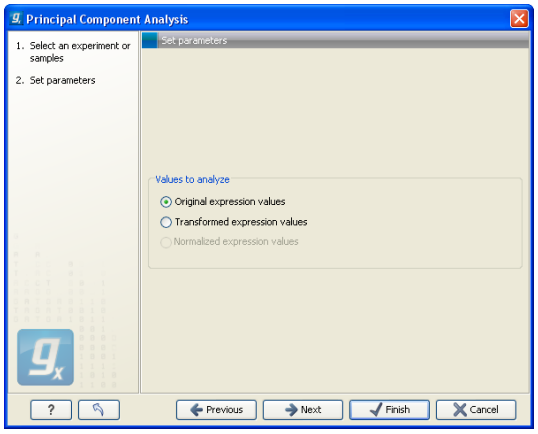



Figure 28.91: *Selcting which values the principal component analysis should be based on.*

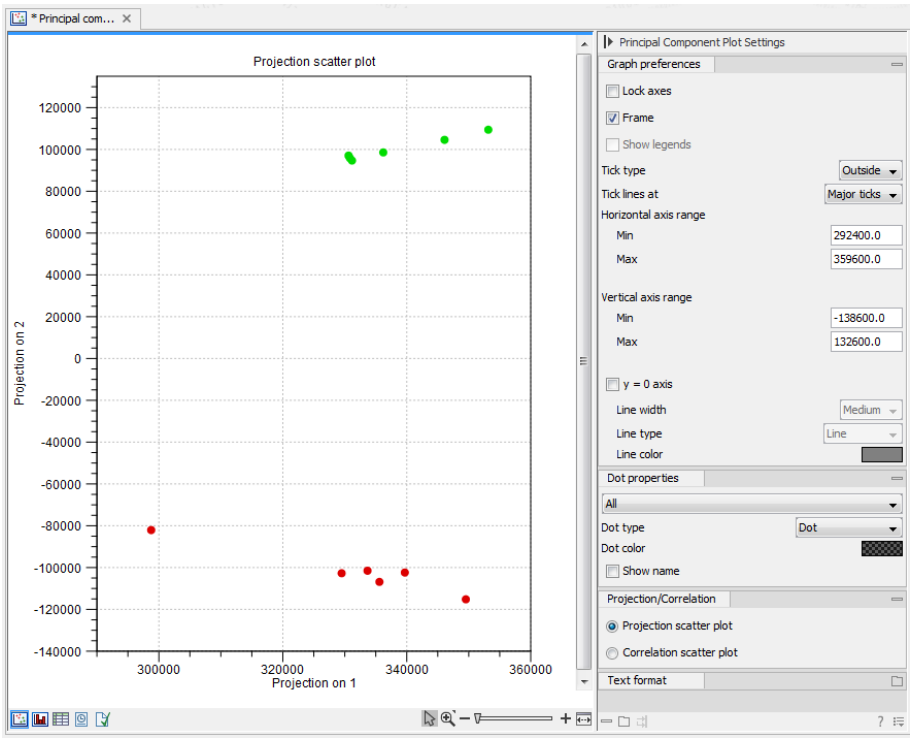



Figure 28.92: *A principal component analysis colored by group.*

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 28.92 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.
  - Outside
  - Inside
- **Tick lines at.** Choosing Major ticks will show a grid behind the graph.
  - None
  - Major ticks
- **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
    - Wide
  - **Line type**
    - None
    - Line
    - Long dash
    - Short dash
  - **Line color.** Allows you to choose between many different colors. Click the color box to select a color.

Below the general preferences, you find the **Dot properties**:

- **Drop down menu** In this you choose which of the samples (that is, which 'dots') the choices you make below should apply to. You can choose between 'All', a particular group in your experiment, or a particular samples in your experiment.
- **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**
  - None
  - Cross
  - Plus
  - Square
  - Diamond
  - Circle
  - Triangle
  - Reverse triangle
  - Dot
- **Dot color.** Allows you to choose between many different colors. 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 28.92, 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 the each of the principal components. The first principal component explains about 99 percent 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.
  - Outside
  - Inside
- **Tick lines at.** Choosing Major ticks will show a grid behind the graph.
  - None
  - Major ticks
- **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**
  - None
  - Cross
  - Plus
  - Square
  - Diamond
  - Circle
  - Triangle
  - Reverse triangle
  - Dot
- **Dot color.** Allows you to choose between many different colors. Click the color box to select a color.
- **Line width**
  - Thin
  - Medium
  - Wide
- **Line type**
  - None
  - Line
  - Long dash
  - Short dash

- **Line color.** Allows you to choose between many different colors. 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).

## 28.8 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 28.8.1 and section 28.8.2) and tests that assume that the data is real-valued, has Gaussian distributions and compare means (described in section 28.8.3).

To run the statistical analysis:

**Toolbox** | **Transcriptomics Analysis ( )**| **Statistical Analysis** | **Empirical Analysis of DGE ( )**

**Toolbox** | **Transcriptomics Analysis ( )**| **Statistical Analysis** | **On Proportions ( )**

or **Toolbox** | **Transcriptomics Analysis ( )**| **Statistical Analysis** | **On Gaussian Data ( )**

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 28.4.1).

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.

### 28.8.1 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. This is typically the case for RNA-seq expression analysis. The test 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 '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 28.8.1).

#### 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 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. Overall, the user has a somewhat compromised run time but gains greater confidence in the results at the end.

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** | **Transcriptomics 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 28.93). 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.

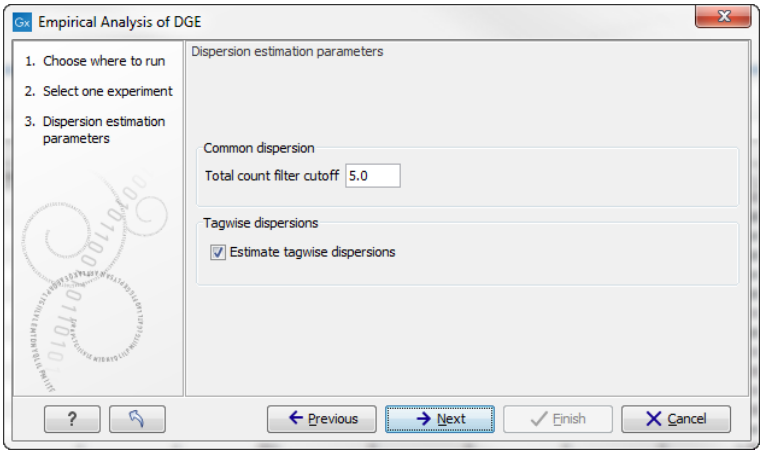



Figure 28.93: *Empirical analysis of DGE: setting the parameters related to dispersion.*

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 28.94). 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). The user can specify if Bonferroni and FDR corrected p-values should be calculated (see Section 28.8.4).

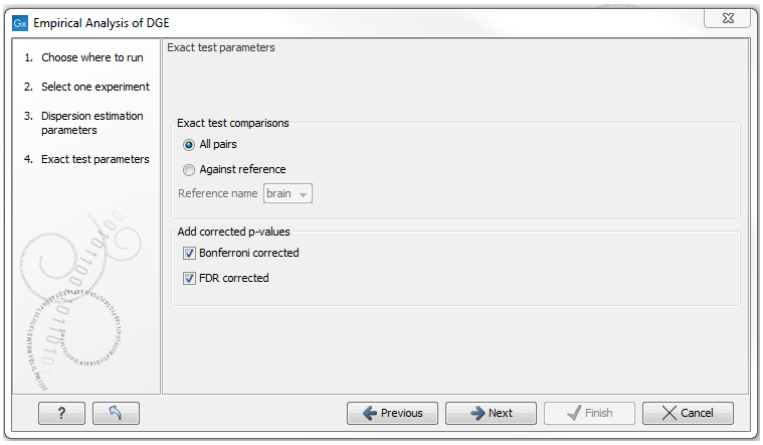



Figure 28.94: *Empirical analysis of DGE: setting comparisons and corrected p-value options.*

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.

### 28.8.2 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 RNA-Seq or tag profiling. 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 28.4.1), 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 (see 28.8.4).

#### 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 (see 28.8.4).

### 28.8.3 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 28.95.

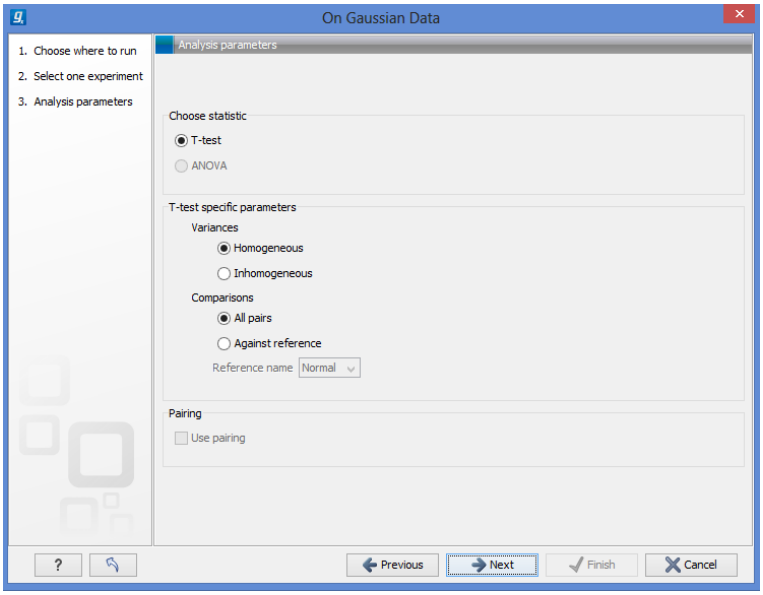



Figure 28.95: *Selecting a t-test.*

There are different types of t-tests, depending on the assumption you make about the variances in the groups. By selecting 'Homogeneous' (the default) calculations are done assuming that the groups have equal variances. When 'In-homogeneous' is selected, this assumption is not made.

The t-test can also be chosen if you have a multi-group experiment. In this case you may choose either to have t-tests produced for all pairs of groups (by clicking the 'All pairs' button) or to have a t-test produced for each group compared to a specified reference group (by clicking the 'Against reference' button). In the last case you must specify which of the groups you want to use as reference (the default is to use the group you specified as Group 1 when you set up the experiment).

If a experiment with pairing was set up (see section 28.4.1) 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 (see 28.8.4).

### ANOVA

For experiments with more than two groups you can choose **T-test** as described above, or **ANOVA** as shown in figure 28.96.

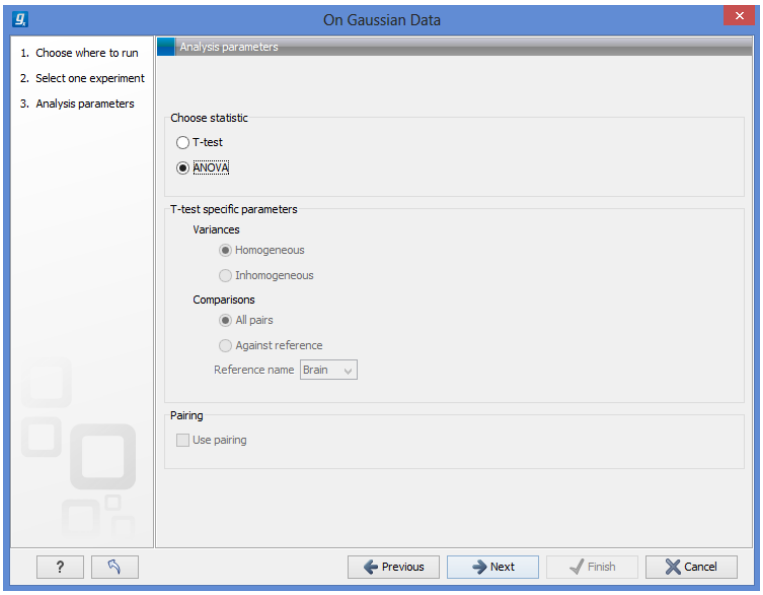



Figure 28.96: *Selecting ANOVA.*

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 28.4.1) 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 (see 28.8.4).

### 28.8.4 Corrected p-values

Clicking **Next** will display a dialog as shown in figure 28.97.

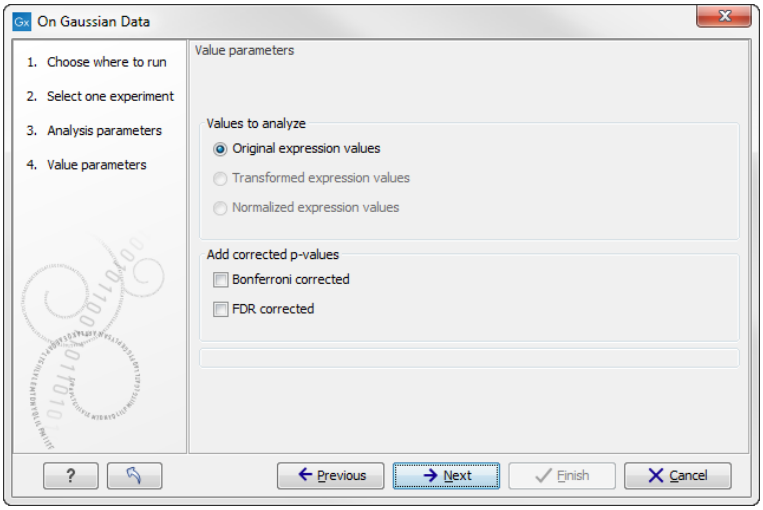



Figure 28.97: *Additional settings for the statistical analysis.*

At the top, you can select which values to analyze (see section 28.6.1).

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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

Note that if you have already performed statistical analysis on the same values, the existing one will be overwritten.

### 28.8.5 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 28.4.2). 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 8.3).

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 28.4.5, 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 28.98.

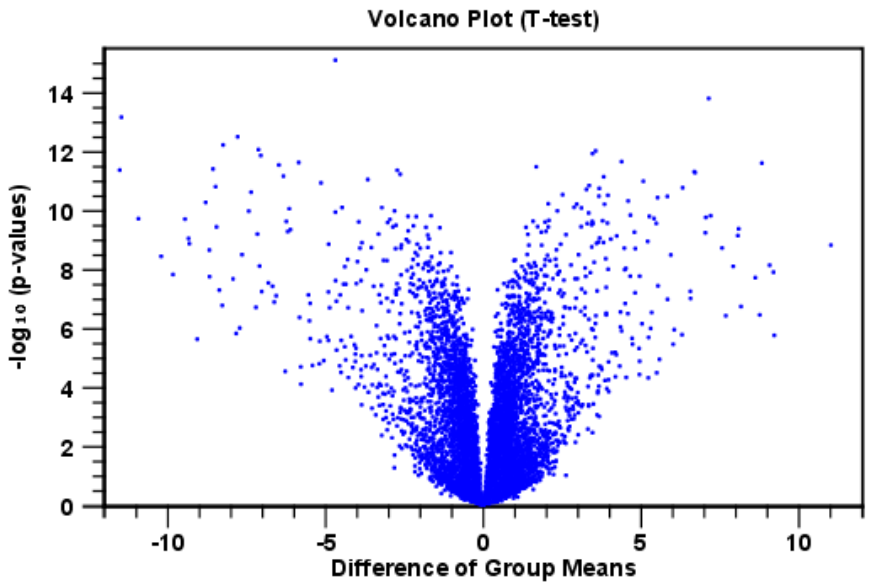



Figure 28.98: *Volcano plot.*

The volcano plot shows the relationship between the p-values of a statistical test and the magnitude of the difference in expression values of the samples in the groups. On the y-axis the $-\log_{10}$ p-values are plotted. For the x-axis you may choose between two sets of values by choosing either 'Fold change' or 'Difference' in the volcano plot side panel's 'Values' part. If you choose 'Fold change' the log of the values in the 'fold change' (or 'Weighted fold change') column for the test will be displayed. If you choose 'Difference' the values in the 'Difference' (or 'Weighted difference') column will be used. Which values you wish to display will depend upon the scale of you data (Read the note on fold change in section 28.4.2).

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.
  - Outside
  - Inside
- **Tick lines at.** Choosing Major ticks will show a grid behind the graph.
  - None
  - Major ticks
- **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**
  - None
  - Cross
  - Plus
  - Square

  - Diamond
  - Circle
  - Triangle
  - Reverse triangle
  - Dot

- **Dot color.** Allows you to choose between many different colors. 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 28.4.2.

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).

## 28.9 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.

### 28.9.1 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 | Transcriptomics 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 28.4.2.

Clicking **Next** will display a dialog as shown in figure 28.99. The hierarchical clustering algorithm requires that you specify a distance measure and a cluster linkage. The distance measure is used specify how distances between two features should be calculated. The cluster linkage specifies how you want the distance between two clusters, each consisting of a number of features, to be calculated.

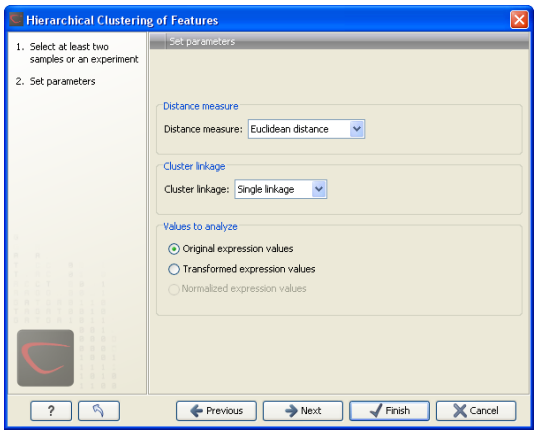



Figure 28.99: *Parameters for hierarchical clustering of features.*

At the top, you can choose 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

$$|u - v| = \sum_{i=1}^{n} |u_i - v_i|.$$

Next, you can select different ways to calculate distances between clusters. The possible cluster linkage to use 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 28.6.1). Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

#### Result of hierarchical clustering of features

The result of a feature clustering is shown in figure 28.100.

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 28.101).

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 28.100). 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 28.102).

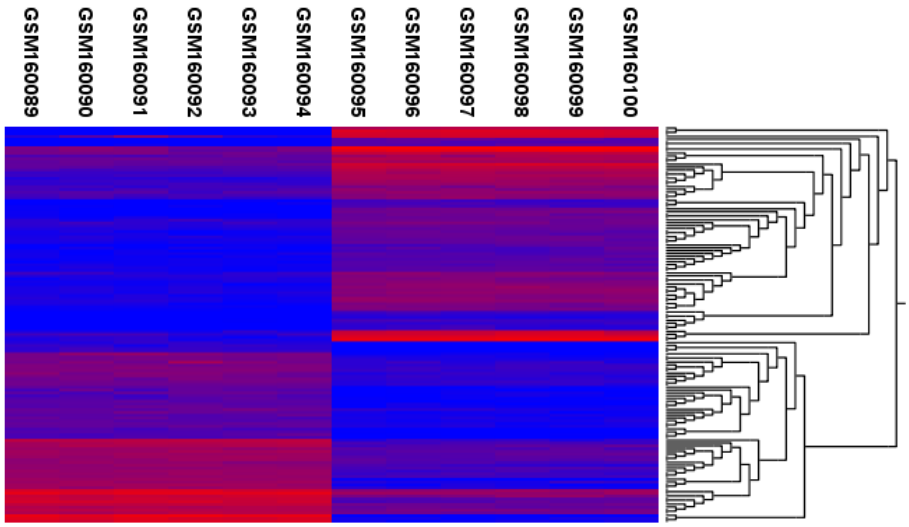



Figure 28.100: *Hierarchical clustering of features.*

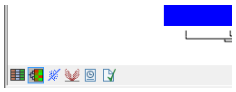


Figure 28.101: *Showing the hierarchical clustering of an experiment.*

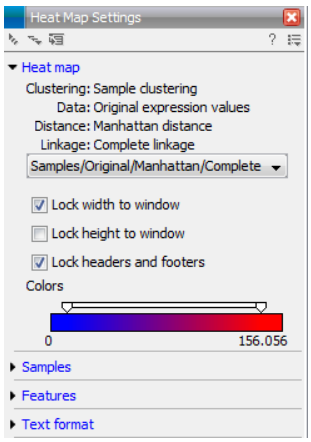



Figure 28.102: *Side Panel of heat map.*

At the top, there is information about the heat map currently displayed. The information regards type of clustering, expression value used together with distance and linkage information. If you have performed more than one clustering, you can choose between the resulting heat maps in a drop-down box (see figure 28.103).

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.

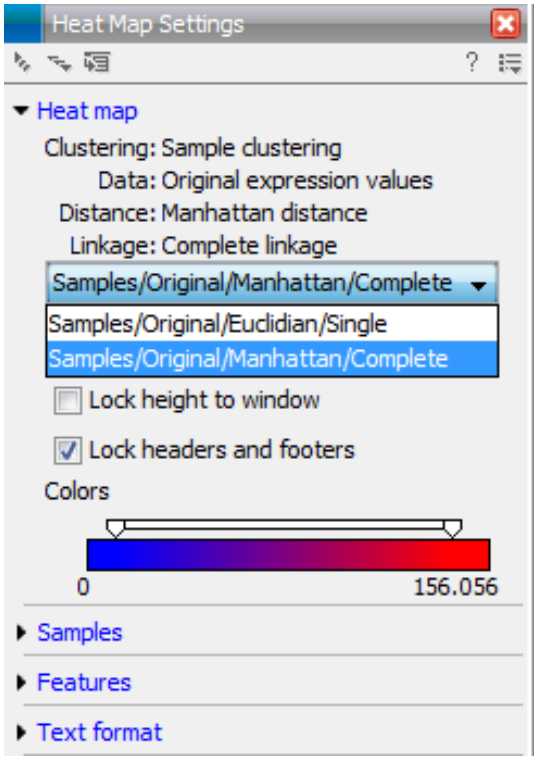



Figure 28.103: *When more than one clustering has been performed, there will be a list of heat maps to choose from.*

- **Lock height to window**. This is the corresponding option for the height. Note that if you check both options, you will not be able to zoom at all, since both the width and the height is fixed.
- **Lock headers and footers**. This will ensure that you are always able to see the sample and feature names and the trees when you zoom in.
- **Colors**. The expression levels are visualized using a gradient color scheme, where the right side color is used for high expression levels and the left side color is used for low expression levels. You can change the coloring by clicking the box, and you can change the relative coloring of the values by dragging the two knobs on the white slider above.

Below you find the **Samples** and **Features** groups. They contain options to show names, legend, and tree above or below the 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).

### 28.9.2 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 | Transcriptomics 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 28.4.2.

Clicking **Next** will display a dialog as shown in figure 28.104.

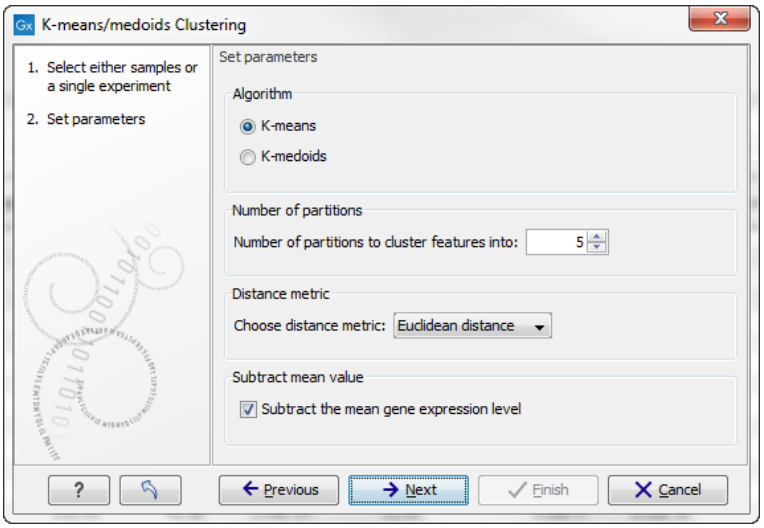



Figure 28.104: *Parameters for k-means/medoids clustering.*

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 $\mu_i$ 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 number of partitions to cluster features into.
- **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 28.105.

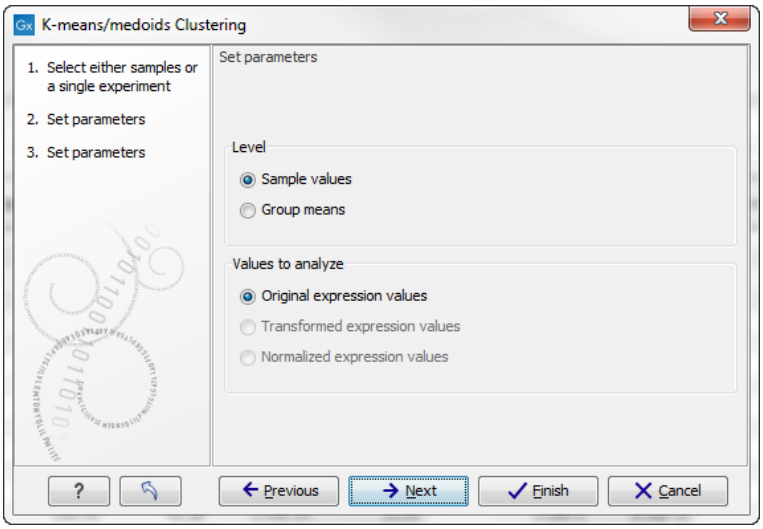



Figure 28.105: *Parameters for k-means/medoids clustering.*

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 28.6.1).

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

### 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 28.104) - there is one graph per cluster. Using drag and drop as explained in section 2.1.6, you can arrange the views to see more than one graph at the time.

Figure 28.106 shows an example where four clusters have been arranged side-by-side.

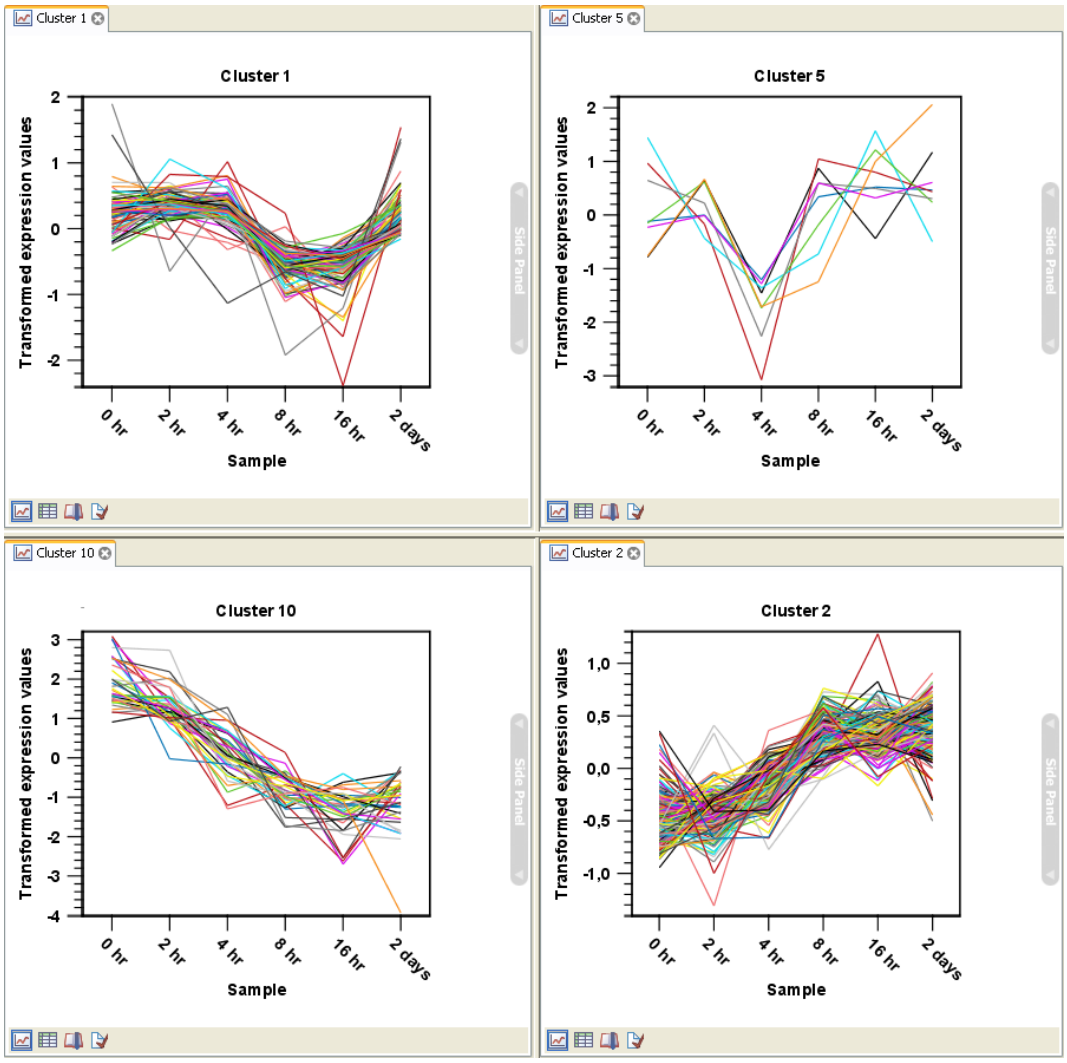



Figure 28.106: *Four clusters created by k-means/medoids clustering.*

The samples used are from a time-series experiment, and you can see that the expression levels for each cluster have a distinct pattern. The two clusters at the bottom have falling and rising expression levels, respectively, and the two clusters at the top both fall at the beginning but then

rise again (the one to the right starts to rise earlier 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 28.4.6.

## 28.10 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 28.4.4.

### 28.10.1 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** | **Transcriptomics 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 28.4.2).

Click **Next**. This will display the dialog shown in figure 28.107.

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:

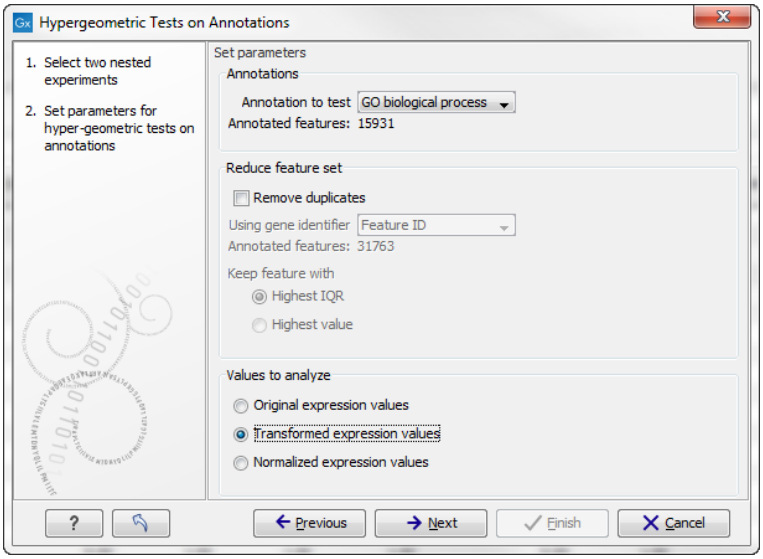



Figure 28.107: *Parameters for performing a hypergeometric test on annotations.*

- **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 28.6.1). 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:

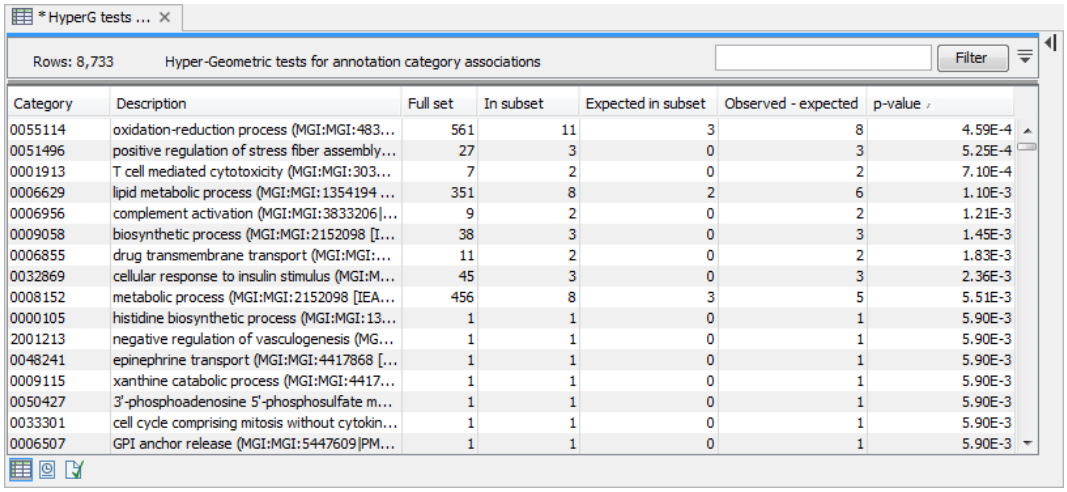
- If there are features with values of plus infinity, minus infinity or NaN, those features will not be taken forward into the test. This can be a consideration when choosing transformed values, where the mathematical manipulations involved may lead to such values.
- If you chose to remove duplicates, 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.

The final number of features used for the test is reported in this history view of the test results.

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

#### Result of hypergeometric tests on annotations

The result of performing hypergeometric tests on annotations using GO biological process is shown in figure 28.108.



*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 28.108: *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 categories that are over or under-represented on the features in the subset relative to the full set.

### 28.10.2 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** | **Transcriptomics Analysis (****)**| **Annotation Test** | **Gene Set Enrichment Analysis (GSEA) (**  **)**

Select an experiment and click **Next**.

Click **Next**. This will display the dialog shown in figure 28.109.

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

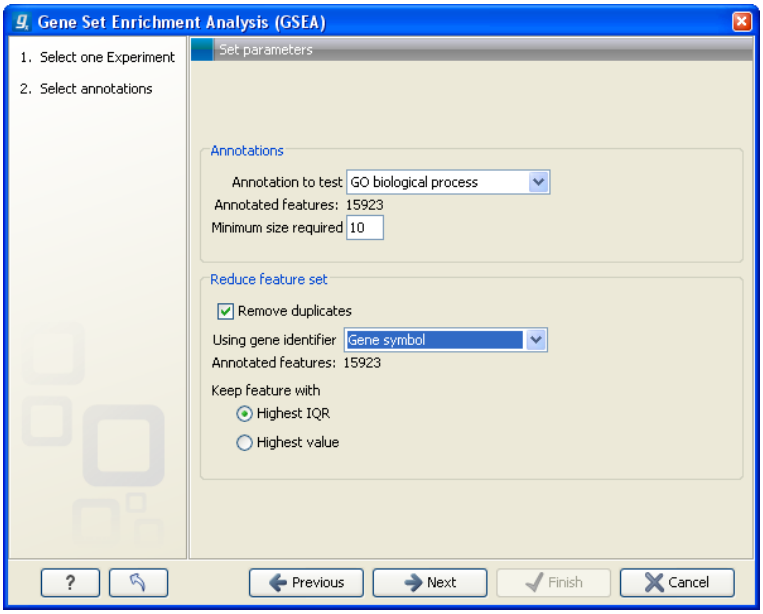



Figure 28.109: *Gene set enrichment analysis on GO biological process*

feature in an experiment. If this is not taken into account it may lead to a biased result. The standard way to deal with this is to reduce the set of features considered, so that each gene is represented only once. Check the **Remove duplicates** check box to reduce the feature set, and you can choose how you want this to be done:

- **Using gene identifier**.
- **Keep feature with**:
  - **Highest IQR**. The feature with the highest interquartile range (IQR) is kept.
  - **Highest value**. The feature with the highest expression value is kept.

First you specify which annotation you want to use as gene identifier. Once you have selected this, you will see the number of features carrying this annotation below. Next you specify which feature you want to keep for each gene. This may be either the feature with the highest inter-quartile range or the highest value.

Clicking **Next** will display the dialog shown in figure 28.110.

At the top, you can select which values to analyze (see section 28.6.1).

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 **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

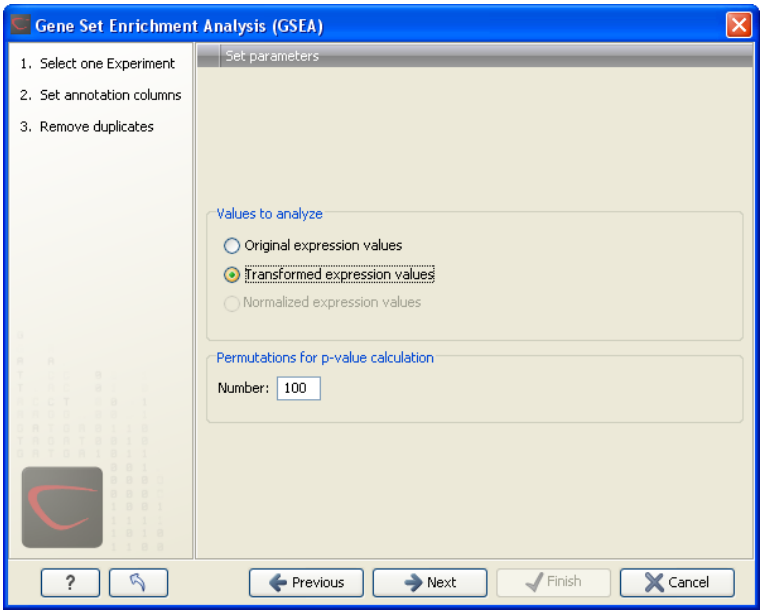
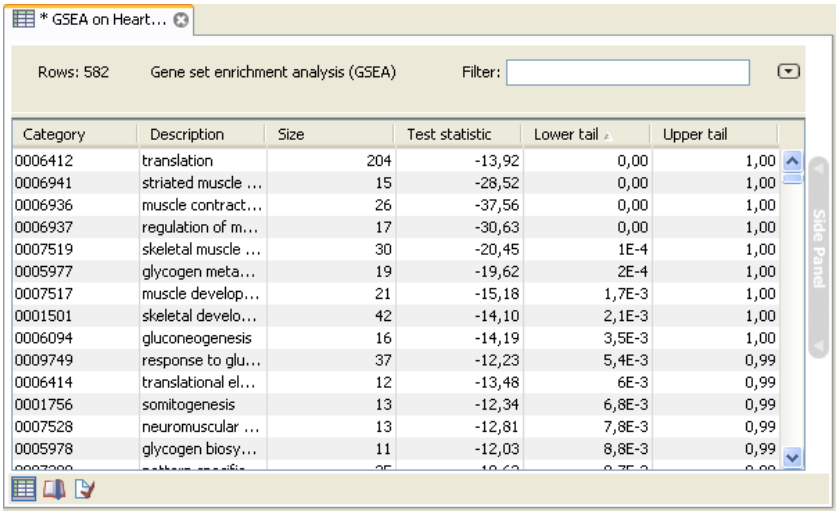



Figure 28.110: *Gene set enrichment analsysis parameters.*

### Result of gene set enrichment analysis

The result of performing gene set enrichment analysis using GO biological process is shown in figure 28.111.




| Category | Description | Size | Test statistic | Lower tail | Upper tail |
|---|---|---|---|---|---|
| 0006412 | translation | 204 | -13,92 | 0,00 | 1,00 |
| 0006941 | striated muscle ... | 15 | -28,52 | 0,00 | 1,00 |
| 0006936 | muscle contract... | 26 | -37,56 | 0,00 | 1,00 |
| 0006937 | regulation of m... | 17 | -30,63 | 0,00 | 1,00 |
| 0007519 | skeletal muscle ... | 30 | -20,45 | 1E-4 | 1,00 |
| 0005977 | glycogen meta... | 19 | -19,62 | 2E-4 | 1,00 |
| 0007517 | muscle develop... | 21 | -15,18 | 1,7E-3 | 1,00 |
| 0001501 | skeletal develo... | 42 | -14,10 | 2,1E-3 | 1,00 |
| 0006094 | gluconeogenesis | 16 | -14,19 | 3,5E-3 | 1,00 |
| 0009749 | response to glu... | 37 | -12,23 | 5,4E-3 | 0,99 |
| 0006414 | translational el... | 12 | -13,48 | 6E-3 | 0,99 |
| 0001756 | somitogenesis | 13 | -12,34 | 6,8E-3 | 0,99 |
| 0007528 | neuromuscular ... | 13 | -12,81 | 7,8E-3 | 0,99 |
| 0005978 | glycogen biosy... | 11 | -12,03 | 8,8E-3 | 0,99 |

Figure 28.111: *The result of gene set enrichment analysis on GO biological process.*

The table shows the following information:

- **Category**. This is the identifier for the category.
- **Description**. This is the description belonging to the category. Both of these are simply extracted from the annotations.
- **Size**. The number of features with this category. (Note that this is after removal of duplicates).
- **Test statistic**. This is the GSEA test statistic.

- **Lower tail**. This is the mass in the permutation based p-value distribution below the value of the test statistic.
- **Upper tail**. This is the mass in the permutation based p-value distribution above the value of the test statistic.

A small lower (or upper) tail p-value for an annotation category is an indication that features in this category viewed as a whole are perturbed among the groups in the experiment considered.

## 28.11 General plots

The last folder in the **Expression Analysis** ( ) folder in the **Toolbox** is **General Plots**. Here you find three general plots that may be useful at various point of your analysis work flow. The plots are explained in detail below.

### 28.11.1 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** | **Transcriptomics 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 28.112.

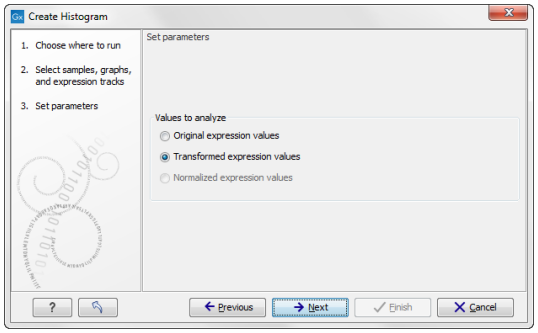



Figure 28.112: *Selcting which values the histogram should be based on.*

In this dialog, you select the values to be used for creating the histogram (see section 28.6.1).

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

#### Viewing histograms

The resulting histogram is shown in a figure 28.113

The histogram shows the expression value on the x axis (in the case of figure 28.113 the transformed expression values) and the counts of these values on the y axis.

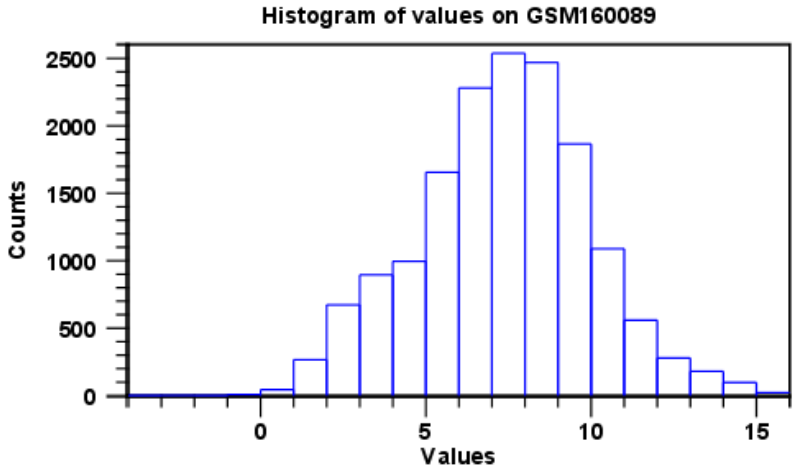



Figure 28.113: *Histogram showing the distribution of transformed expression values.*

In the **Side Panel** to the left, there is a number of options to adjust the view. Under **Graph preferences**, you can adjust the general properties of the plot.

- **Lock axes.** This will always show the axes even though the plot is zoomed to a detailed level.
- **Frame.** Shows a frame around the graph.
- **Show legends.** Shows the data legends.
- **Tick type**. Determine whether tick lines should be shown outside or inside the frame.
  - Outside
  - Inside
- **Tick lines at.** Choosing Major ticks will show a grid behind the graph.
  - None
  - Major ticks
- **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 28.114.

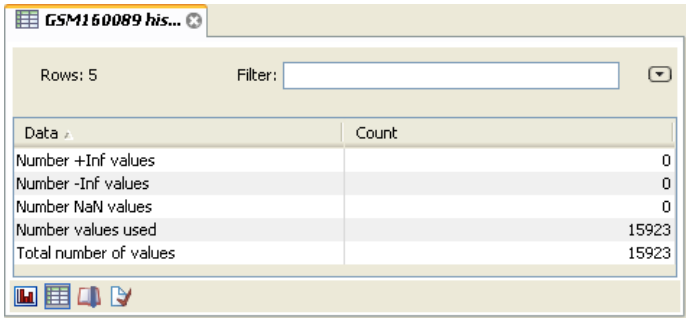



Figure 28.114: *Table view of a histogram.*

The table lists the following properties:

- **Number +Inf values**
- **Number -Inf values**
- **Number NaN values**
- **Number values used**
- **Total number of values**

### 28.11.2 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** | **Transcriptomics Analysis (** )| **General Plots** | **Create MA Plot (** **)**

Select two samples ( ( ), ( ) or ( )). Clicking **Next** will display a dialog as shown in figure 28.115.

In this dialog, you select the values to be used for creating the MA plot (see section 28.6.1).

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

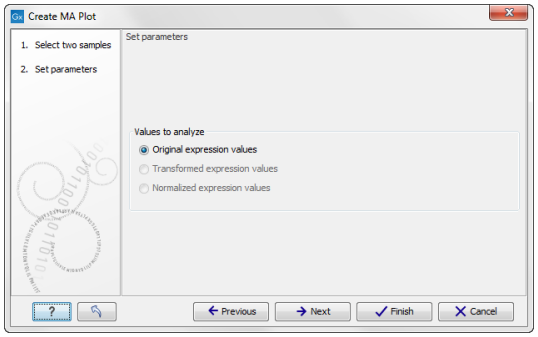



Figure 28.115: *Selcting which values the MA plot should be based on.*

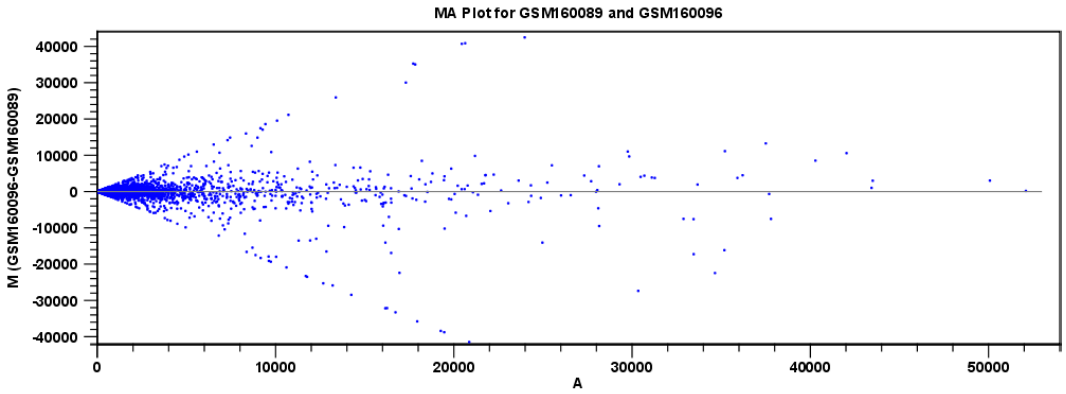



Figure 28.116: *MA plot based on original expression values.*

### Viewing MA plots

The resulting plot is shown in a figure 28.116.

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 28.116 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 28.6.2).

Figure 28.117 shows the same two samples where the MA plot has been created using log2 transformed values.

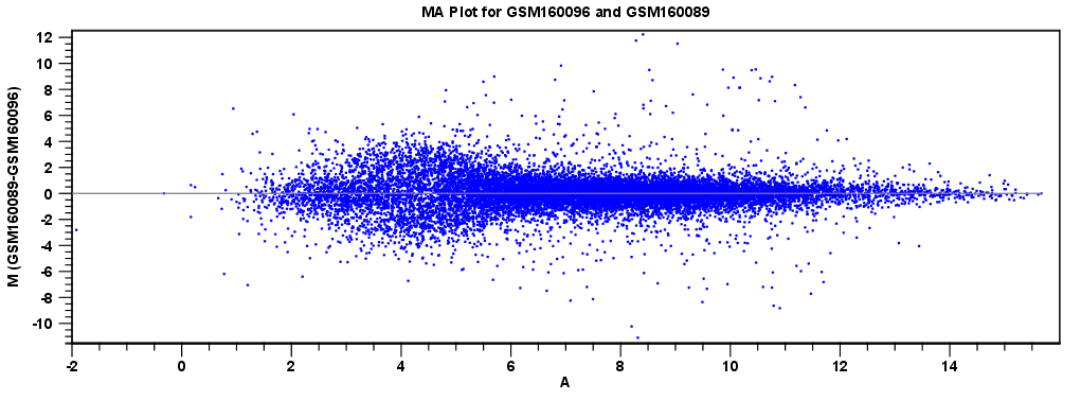



Figure 28.117: *MA plot based on transformed expression values.*

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.
  - Outside
  - Inside
- **Tick lines at.** Choosing Major ticks will show a grid behind the graph.
  - None
  - Major ticks
- **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
    - Wide
  - **Line type**
    - None
    - Line
    - Long dash
    - Short dash
  - **Line color.** Allows you to choose between many different colors. Click the color box to select a color.
- **Line width**
  - Thin
  - Medium

  - Wide

- **Line type**
  - None
  - Line
  - Long dash
  - Short dash

- **Line color.** Allows you to choose between many different colors. 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**
  - None
  - Cross
  - Plus
  - Square
  - Diamond
  - Circle
  - Triangle
  - Reverse triangle
  - Dot

- **Dot 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 scatter plot, you need to save the settings of the **Side Panel** (see section 4.6).

### 28.11.3 Scatter plot

As described in section 28.4.5, an experiment can be viewed as a scatter plot. However, you can also create a "stand-alone" scatter plot of two samples:

**Toolbox** | **Transcriptomics Analysis (**  **)**| **General Plots** | **Create Scatter Plot (**  **)**

Select two samples ( (  ), (  ) or (  )). Clicking **Next** will display a dialog as shown in figure 28.118.

In this dialog, you select the values to be used for creating the scatter plot (see section 28.6.1).

Click **Next** if you wish to adjust how to handle the results (see section 8.2). If not, click **Finish**.

For more information about the scatter plot view and how to interpret it, please see section 28.4.5.

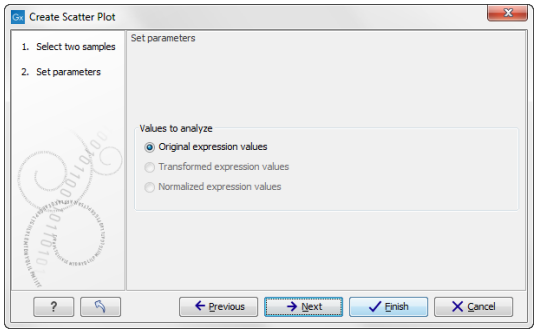



Figure 28.118: *Selcting which values the scatter plot should be based on.*

# Chapter 29

# De novo sequencing

**Contents**


**29.1** *De novo* **assembly** . . . . . **830**
29.1.1 How it works . . . . . 831
29.1.2 Resolve repeats using reads . . . . . 833
29.1.3 Automatic paired distance estimation . . . . . 836
29.1.4 Optimization of the graph using paired reads . . . . . 837
29.1.5 AGP export . . . . . 839
29.1.6 Bubble resolution . . . . . 840
29.1.7 Converting the graph to contig sequences . . . . . 841
29.1.8 Summary . . . . . 841
29.1.9 Randomness in the results . . . . . 842
29.1.10 SOLiD data support in de novo assembly . . . . . 842
29.1.11 De novo assembly parameters . . . . . 843
29.1.12 De novo assembly report . . . . . 845
**29.2 Map reads to contigs** . . . . . **846**


## 29.1 *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).s

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 29.1.1.
2. Second, all the reads are mapped using the simple contig sequence as reference. This is done in order to show e.g. coverage levels along the contigs and enabling 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 have the simple contig sequences as output, this can be chosen when starting the de novo assembly (see section 29.1.11).

### 29.1.1 How it works

CLC bio's de novo assembly algorithm works by using de Bruijn graphs. This is similar to how most new de novo assembly algorithms work [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 (the word size is 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 29.1.

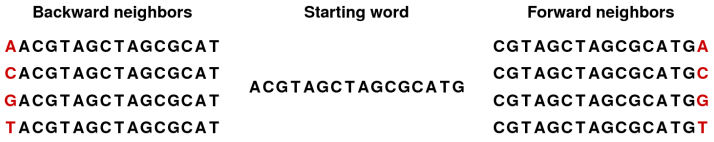



Figure 29.1: *The word in the middle is 16 bases long, and it shares the 15 first bases with the backward neighboring word and the last 15 bases with the forward neighboring word.*

Typically, only one of the backward neighbors and one of the forward neighbors will be present in the table. A graph can then be made where each node is a word that is present in the table and edges connect nodes that are neighbors. This is called a de Bruijn graph.

For genomic regions without repeats or sequencing errors, we get long linear stretches of connected nodes. We may choose to reduce such stretches of nodes with only one backward and one forward neighbor into nodes representing sub-sequences longer than the initial words.

Figure 29.2 shows an example where one node has two forward neighbors:




Figure 29.2: *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 29.3.

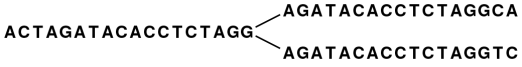



Figure 29.3: *The five nodes are compacted into three. Note that the first node is now 18 bases and the second nodes are each 17 bases.*

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 length 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 29.1.6) as shown in figure 29.4.




Figure 29.4: *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, we would see that one path through the bubble will only be words seen a single time. On the other hand if this was a heterozygous SNP we would see both paths represented more or less equally. 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 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.

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 29.5.




Figure 29.5: *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 length 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 extra nucleotide in the words, 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 length is 20. If the word length is 25, we get only 11 words from each read.

To strike a balance, CLC bio's de novo assembler chooses a word length 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. Please note that the range of word sizes is 12-24 on 32-bit computers and 12-64 on 64-bit computers. See how to adjust the word size in section 29.1.11

### 29.1.2 Resolve repeats using reads

Having build the de Bruijn graph using words CLC bio's 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.

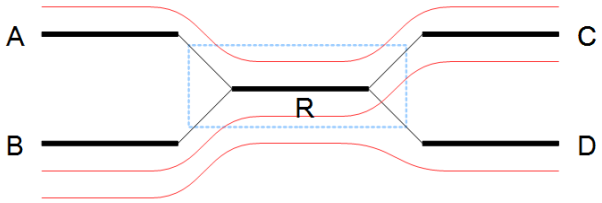



Figure 29.6: *A set of nodes.*

In the example in figure 29.6 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 29.7.

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 29.8.

The algorithm for resolving repeats without conflict can be described the following way:

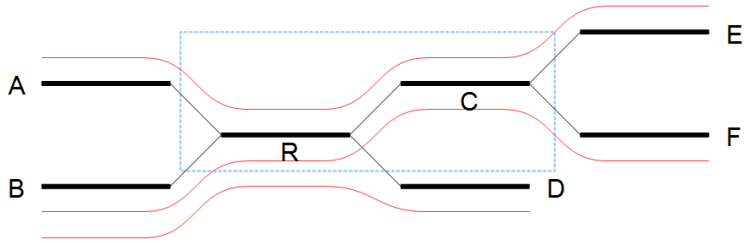



Figure 29.7: *Expanding the window to include more nodes.*

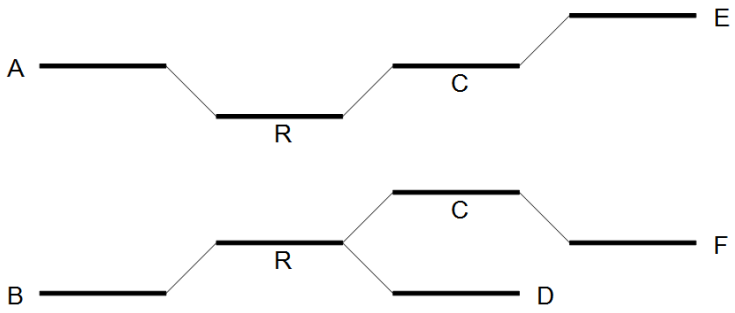



Figure 29.8: *Resolving the repeat.*

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 29.9 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 29.10.

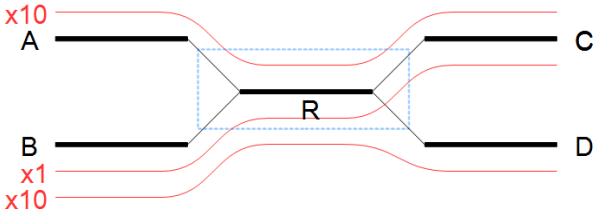



Figure 29.9: *A repeat with conflicts.*

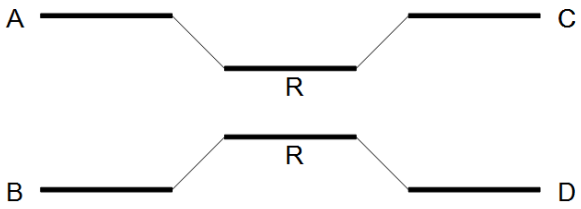



Figure 29.10: *Resolving a repeat with conflicts.*

### 29.1.3 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 29.11). 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 contain $<$2X observations compared to the smaller peak, the distance estimate is only computed if one peak was at a positive distance and the other was at a negative distance. 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.

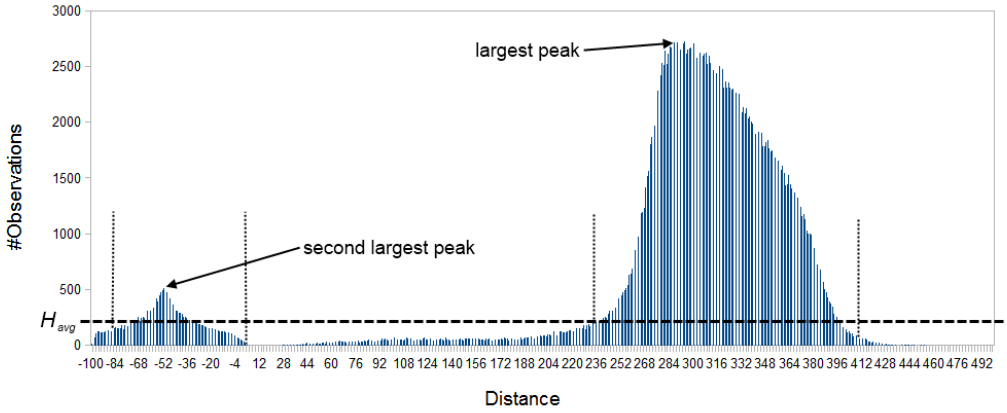



Figure 29.11: *Histogram of paired distances where $H_{avg}$ is indicated by the horizontal dashed line. There is two peaks, one is at a negative distance while the other larger peak is at a positive distance. The extended interval $[k, l]$ for each peak is indicated by the vertical dotted lines.*

### 29.1.4 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 29.12).

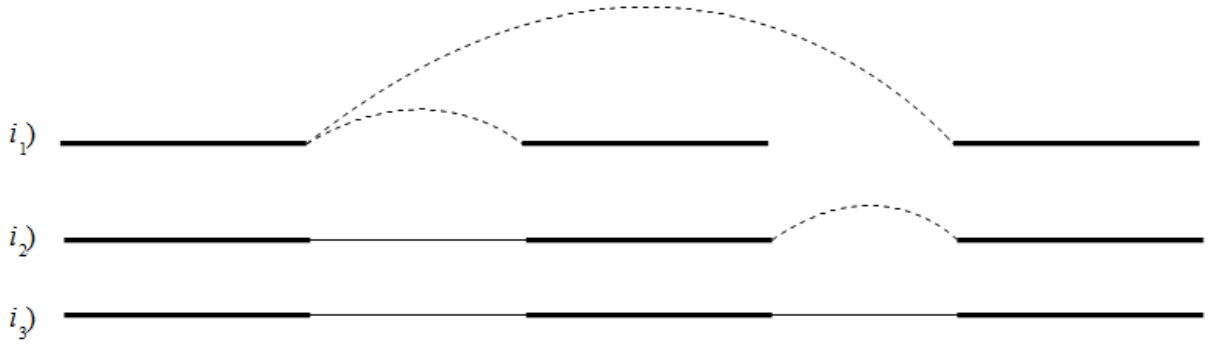



Figure 29.12: *Performing iterative scaffolding of the shortest gaps allows long pairs to be optimally used.* $i_1$ *shows three contigs with dashed arches indicating potential scaffolding.* $i_2$ *is after first iteration when the shortest gap has been closed and long potential scaffolding has been updated.* $i_3$ *is the final results with three contigs in one scaffold.*

Contigs in the same scaffold are output as one large contig with Ns inserted in between. The number of Ns inserted correspond to the estimated distance between contigs, which is calculated based on the paired read information. More precisely, for each set of paired reads spanning two contigs a distance estimate is calculated based on the supplied distance between the reads. The average of these distances is then used as the final distance estimate. The distance estimate will often be negative which happens when the paired information indicate that two contigs overlap. The assembler will attempt to align the ends of such contigs and if a high quality overlap is found the contigs are joined into a single contig. If no overlap is found, the distance estimate is set to two so that all remaining scaffolds have positive distance estimates.

Furthermore, Ns can also be present in output contigs in cases where input sequencing reads themselves contain Ns.

Please note that in CLC Genomics Workbench 6.0.1, Genomics Server 5.0.1, Assembly Cell 4.0.2 and all earlier versions of these products a performance optimization gave rise to Ns being inserted in certain non-scaffold regions, which in the current version can be solved with reads covering such specific regions.

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 29.1.12). This information can also be exported in AGP format (see section 29.1.5).

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, 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 join of two contigs connected by a repeat or another ambiguous structure in the graph, which 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, which resulted in a join of two contigs.

### 29.1.5 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 29.13).

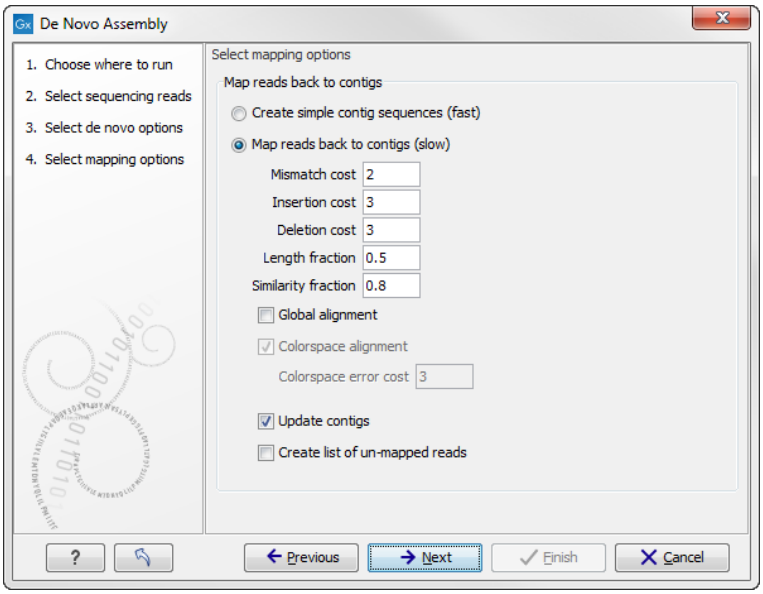



Figure 29.13: *Select "update contigs" by ticking the box if you want to resolve scaffolds based on a read mapping.*

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.

### 29.1.6 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 29.14.




Figure 29.14: *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 29.15 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.

```
AGATGACCAGGGTGTCGATAAAAAATGCCAATCATCTGGAC
AGATGACCAGGGTGTCGAT-AAAAATGCCAATCATCTGGAC
AGATGACCAGGGTGTCGAT-AAAAATGCCAATCATCTGGAC
AGATGACCAGGGTGTCGAT-AAAAATGCCAATCATCTGGAC
AGATGACCAGGGTGTCGATAAAAAATGCCAATCATCTGGAC
AGATGACCAGGGTGTCGATAAAAAATGCCAATCATCTGGAC
AGATGACCAGGGTGTCGAT-AAAAATGCCAATCATCTGGAC
AGATGACCAGGGTGTCGAT-AAAAATGCCAATCATCTGGAC
AGATGACCAGGGTGTCGAT-AAAAATGCCAATCATCTGGAC
AGATGACCAGGGTGTCGAT-AAAAATGCCAATCATCTGGAC
```

Figure 29.15: *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 29.16).

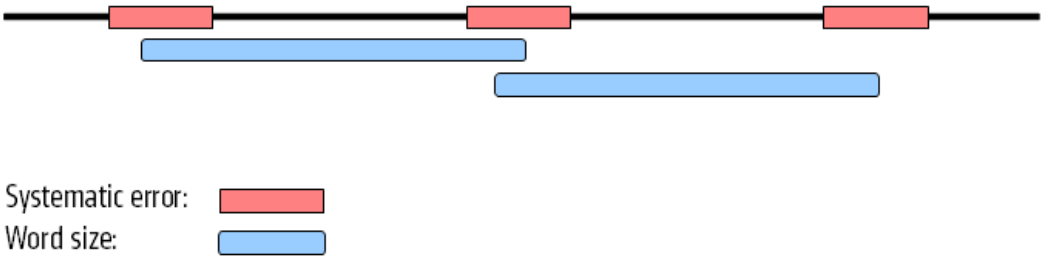



Figure 29.16: *Several sites of errors that are close together compared to the word size.*

In this case, the bubble will be very large because there are no complete words in the regions between the homopolymer sites, and the graph will look like figure 29.17.

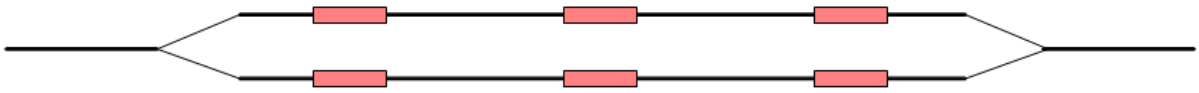


Figure 29.17: *The bubble in the graph gets very large.*

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 29.17, a bubble size spanning the three error sites will mean that the bubble will be resolved (see figure 29.18).

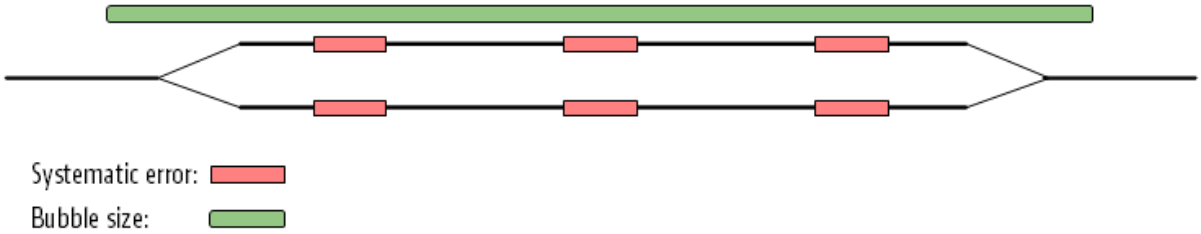



Figure 29.18: *The bubble size needs to be set high enough to encompass the three sites.*

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.

### 29.1.7 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 29.1.4.

### 29.1.8 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.

### 29.1.9 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.

We are working on addressing the fact that slightly different output is returned with different runs of the de novo assembler without appreciably affecting the speed of the assembler. For the moment, the output of runs may vary slightly, but the overall information content of the assembly should not be markedly different.

### 29.1.10 SOLiD data support in de novo assembly

SOLiD sequencing is done in color space. When viewed in nucleotide space this means that a single sequencing error changes the remainder of the read. An example read is shown in figure 29.19.




Figure 29.19: *How an error in color space leads to a phase shift and subsequent problems for the rest of the read sequence*

Basically, this color error means that C's become A's and A's become C's. Likewise for G's and T's. For the three different types of errors, we get three different ends of the read. Along with the correct reads, we may get four different versions of the original genome due to errors. So if SOLiD reads are just regarded in nucleotide space, we get four different contig sequences with jumps from one to another every time there is a sequencing error.

Thus, to fully accommodate SOLiD sequencing data, the special nature of the technology has to be considered in every step of the assembly algorithm. Furthermore, SOLiD reads are fairly short

and often quite error prone. Due to these issues, we have chosen not to include SOLiD support in the first algorithm steps, but only use the SOLiD data where they have a large positive effect on the assembly process: when applying paired information.

### 29.1.11 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 29.20.

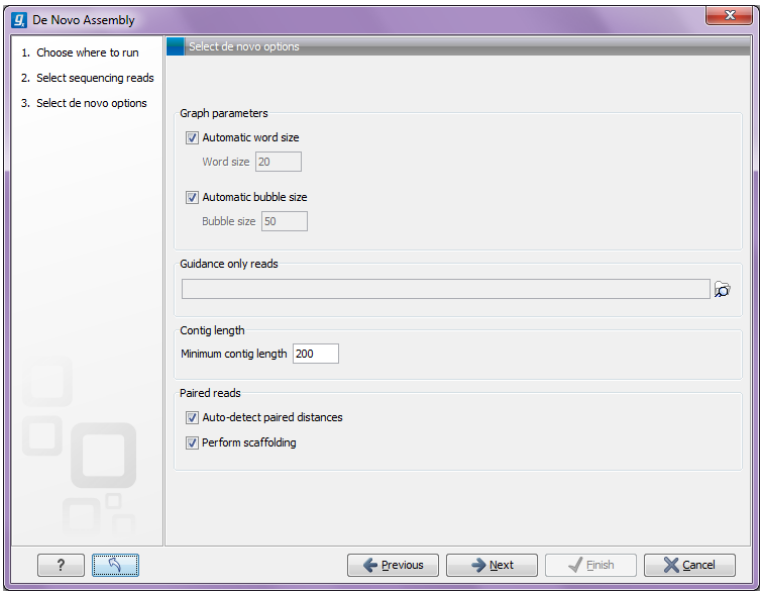



Figure 29.20: *Setting parameters for the assembly.*

At the top, you select the **Word size** and the **Bubble size** to be used. The principles of setting the word size are described in section 29.1.1. When using automatic calculation, you can see the word size in the **History** ( ) of the result files. Please note that the range of word sizes is 12-24 on 32-bit computers and 12-64 on 64-bit computers.

The meaning of the bubble size parameter is explained in section 29.1.6. 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 29.1.2 and section 29.1.4). With mixed data sets from

different sequencing platforms, we recommend using sequencing data with low error rates as the main input for the assembly, whereas data with more errors should be specified only as **Guidance only reads**. This would typically be long reads or paired data sets.

You can also specify the **Minimum contig length** when doing de novo assembly. Contigs below this length will not be reported. The default value is 200 bp. For very large assemblies, the number of contigs can be huge (over a million), in which case the data structures when mapping reads back to contigs will be very large and take a very long time to handle. In this case, it is a great advantage to raise the minimum contig length to reduce the number of contigs that have to be incorporated into this data structure.

At the bottom, there is an option to **Perform scaffolding**. The scaffolding step is explained in greater detail in section 29.1.4. 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 6.3.8).

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 25.3.1). 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 29.21

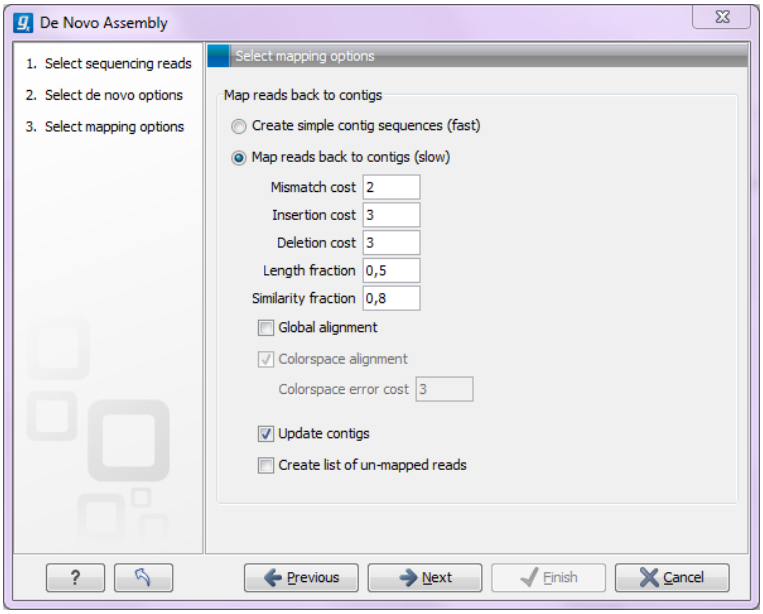



Figure 29.21: *Parameters for mapping reads back to the contigs.*

At the top, you choose whether a read mapping should be performed after the initial contig creation. If you choose to do that, you can specify the parameters for the read mapping. These are all explained in section 25.1.

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 entered into the contig sequence. Contig regions where no reads map will be removed. Note that if such a region occurs within a contig, it is removed and the surrounding regions are joined together.
- The most common nucleotide among the mapped reads at a given position is the one assigned to the contig sequence. In NGS data, it would be very unlikely that at a given position there would be an equal number of reads with different nucleotides. Should this occur however, then the nucleotide that comes first in the alphabet would be included in the consensus.

Note that if this option is selected, the contig lengths may get below the threshold specified in figure 29.20 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.

### 29.1.12 De novo assembly report

In the last dialog of the de novo assembly, you can choose to create a report of the results (see figure 29.22).

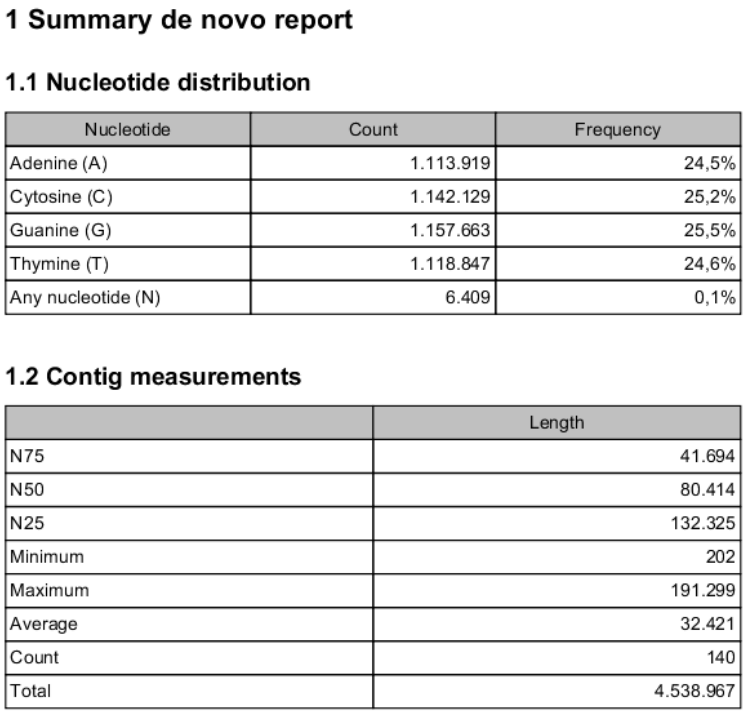
The report contains the following information when both scaffolding and read mapping is performed:

**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.



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 29.22: *Creating a de novo assembly report.*

**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.

**Contig length distribution** . A graph showing the number of contigs of different lengths.

**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.

**Mapping information** . The rest of the sections provide statistics from the read mapping (if performed). These are explained in section 25.3.2.

## 29.2 Map reads to contigs

The "Map reads to contigs" tool allows mapping of reads to contigs. This can be relevant in situations 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

Hence, in any situation where the reference of a mapping is contigs, the "Map reads to contigs" tool can be useful. The "Map reads to contigs" tool is similar to the "Map reads to Reference"

tool in that both tools make use of the same read mapper and accept the same input reads. 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, which is 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 29.23.

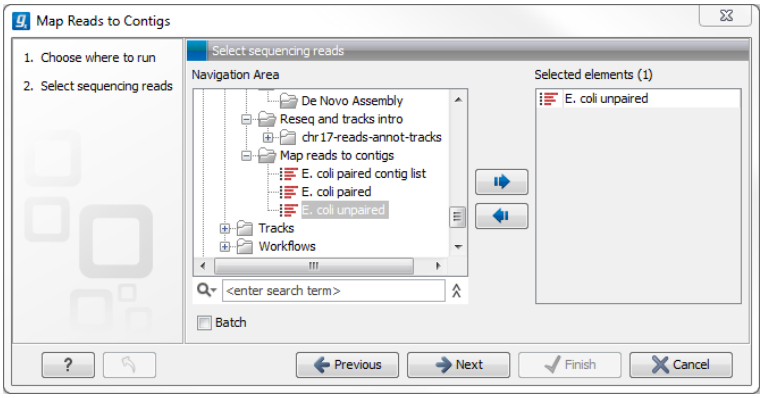



Figure 29.23: *Select reads. The contigs will be selected in the next step.*

The next step is to select the contigs to map the reads against (figure 29.24). Under "Contig masking", specify whether to include or exclude specific regions (for a description of this see section 25.1.2). The contigs can be updated as part of the "Map Reads to Contigs" tool by selecting "Update contigs" at the bottom of the wizard. The advantage of using the read mapping in "Map Reads to Contigs" tool to update the contigs is that the read mapper is better than the de novo assembler at handling errors in reads.

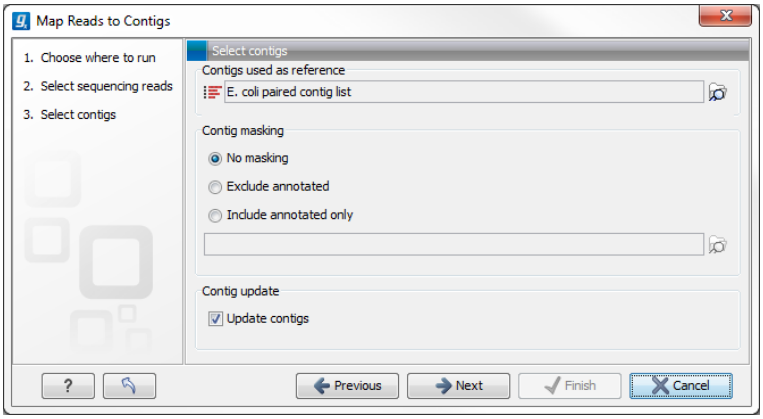



Figure 29.24: *Select contigs and specify whether to use masking and the "Update contigs" function.*

The next wizard steps are identical to the steps found in the "Map Reads to Reference" tool. For a description of these steps, please see section 25.1.3).

The output from the "Map Reads to Contigs" tool depends on whether tracks or stand-alone read mappings were selected in the last dialog. When stand-alone read mappings have been selected as output, it is possible to edit and delete in the contig sequences. Figure 29.25 shows the result of 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. Note that the output from the "Map Reads to Contigs" do not have a consensus sequence as the Contig itself will be the consensus sequence if "Update contigs" was selected.

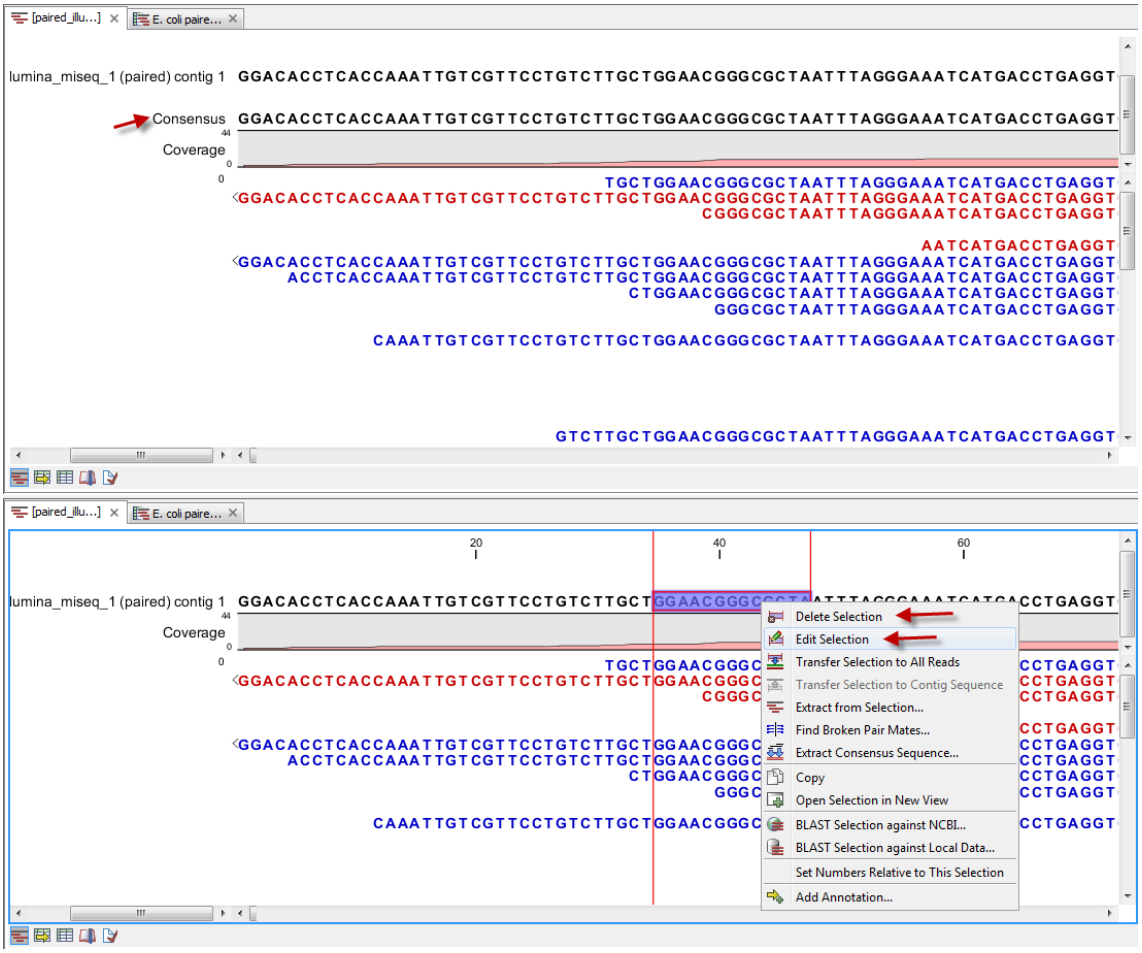



Figure 29.25: *Two different read mappings performed with "Map Reads to Reference" (top) and "Map Reads to Contigs" (bottom). The differences are highlighted with red arrows.*

By selecting "Update contigs" at the bottom of the wizard, the contigs generated by the de novo assembly are used as references that the reads used for the assembly input are mapped back to. The contigs themselves are updated based on the mapping results of the "Map Reads to Contigs". One advantage of using the read mapping in "Map Reads to Contigs" tool to update the contigs 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 the surrounding regions of the contig to be put together as one (figure 29.25).
- 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.

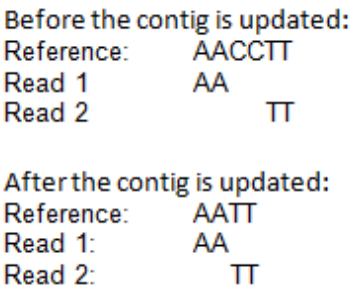



Figure 29.26: *When selecting "Update Contig" in the wizard, contigs will be updated according to the reads. This means that regions of a contig where no reads map will be removed.*

# Chapter 30

# Epigenomics

**Contents**


**30.1 ChIP-Seq Analysis . . . . . 850**
30.1.1 Quality Control of ChIP-seq data . . . . . 851
30.1.2 Learning peak shapes . . . . . 852
30.1.3 Applying peak shape filters to call peaks . . . . . 853
30.1.4 Running the ChIP-Seq Analysis tool . . . . . 853
**30.2 Annotate with nearby gene information . . . . . 857**


## 30.1 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 30.1).

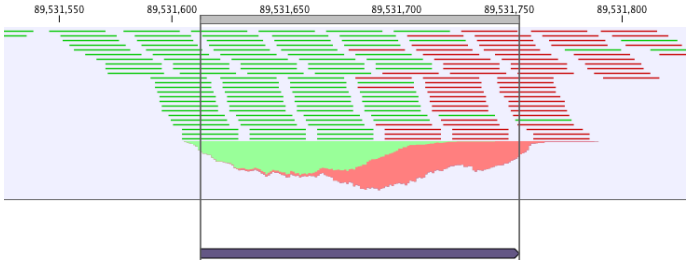



Figure 30.1: *Distribution of forward (green) and reverse (red) reads around a binding site of the transcription factor NRSF.*

The tool makes use of this characteristic shape to identify enriched regions (peaks) in ChIP-seq data.

### 30.1.1 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 30.2), 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 30.2) for successful ChIP-seq experiments.

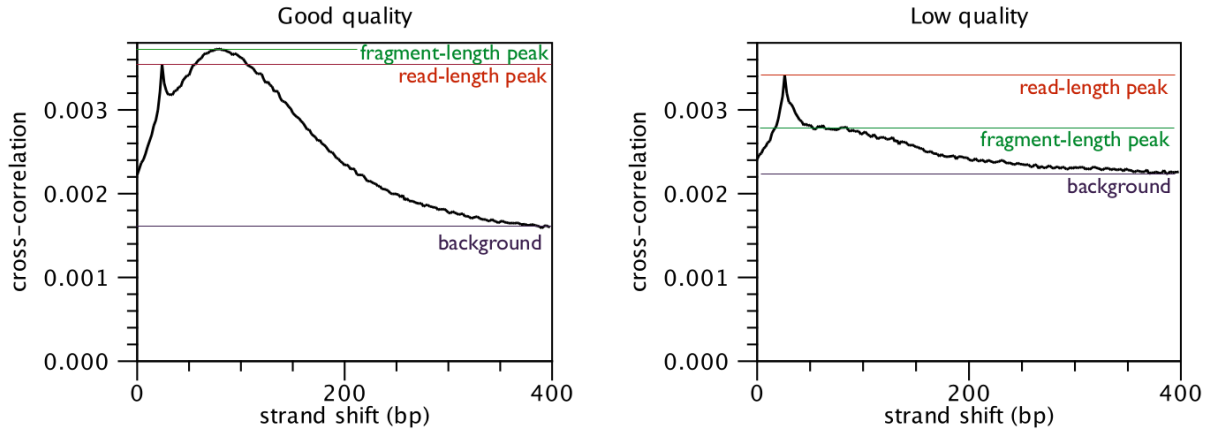



Figure 30.2: *Difference in cross-correlation profiles in ChIP experiments of good and low quality.*

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].

### 30.1.2 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 30.3.

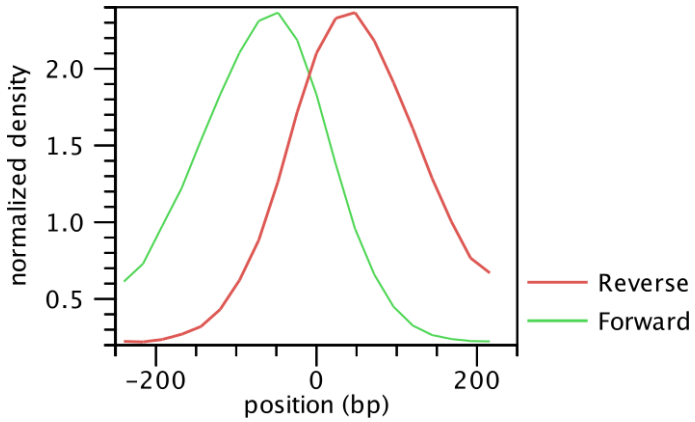



Figure 30.3: *Average peak shape of the transcription factor NRSF.*

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 uses regions with very high coverage in the experiment ChIP-seq as positive examples. If control ChIP-seq experiments are given, regions with high coverage in the control and low in the experimental ChIP-seq data are used as negative examples, as they are probably originated from PCR artifacts. If there is no information to build a negative profile from, the profile is estimated from the sequencing noise.

Once the positive and negative regions have been identified, the ChIP-Seq Analysis tool learns a filter that matches the average peak shape, which we term peak shape filter. The filter implemented is called Hotelling Observer and was chosen because it is the matched filter that maximizes the AUCROC (Area Under the Curve of the Receiver Operator Characteristic), one of the most widely used measures for algorithmic performance.

The Hotelling observer $h$ is defined as:

$$h = \left(\frac{R_p + R_n}{2}\right)^{-1} (\mathbb{E}[X_p] - \mathbb{E}[X_n]), \tag{30.1}$$

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 30.4.

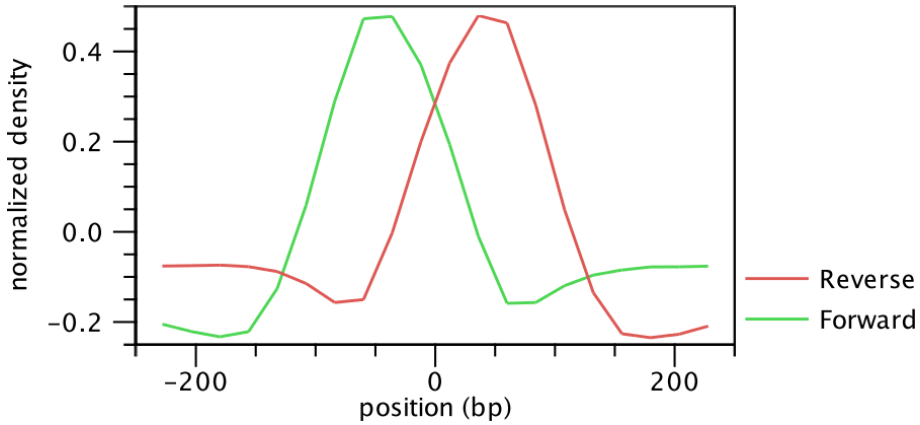



Figure 30.4: *Peak shape filter for the transcription factor NRSF.*

### 30.1.3 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 $= \Phi(-\text{Peak shape score of the peak center})$, where $\Phi$ 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.

### 30.1.4 Running the ChIP-Seq Analysis tool

To run the ChIP-Seq Analysis tool:

**Toolbox** | **Epigenomics Analysis** ( ) | **ChIP-Seq Analysis** ( )

This will open up the wizard shown in figure 30.5 where you can select the input data (the mapped ChIP-seq reads). Multiple inputs (e.g. replicate experiments) are accepted, provided that they refer to the same genome. Track based read mappings ( ) and stand-alone read mappings ( ) / ( ) are both accepted.

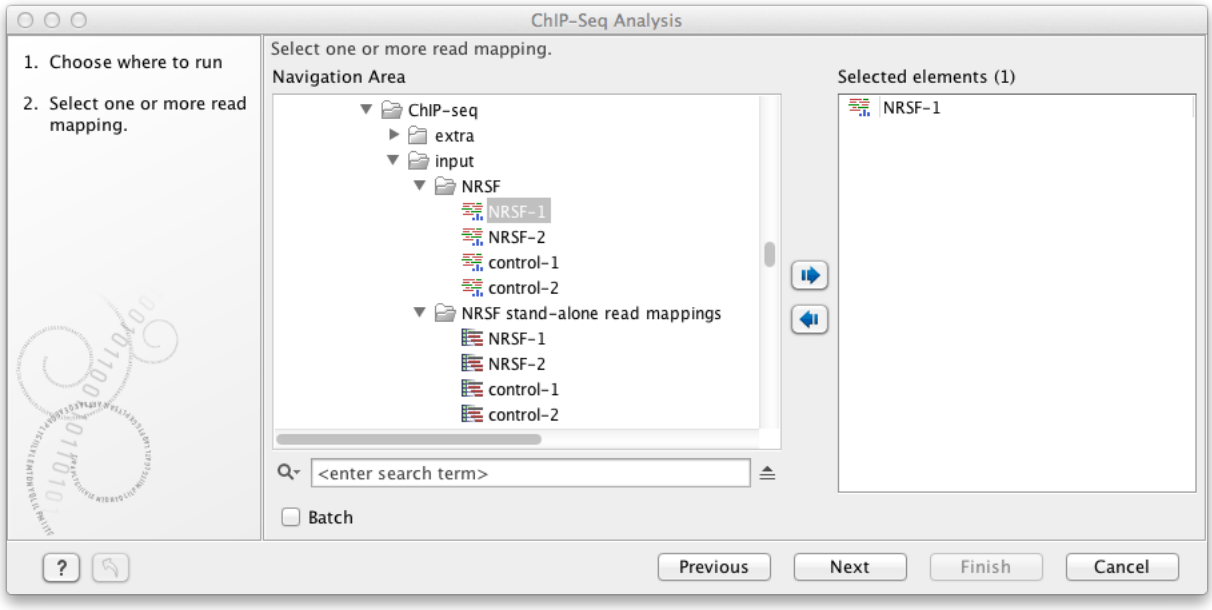



Figure 30.5: *Select the input data for ChIP-Seq Analysis.*

Click on the button labeled **Next** to go to the next wizard step (shown in figure 30.6).

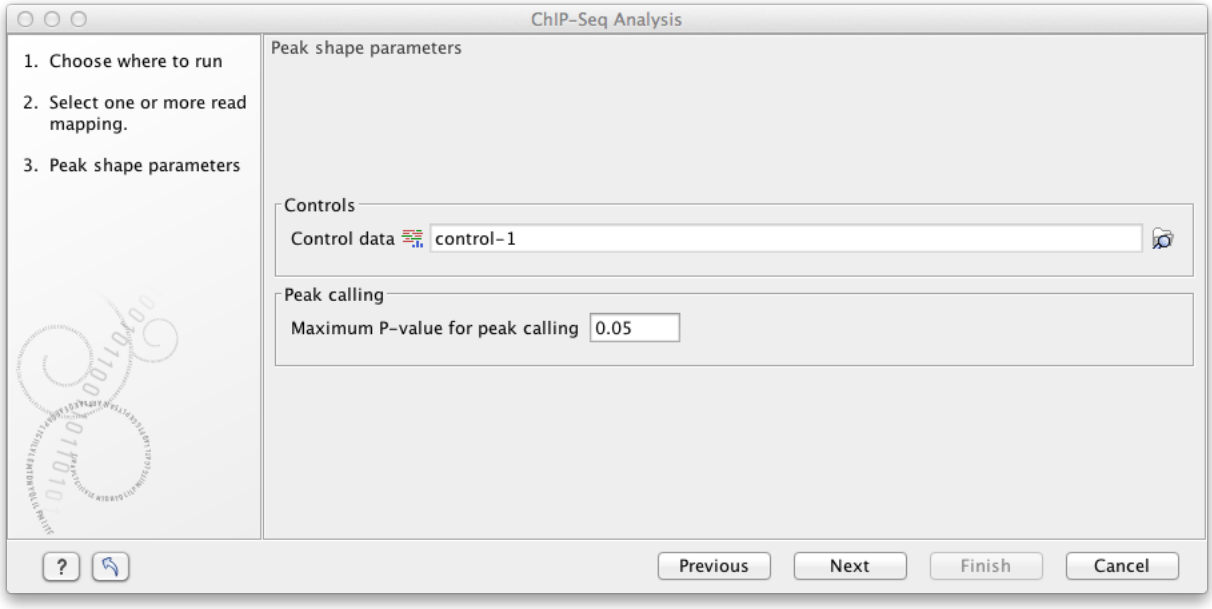



Figure 30.6: *Options for ChIP-Seq Analysis.*

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 the option.

Click on the button labeled **Next** to go to the wizard step shown in figure 30.7.

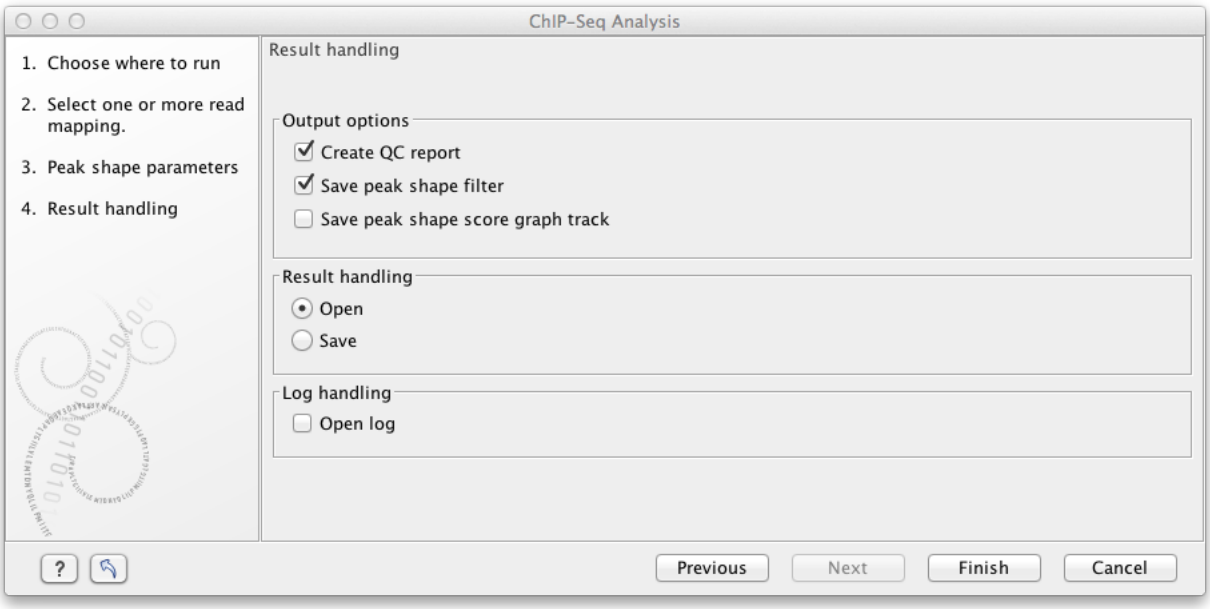



Figure 30.7: *Output options for ChIP-Seq Analysis.*

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 30.1.1.
- **Peak shape filter** ( ) The peak shape filter contains the Hotelling Observer filter that was learned by ChIP-Seq Analysis. For the definition of Peak shape, see section 30.1.3.
- **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 30.1.3.

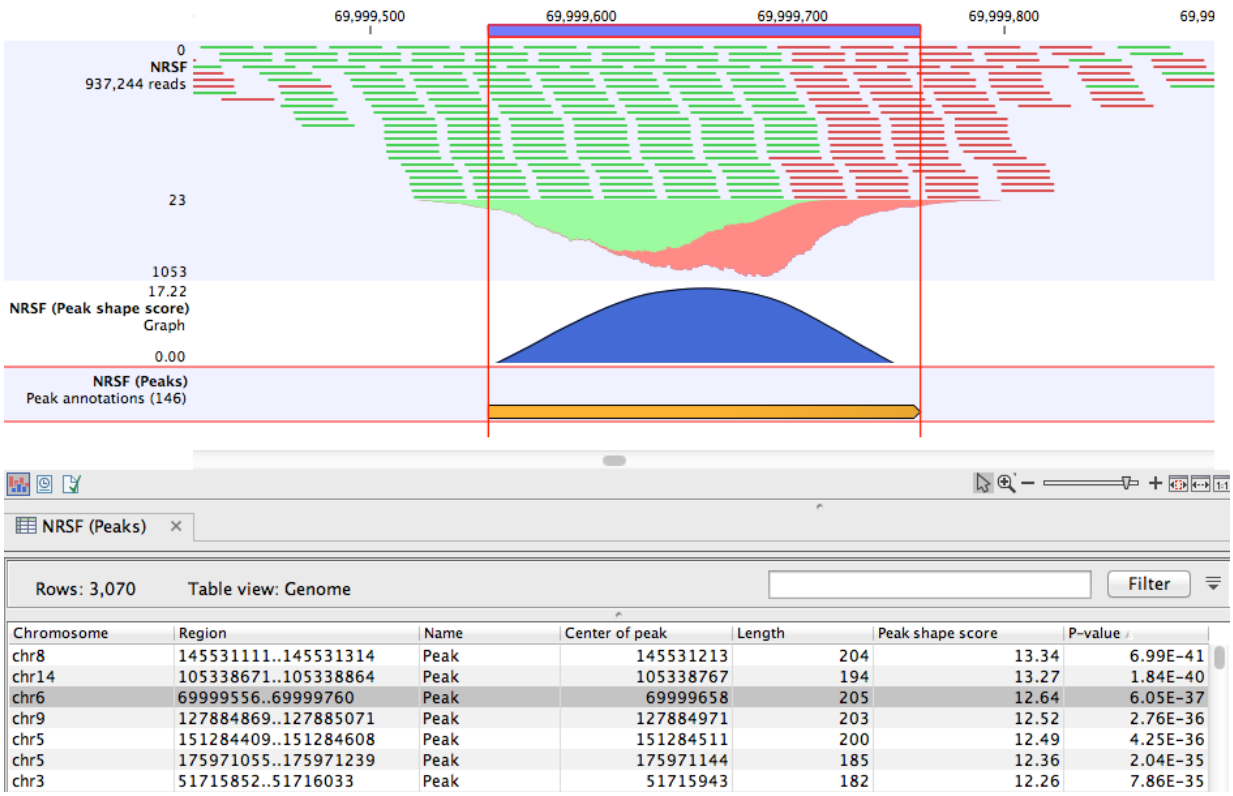
Choose whether you want to open the results directly, or save the results in the **Navigation Area**. If you choose to save the results, you will be asked to specify where you would like to save them.

### Peak track

The main result of the ChIP-Seq Analysis 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:

- **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 (see section 30.1.3).
- **Length** The length of the peak.
- **Peak shape score** The peak shape score of the peak (see section 30.1.3).
- **P-value** The p-value of the peak (see section 30.1.3).

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://www.clcsupport.com/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 30.8.




| Chromosome | Region | Name | Center of peak | Length | Peak shape score | P-value |
|---|---|---|---|---|---|---|
| chr8 | 145531111..145531314 | Peak | 145531213 | 204 | 13.34 | 6.99E-41 |
| chr14 | 105338671..105338864 | Peak | 105338767 | 194 | 13.27 | 1.84E-40 |
| chr6 | 69999556..69999760 | Peak | 69999658 | 205 | 12.64 | 6.05E-37 |
| chr9 | 127884869..127885071 | Peak | 127884971 | 203 | 12.52 | 2.76E-36 |
| chr5 | 151284409..151284608 | Peak | 151284511 | 200 | 12.49 | 4.25E-36 |
| chr5 | 175971055..175971239 | Peak | 175971144 | 185 | 12.36 | 2.04E-35 |
| chr3 | 51715852..51716033 | Peak | 51715943 | 182 | 12.26 | 7.86E-35 |

Figure 30.8: *Inspection of the result of the ChIP-Seq Analysis.*

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.

## 30.2 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 30.9).

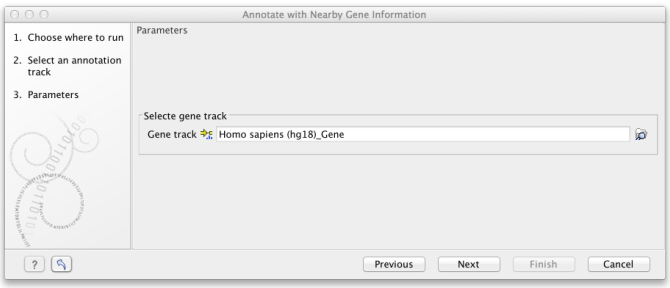



Figure 30.9: *Select gene track.*

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 30.10).
- **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.

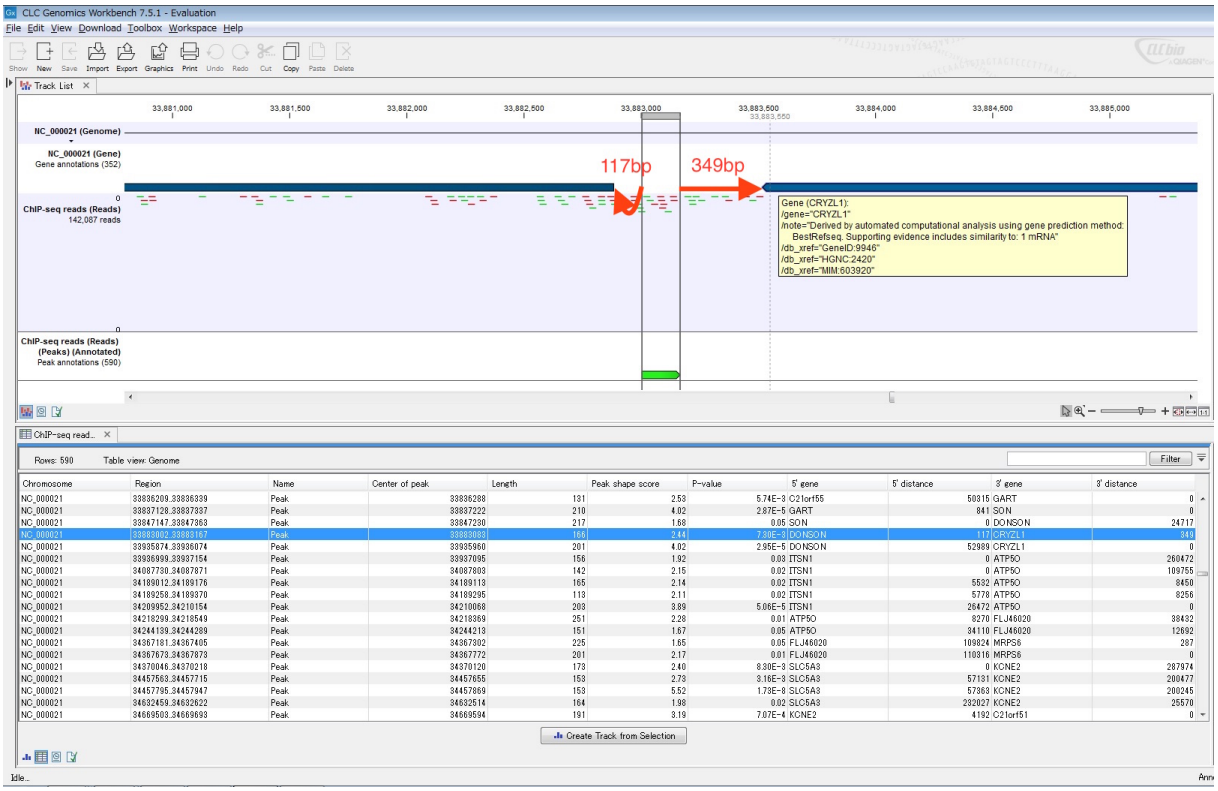



Figure 30.10: *The figure is a split view between a table view and a track list showing the gene annotation track, the ChIP-seq reads, and the annotated ChIP peaks.The red arrows and numbers illustrate the 5' distance and 3' distance.*

# Chapter 31

# Legacy tools

## Contents


**31.1 Quality-based variant detection . . . . . 860**
31.1.1 Assessing the quality of the neighborhood bases . . . . . 860
31.1.2 Significance of variant . . . . . 863
31.1.3 Ploidy and genetic code . . . . . 865
31.1.4 Reporting the variants . . . . . 866
**31.2 Probabilistic variant detection . . . . . 866**
31.2.1 Calculation of the prior and error probabilities . . . . . 867
31.2.2 Calculation of the likelihood . . . . . 868
31.2.3 Calculation of the posterior probability for each site type at each position in the genome . . . . . 869
31.2.4 Comparison with the reference sequence and identification of candidate variants . . . . . 869
31.2.5 Posterior filtering and reporting of variants . . . . . 870
31.2.6 Running the variant detection . . . . . 871
31.2.7 Setting ploidy and genetic code . . . . . 873
31.2.8 Reporting the variants found . . . . . 874
**31.3 ChIP-Seq Analysis (legacy) . . . . . 875**
31.3.1 Peak finding and false discovery rates . . . . . 875
31.3.2 Read shifting . . . . . 877
31.3.3 Peak refinement . . . . . 877
31.3.4 Reporting the results . . . . . 879


A folder with the name **Legacy Tools** (  ) is found at the bottom of the toolbox. The tools found in this folder are tools that have been used in older versions of the Workbench but now have been replaced with new tools. The tools in the Legacy Tools folder will no longer be updated and will be retired in future versions of the Workbench. If possible, we recommend that you base your analysis on the new tools rather than using the legacy tools.

## 31.1 Quality-based variant detection

The quality-based variant detection in *CLC Genomics Workbench* is based on the *Neighborhood Quality Standard (NQS)* algorithm of [Altshuler et al., 2000] (also see [Brockman et al., 2008] for more information). Using a combination of quality filters and user-specified thresholds for coverage and frequency, this tool finds all variants that are covered by aligned reads.

To run the variant detection:

**Toolbox** | **Legacy Tools ( )** | **Quality-based Variant Detection (legacy) ( )**

This opens a dialog where you can select mapping results ( )/ ( )/ ( ) or RNA-Seq analysis results ( ).

Clicking **Next** will display the dialog shown in figure 31.1

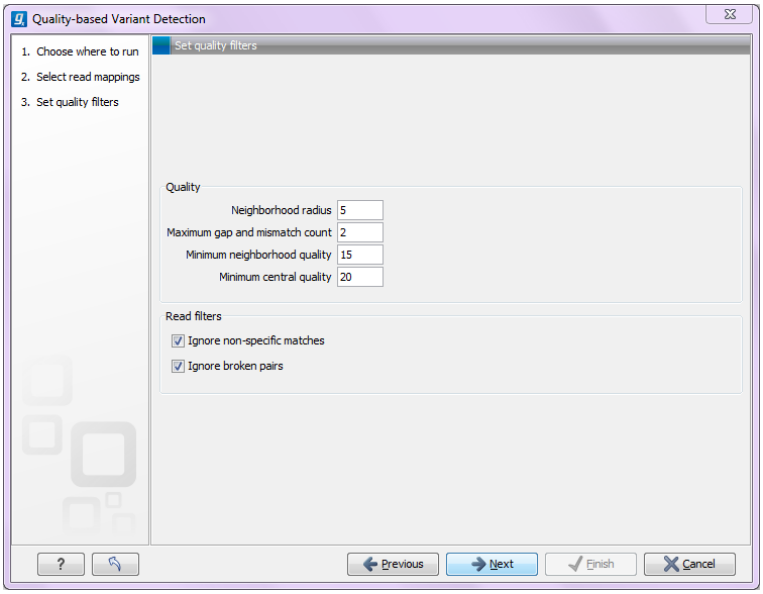



Figure 31.1: *Quality filtering.*

### 31.1.1 Assessing the quality of the neighborhood bases

The variant detection will look at each position in the mapping to determine if there is an SNV, MNV, replacement, deletion or insertion at this position.

Variants that are adjacent are reported as one. E.g. two SNVs next to each other will be reported as one MNV. Similarly, an SNV and an adjacent deletion will be reported as one replacement. Note that variants are only reported as one when they are supported by the same reads.

The size of insertions and deletions that can be found depend on how the reads are mapped: Only indels that are spanned by reads will be detected. This means that the reads have to align both before and after the indel. In order to detect larger insertions and deletions, please use the structural variation tool described in section 27.10 instead.

Please note that the variants reported by the structural variation tool can be fed into the local realignment tool (see section 25.6) to re-adjust the alignment of the reads to span the indels, making some of the indels detected by the structural variation ready to be picked up by the quality-based variant detection.

In order to make a qualified assessment, the quality-based variant detection also considers the general quality of the neighboring bases. The **Neighborhood radius** is used to determine how far away from the current variant this quality assessment should extend, and it can be specified in the upper part of the dialog. Note that at the ends of the read, an asymmetric window of the specified length is used.

If the mapping is based on local alignment of the reads, there will be some reads with un-aligned ends (these ends are faded when you look at the mapping). These unaligned ends are not included in the scanning for variants but they are included in the quality filtering (elaborated below).

In figure 31.2, you can see an example with a neighborhood radius of 5. The current position is high-lighted, and the horizontal high-lighting marks the nucleotides considered for a read with the radius set to 5.

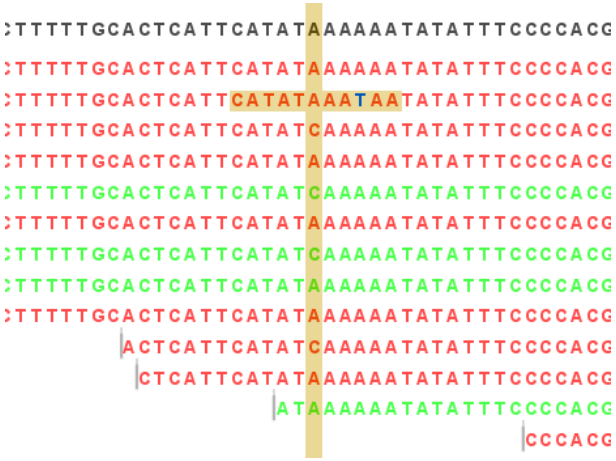


Figure 31.2: *An example of a neighborhood radius of 5 nucleotides.*

For each read and within the given radius,[1] the following two parameters are used to assess the quality:

- **Minimum neighborhood quality**. The average quality score of the nucleotides in a read within the specified radius has to exceed this threshold for the base to be included in the calculation for this position (learn more about importing quality scores from different sequencing platforms in section 6.3).
- **Maximum gap and mismatch count**. The number of gaps and mismatches allowed within the window length of the read. Note that this is excluding the "mismatch" or gap that is considered a potential variant. If there are more gaps or mismatches than this threshold within the radius, this read will not be included in the variant calculation at this position. Unaligned regions (the faded parts of a read) also count as mismatches, *even if some of the bases match*.

Note that for sequences without quality scores, the quality score settings will have no effect. In this case only the gap/mismatch threshold will be used for filtering low quality reads.

[1]The radius is defined as the number of positions in the local alignment between that particular read and the reference sequence (for de novo assembly it would be the consensus sequence).)

Figure 31.2 shows an example of a read with a mismatch, marked in dark blue. The mismatch is inside the radius of 5 nucleotides.

When looking at a position near the end of a read (like the read at the bottom in figure 31.2), the window will be asymmetric as shown in figure 31.3.

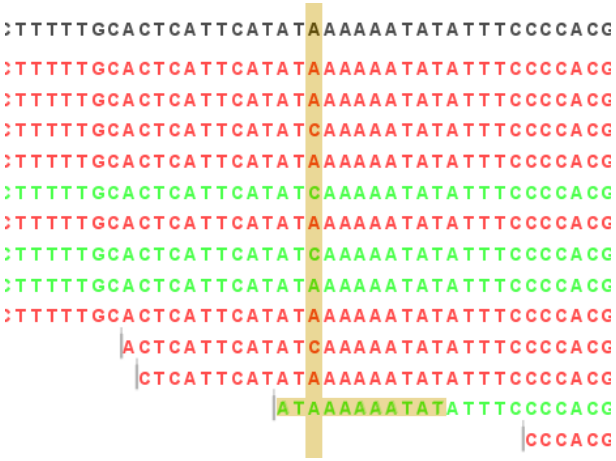


Figure 31.3: *A window near the end of a read.*

Besides looking horizontally within a window for each read, the quality of the central base is also examined: **Minimum quality of central base**. This is the quality score for the central base, i.e. the bases in the column high-lighted in figure 31.4. Bases with a quality score below this value are not considered in the variant calculation at this position.

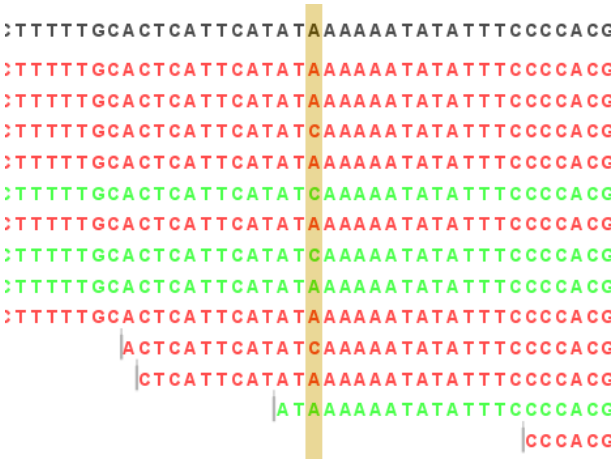


Figure 31.4: *A column of central bases in the neighborhood.*

In addition to low-quality reads, reads can also be filtered further:

**Ignore non-specific matches** This will ignore all reads that are marked as non-specific matches (see section 25.1.3). This is generally recommended, since there is no way of knowing whether the reads and thereby the variant are mapped to the correct position.

**Ignore broken pairs** This will ignore all reads that come from broken pairs (see section 25.1.3). We recommend to switch on the 'Ignore broken reads' filter in case data included paired-reads. As paired-reads have a larger overall alignment with the reference genome, the alignment is more trustworthy than an alignment with a single read, because the probability

that the pair could map somewhere else is lower. However, variants in regions with larger deletions, insertions or rearrangements will be ignored, as broken pairs are often indicators for these kinds of events. Note that if you have mapped a combination of single and paired reads, the reads that were marked as single when running the mapping will still be part of the variant detection, even if you have chosen to ignore broken pairs.

Please note that all the filtering described here means that sometime there is a difference between the coverage of the mapping and the actual counts reported for a variant. The difference would be the number of reads that have been filtered before variant calling.

### 31.1.2 Significance of variant

At a given position, when the reads have been filtered, the remaining reads will be compared to the reference sequence to see if they are different at this position (for *de novo* assembly the consensus sequence is used for comparison). For a variant to be reported, it has to comply with the significance threshold specified in the dialog shown in figure 31.5.

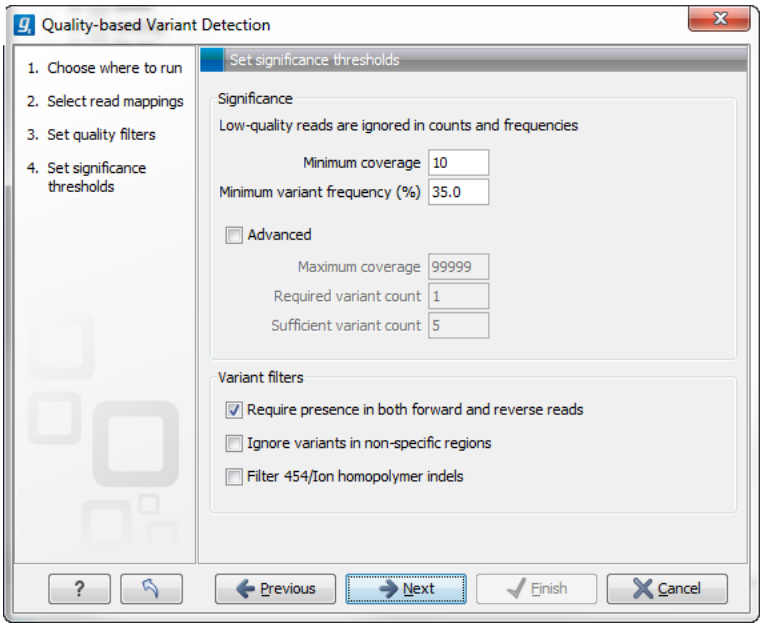



Figure 31.5: *Significance thresholds.*

- **Minimum coverage**. If variants were called in areas of low coverage, you would get a higher amount of false positives. Therefore you can set the minimum coverage threshold. Note that the coverage is counted as the number of valid reads at the current position (i.e. the reads remaining when the quality assessment has filtered out the bad ones).
- **Minimum variant frequency**. This option is the threshold for the number of reads that display a variant at a given position, or in other words, the reported zygosity depends on the setting of the variant frequency parameter. Setting the percentage at 35% means that at least 35% of the validated reads at this position should have a different base than the reference in order to be considered heterozygous rather than homozygous. This means that if, in one reference position, A is represented in more than 35% of the reads and C is also represented in more than 35% of the reads, the variant would be considered heterozygous

because two different alleles were called for the same variant. If one of these bases (A and C in this example) is the reference base, then it will be reported in the variant track as the reference allele variant, but not in the annotated table.

Below, there is an **Advanced** option letting you specify additional requirements. These will only take effect if the **Advanced** checkbox is checked.

- **Maximum coverage**. Although it sounds counter-intuitive at first, there is also a good reason to be suspicious about high-coverage regions. Read coverage often displays peaks in repetitive regions where the alignment is not very trustworthy. Setting the maximum coverage threshold higher than the expected average coverage (allowing for some variation in coverage) can be helpful in ruling out false positives from such regions. You can see the distribution of coverage by creating a detailed mapping report (see section 25.3.1).

  The result table, created by the variant detection, includes information about coverage, so you can specify a high threshold in this dialog, check the coverage in the result afterwards, and then run the variant detection again with an adjusted threshold.

- **Required variant count**. This option is the threshold for the number of reads that display a variant at a given position. In addition to the percentage setting in the simple panel above, this setting is based on absolute counts. If the count required is set to 3, it means that **even though** the required percentage of the reads has a variant base, it will still not be reported if there are less than 3 reads supporting the variant.

- **Sufficient variant count**. This option can be used for deep sequencing data where you have very high coverage and many different alleles. In this case, the percentage threshold is not suitable for finding valid variants only present in a small number of alleles. If the sufficient variant count is set to 5, it means that as long as there are 5 reads supporting a variant, it will be called irrespective of the frequency setting (it still has to be above the required variant count which should always be lower than the sufficient variant count).

When there are **ambiguity** bases in the reads, they will be treated as separate variants. This means that e.g. a `Y` will not be collapsed with `C` or `T` in other reads. Rather, the Ys will be counted separately.

**Variant filters**

Below the significance settings, there are filters that can be useful for removing false positives:

- **Require presence in both forward and reverse 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 (see [Nguyen et al., 2011] for a recent study with Illumina data). This can easily lead to false positive variant calls, and by checking this filter, the minimum ratio between forward and reverse reads supporting the variant should be at least 0.05. In this way, systematic sequencing errors of this kind can be eliminated. The forward/reverse reads balance is also reported for each variant in the result (see section 27.11).

- **Ignore variants in non-specific regions**. Variants in regions covered by one or more non-specific reads are ignored.

- **Filter 454/Ion homopolymer indels**. The 454 and Ion Torrent/Proton sequencing platforms exhibit weaknesses when determining the correct number of the same kind of nucleotides in a homopolymer region (e.g. AAA). This leads to a high false positive rate for calling InDels in these regions. This filter is very basic: it removes all indels that are found within or just next to a homopolymer region. A homopolymer region is defined as at least two consecutive identical bases in the reference.

### 31.1.3 Ploidy and genetic code

Clicking **Next** offers options for setting ploidy and genetic code (see figure 31.7:

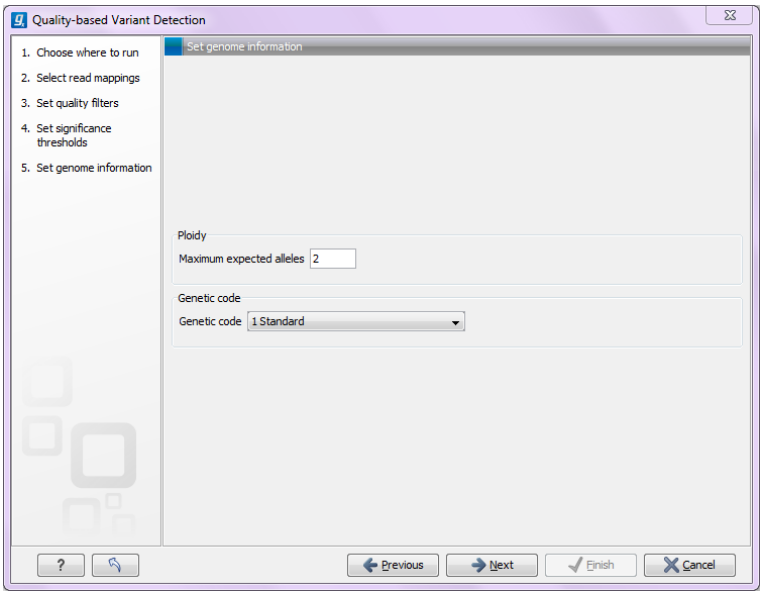



Figure 31.6: *Ploidy and genetic code.*

- **Maximum expected alleles**. Allows the user to flag variants that fall in locations with an unexpectedly high number of observed alleles. For a given variant, the entry in the 'hyper-allelic' column of the variant table will contain 'yes', if more than the user-specified 'maximum expected alleles' is observed at the variant position, other observations will result in 'no'.

  Note, that with this interpretation the "yes" flag holds true regardless of whether the sequencing data are generated from a population sample or from an individual sample. For example, using a minimum variant frequency of 30% with a diploid organism, you are allowing variants with up to 3 different alleles within the sequencing reads, and by then setting the maximum expected variants count to 2 (the default), any variant with 3 different alleles will be marked as "yes".

- **Genetic code**. For the table report, the variant's effect on the protein level is calculated, and the translation table specified here is used. When reporting the variant as a track, this setting has no effect, since the amino acid consequences are calculated separately (see section 27.15.1).

### 31.1.4 Reporting the variants

When you click **Next**, you will be able to specify how the variants should be reported (see figure 31.7).

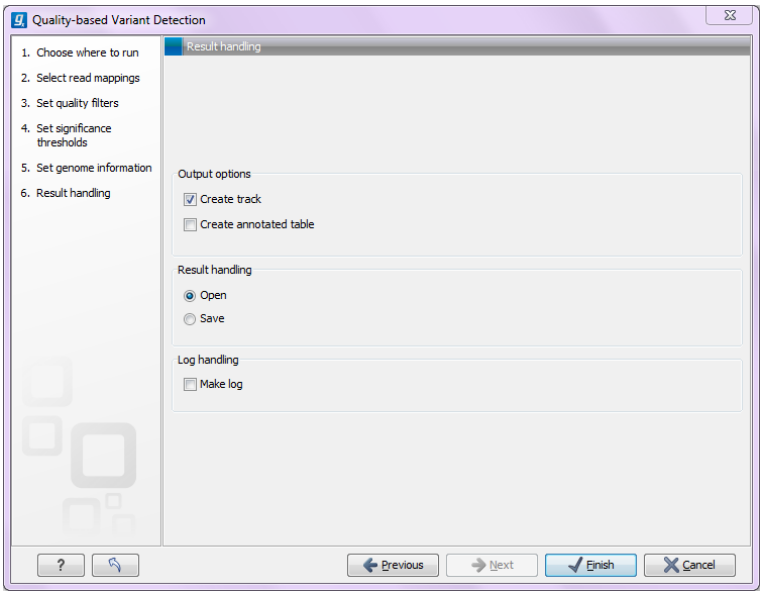



Figure 31.7: *Output options.*

- **Create track**. This will create a variant track that can be further annotated (functional consequences, annotation overlap etc) and used for comparative analysis and visualization(see section 27.13)..

  Note that the track can be displayed in a table view ( ) as well. See a description of the output in section 27.11.1.

- **Create annotated table**. This will create a table showing all the variants including information about overlapping annotations and amino acid changes. See a description of the output in section 27.11.2.

## 31.2 Probabilistic variant detection

The purpose of the Probabilistic Variant Caller is to identify variants in a sample by using a probabilistic model built from read mapping data. This tool can detect variants in data sets from haploid (e.g. Bacteria), diploid (e.g. Human) and polyploid organisms (e.g. Cancer and higher plants) with a high sensitivity and specificity.

The algorithm used is a combination of a Bayesian model and a Maximum Likelihood approach to calculate prior and error probabilities for the Bayesian model.

Parameters are calculated on the mapped reads alone. The reference sequence is not considered at this stage. After observing a certain combination of nucleotides from the reads at every position in the genome, the probability for each combination of alleles is calculated. These probabilities are then used to determine which one of the allele combinations is the most likely combination for each position. In the case where the ploidy is expected to be 2, the types of cases considered would be homozygous A/A, heterozygous A/G, heterozygous A/C and so on. In the case where

the ploidy is expected to be 3, the cases considered would be homozygous A/A/A, heterozygous A/G/C, heterozygous A/C/C and so on. This can then be compared with the reference allele to find out if it is different from the reference sequence and therefore can be called as a variant. Please refer to the white paper at http://www.clcbio.com/white-paper/ for more information including benchmarks.

Variants that are adjacent are reported as one. E.g. two SNVs next to each other will be reported as one MNV. Similarly, an SNV and an adjacent deletion will be reported as one replacement. Note that variants are only reported as one when they are supported by the same reads.

The size of insertions and deletions that can be found depend on how the reads are mapped: Only indels that are spanned by reads will be detected. This means that the reads have to align both before and after the indel. In order to detect larger insertions and deletions, please use the structural variation tool described in section 27.10 instead.

Please note that the variants reported by the structural variation tool can be fed into the local realignment tool (see section 25.6) to re-adjust the alignment of the reads to span the indels, making some of the indels detected by the structural variation ready to be picked up by the probabilistic variant detection.

Probabilistic Variant Detection is not designed to detect low frequency alleles. More specifically, it was not designed for calling variants that are not present in allelic frequencies that are in accordance with the ploidy assumption. Increasing the coverage will lead to higher confidence in the variants that are called, but will not result in the calling of low frequency variants. We recommend using the Low Frequency Variant Detection tool for this purpose (see section 27.5) .

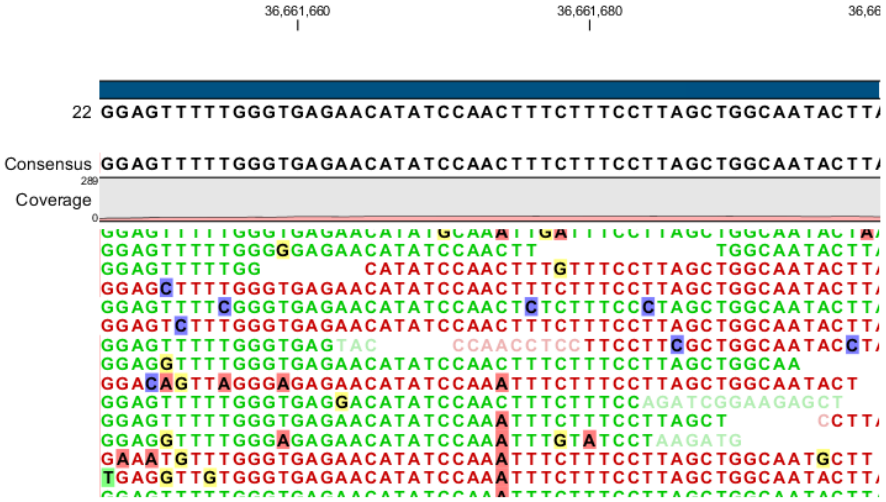



Figure 31.8: *An example of a heterozygous variant surrounded by a lot of noise from sequencing errors.*

### 31.2.1 Calculation of the prior and error probabilities

The prior probabilities are estimated using only the mapped reads through four rounds of Expectation Maximization and are calculated for each potential combination of alleles (site types). Thus, the prior probabilities reflect the likelihood of observing each combination of alleles in the genome studied. The reference sequence is not taken into account during the first part of the analysis. More about the Maximum Likelihood estimation (MLE) can be found at

http://en.wikipedia.org/wiki/Maximum_likelihood.

For a diploid organism, the initial parameters for the priors, which are then updated, are shown in Table 31.1. The sum of the probabilities for all site types is always 1.

| Site Type | Prior probability |
|---|---|
| A/A | 0.2475 |
| A/C | 0.001 |
| A/G | 0.001 |
| A/T | 0.001 |
| T/C | 0.001 |
| T/G | 0.001 |
| T/T | 0.2475 |
| G/C | 0.001 |
| C/C | 0.2475 |
| G/G | 0.2475 |
| G/- | 0.001 |
| A/- | 0.001 |
| C/- | 0.001 |
| T/- | 0.001 |

Table 31.1: Site Types for a diploid organism with example probabilities.

If the expected ploidy level is set to 1, analogous values to table 31.1 are calculated. Here, only the values for the homozygous site types like A, C, G, T and - would be calculated.

If the expected ploidy is set to 3, the analogous values are calculated, which here would be values for site types like A|A|A, A|C|G, G|G|- and so on.

Error probabilities are calculated alongside the priors for each observed allele and assumed reference allele, before the reference sequence is incorporated into the analysis. Table 31.2 illustrates an example of the values calculated in an error probability matrix.

| | **A** | **C** | **G** | **T** | **-** |
|---|---|---|---|---|---|
| **A** | 0.90 | 0.025 | 0.025 | 0.025 | 0.025 |
| **C** | 0.025 | 0.90 | 0.025 | 0.025 | 0.025 |
| **G** | 0.025 | 0.025 | 0.90 | 0.025 | 0.025 |
| **T** | 0.025 | 0.025 | 0.025 | 0.90 | 0.025 |
| **-** | 0.025 | 0.025 | 0.025 | 0.025 | 0.90 |

Table 31.2: Error probability matrix - observed sequenced nucleotide in read versus actual nucleotide at this position.

If quality values are available, an error matrix is calculated for each quality value.

### 31.2.2 Calculation of the likelihood

After the prior and error probabilities have been estimated, the calculation of the likelihood is undertaken. For every combination of reference allele (site types) and nucleotide in every read,

the probability of the observed allele being the same as the reference is calculated. These probabilities are then multiplied for all nucleotides in the reads at that position.

Here is an example:

Assumed reference allele: A/C

Read 1: C [$\frac{1}{2}$ (P(C|A)) + $\frac{1}{2}$(P(C|C))] *

Read 2: C [$\frac{1}{2}$( P(C|A)) + $\frac{1}{2}$(P(C|C))] *

Read 3: A [$\frac{1}{2}$( P(A|A)) + $\frac{1}{2}$(P(A|C))] *

Read 4: A [$\frac{1}{2}$( P(A|A)) + $\frac{1}{2}$(P(A|C))] *

Read 5: T [$\frac{1}{2}$( P(T|A)) + $\frac{1}{2}$(P(T|C))]

Here, P(X|Y) is the probability that we will observe nucleotide X in a read when the true reference sequence is Y.

### 31.2.3 Calculation of the posterior probability for each site type at each position in the genome

Based on the probabilities calculated, one can determine which of the site types is the best fit at each position in the genome. The site type determined to be the most likely at each position can then be compared with the allele in the reference sequence at the same position. If it is likely to be different, it suggests the presence of a variation.

Therefore the posterior probability is formed as follows:

$$P(site\ type|Obs) = \frac{P(Obs|site\ type) * P(site\ type)}{P(Obs)}$$

where

$$P(Obs) = \sum_{Site\ types} P(Obs|site\ type) * P(site\ type)$$

### 31.2.4 Comparison with the reference sequence and identification of candidate variants

Once we have all of the probabilities for each combination of alleles for all positions in the reference sequence, the next step is to determine which of them have the highest probability of existing in the sample. These are the candidate variations. Nucleotide combinations that are the same as the reference sequence are not reported. At this point in the algorithm, a probability threshold is taken into consideration, utilizing a threshold provided by the user.

The threshold provided by the user indicates how sure one would like to be that the candidate variant differs from the reference type. The threshold is applied by the Probabilistic Variant Caller by considering the inverse situation: is the probability of the candidate variant being the same as the reference position lower than 1 minus the threshold. So, for a user-provided threshold of 90%, the Probabilistic Variant Caller requires that any given site type has a probability of less than or equal to 0.1 (i.e. 1 - 0.9) of being the same as the reference type. For example, if a

user gave a threshold of 90%, and a particular position was found to have a probability of 15%, or 0.15, of being the same as the reference (equivalently, having a probability of 85% of being different than the reference), then this position would not be called as a variant. If the threshold had been set to 80%, then this position would have been called as a variant, as 0.15 is less then 0.20, or in other words, the position has a high enough probability of being different than the reference according to the user-defined threshold, to be reported as a variant.

If a variant is called at a given position, the second step performed by the algorithm is to determines the allele combination (type site) with the highest probability. This type site, together with the corresponding probability, will be reported as the candidate variant.

### 31.2.5 Posterior filtering and reporting of variants

The algorithm includes several filters to reduce the rate of false positive variants. These filters can be activated or deactivated by the user.

#### Filtering of variants in homopolymeric regions

Different sequencing platforms generate different types of sequencing errors, which can cause incorrectly called variants. The most common source of sequencing errors across platforms is the determination of nucleotides in so-called homopolymeric regions. These are regions that include stretches of the same nucleotide (e.g. AAAAA or TTTTTTT). As a result of the internal chemistry used on platforms such as 454 and Ion Torrent, the number of identical nucleotides in such regions is often not accurately reported. This causes variant-callers to identify within homopolymer regions, insertions and deletions not actually present in the sample. The Illumina platform has a similar problem in which one nucleotide is surrounded by other nucleotides of the same type (e.g. AAAAGAAAA). Such cases are sometimes misread, with the different base identified as being the same as the surrounding nucleotides. This can lead to incorrect SNV calls. For example, a region of AAAAGAAAA in the sample may appear as AAAAAAAAA in the read. This could lead to a variant allele, A, being called where the G appears in the reference, when in fact the sample itself did contain a G at that position.

The Probabilistic Variant Caller includes an internal filter to recognize and prevent variants being reported in homopolymeric regions.

The 454/Ion Torrent homopolymer filter does not report insertion or deletion variants found at the ends of regions of two or more nucleotides of the same kind (e.g. AA, TT, GGG).

An example is given in figure 31.9:

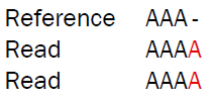



Figure 31.9: *Example of insertions filtered out using the 454/Ion Torrent homopolymer filter.*

The red A will not be reported as a variant when the 454/Ion Torrent filter is applied, as it is characteristic of sequencing errors frequently observed on those platforms.

#### Forward/reverse reads support

This filter is recommended in all cases where an even distribution of forward and reverse reads at every position is expected. However, it should not be used for data sets such as large amplicons, where the ends of an amplicon are likely to be covered by only forward or reverse reads.

Due to sequencing or PCR artifacts and mapping issues, there can be some positions in the reference genome where only forward or only reverse reads are aligned. This can lead to certain alleles being present on one strand only.

If there is a strand bias from sequencing visible in the quality output check after sequencing, these should be regarded as suspicious regions that should be ignored during variant calling. If the user has selected the forward/reverse read support option, only variants that have a forward/reverse read balance of at least 0.05 are reported.

The forward/reverse balance is calculated as:

$$Min((\#forward/\#total)(\#reverse/\#total))$$

where

#forward = number of forward reads supporting the variant
#reverse = number of reverse reads supporting the variant
#total = all reads supporting the variant

### 31.2.6 Running the variant detection

To start the variant calling:

**Toolbox** | **Legacy Tools (** **)** | **Probabilistic Variant Detection (legacy)** ( )

This opens a dialog where you can select mapping results ( )/ ( )/ ( ) or RNA-Seq analysis results ( ).

#### Read filters

Clicking **Next** will display the dialog shown in figure 31.10.

In this dialog, you can specify reads to be filtered away before variant calling:

**Ignore non-specific matches** This will ignore all reads that are marked as non-specific matches (see section 25.1.3). This is generally recommended, since there is no way of knowing whether the reads and thereby the variant are mapped to the correct position.

**Ignore broken pairs** This will ignore all reads that come from broken pairs (see section 25.1.3). We recommend to switch on the 'Ignore broken reads' filter in case data included paired-reads. As paired-reads have a larger overall alignment with the reference genome, the alignment is more trustworthy than an alignment with a single read, because the probability that the pair could map somewhere else is lower. However, variants in regions with larger deletions, insertions or rearrangements will be ignored, as broken pairs are often indicators for these kinds of events. Note that if you have mapped a combination of single and paired reads, the reads that were marked as single when running the mapping will still be part of the variant detection, even if you have chosen to ignore broken pairs.

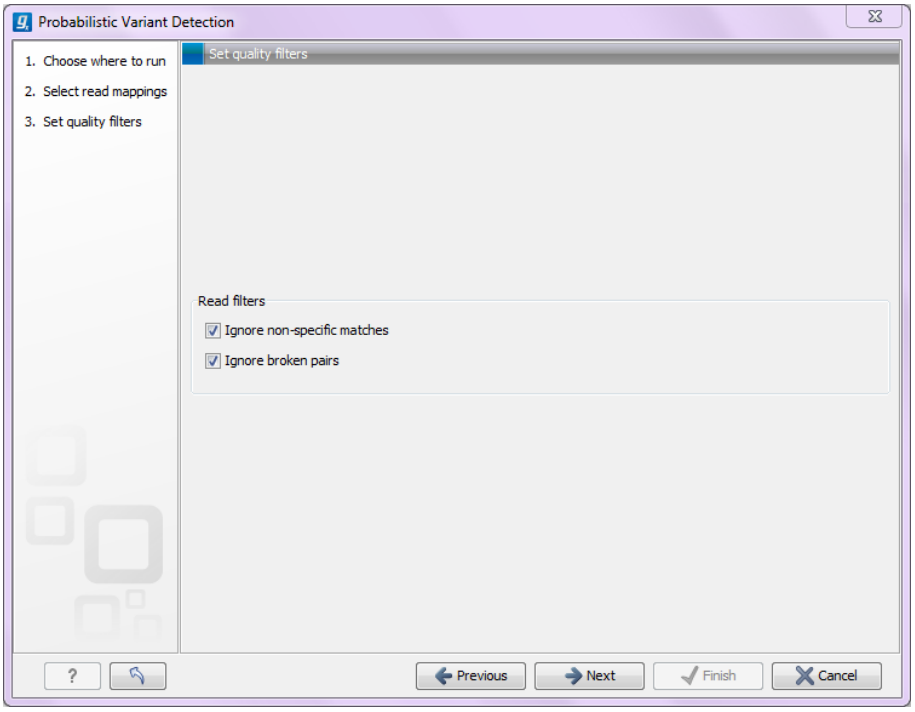



Figure 31.10: *Read filters for the variant detection.*

Please note that all the filtering described here means that sometime there is a difference between the coverage of the mapping and the actual counts reported for a variant. The difference would be the number of reads that have been filtered before variant calling.

### Significance thresholds

Clicking **Next** will display the dialog shown in figure 31.11.

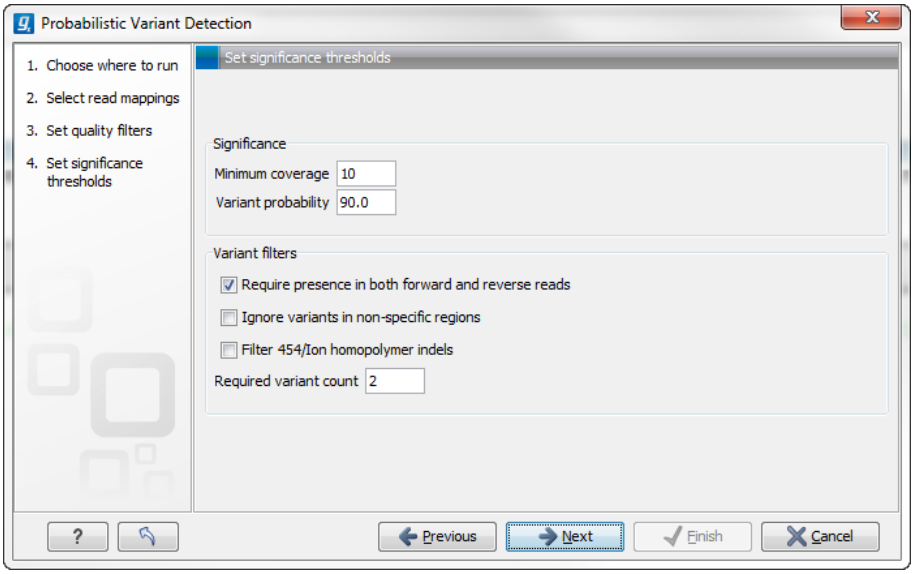



Figure 31.11: *Significance thresholds.*

The follow parameters can be set:
**Significance**

- **Minimum coverage** The minimum number of reads aligned to the site to be considered a potential variant.
- **Variant probability** The minimum total probability that a variant is different from the reference for that position to be reported.

#### Variant filters

Below the significance settings, there are filters that can be useful for removing false positives:

- **Require presence in both forward and reverse 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 (see [Nguyen et al., 2011] for a recent study with Illumina data). This can easily lead to false positive variant calls, and by checking this filter, the minimum ratio between forward and reverse reads supporting the variant should be at least 0.05. In this way, systematic sequencing errors of this kind can be eliminated. The forward/reverse reads balance is also reported for each variant in the result (see section 27.11).
- **Ignore variants in non-specific regions**. Variants in regions covered by one or more non-specific reads are ignored.
- **Filter 454/Ion homopolymer indels**. The 454 and Ion Torrent/Proton sequencing platforms exhibit weaknesses when determining the correct number of the same kind of nucleotides in a homopolymer region (e.g. AAA). This leads to a high false positive rate for calling InDels in these regions. This filter is very basic: it removes all indels that are found within or just next to a homopolymer region. A homopolymer region is defined as at least two consecutive identical bases in the reference.
- **Required Variant Count**. This option is the threshold for the number of reads that display a variant at a given position and is based on absolute counts. If the count required is set to 3, it means that even though the required percentage of the reads has a variant base, it will still not be reported if there are less than 3 reads supporting the variant.

### 31.2.7 Setting ploidy and genetic code

Clicking **Next** offers options for setting ploidy and genetic code (see figure 31.12):

- **Maximum expected alleles**. This is the ploidy of your organism (or actually "the maximum expected number of alleles"). If set to 1, only homozygous alleles are reported even if another allele is present as well. For cancer samples, which often have a lot of genome duplications, we recommend a setting of 3. For polyploid organisms like plants, a setting of 4 should be used.
- **Genetic code**. For the table report, the variant's effect on the protein level is calculated, and the translation table specified here is used. When reporting the variant as a track, this setting has no effect, since the amino acid consequences are calculated separately (see section 27.15.1).

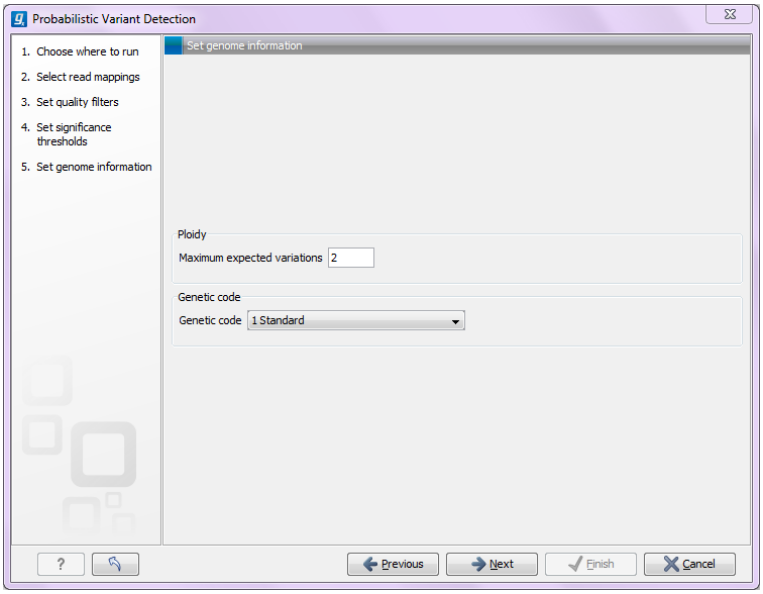



Figure 31.12: *Ploidy and genetic code.*

### 31.2.8 Reporting the variants found

When you click **Next**, you will be able to specify how the variants should be reported (see figure 31.13).

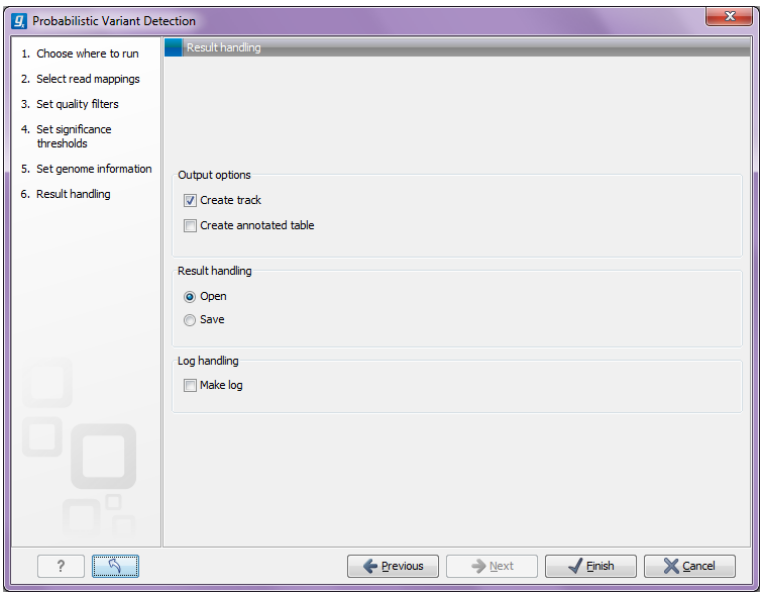



Figure 31.13: *Output options.*

- **Create track**. This will create a variant track that can be further annotated (functional consequences, annotation overlap etc) and used for comparative analysis and visualization(see section 27.13)..

  Note that the track can be displayed in a table view ( ) as well. See a description of the output in section 27.11.1.

- **Create annotated table**. This will create a table showing all the variants including

information about overlapping annotations and amino acid changes. See a description of the output in section 27.11.2.

## 31.3 ChIP-Seq Analysis (legacy)

The legacy ChIP-Seq analysis functionality was part of the *CLC Genomics Workbench* Epigenomics Analysis tools up until version 7.5 where it was replaced by a new ChIP-Seq analysis tool (the former Peak Shape ChIP-Seq Analysis Plugin).

The ChIP-Seq Analysis (legacy) tool can perform analysis of Chromatin immunoprecipitation sequencing (ChIP-Seq) data based on the information contained in a single sample subjected to immunoprecipitation (ChIP-sample) or by comparing a ChIP-sample to a control sample where the immunoprecipitation step is omitted.

The first step in a ChIP-Seq analysis is to map the reads to a reference (see section 25.1), which maps your reads against one or more specified reference sequences. If both a ChIP- and a control sample are used, these must be mapped separately to produce separate ChIP- and control samples.

A stand-alone read mapping is then used as input to the ChIP-Seq tool, which surveys the pattern in coverage to detect significant peaks. Annotations on the reference in the read mapping are carried through to any subsequent ChIP-Seq analysis results.

Please note that stand-alone read mappings, not track-based mappings, are used as input to the Chip-seq tool. In addition, the read mapping must be carried out using a reference that is a sequence ( ) or sequence list ( ). Track-based ( ) sequences can not be used as references for mappings that will be used for Chip-seq analyses. The reason for this is that sequence and sequence list objects can contain annotations and these are included in a read mapping where the annotated sequence or sequence list reference was used. These annotations are important for the current Chip-seq analysis functionality. Annotations for track-based references are held in separate tracks, and this is not yet supported for Chip-seq analysis.

**Toolbox** | **Legacy Tools ( )** | **ChIP-Seq Analysis (legacy) ( )**

This opens a dialog where you can select one or more mapping results ( )/ ( ) to use as ChIP-samples. Control samples are selected in the next step.

### 31.3.1 Peak finding and false discovery rates

Clicking **Next** will display the dialog shown in figure 31.14.

If the option to include control samples is included, the user must select the appropriate sample to use as control data. If the mapping is based on several reference sequences, the Workbench will automatically match the ChIP-samples and controls based on the length of the reference sequences.

The peak finding algorithm includes the following steps:

- Calculate the null distribution of background sequencing signal
- Scan the mappings to identify candidate peaks with a higher read count than expected from the null distribution

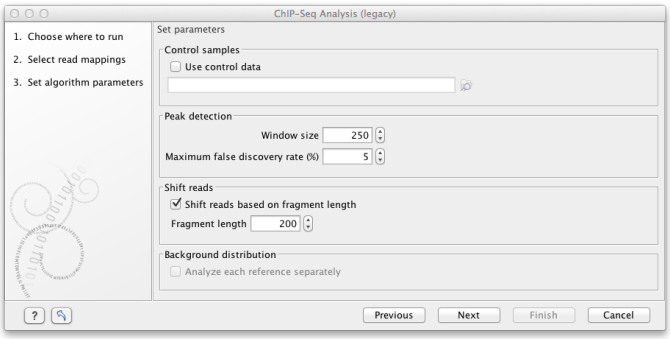



Figure 31.14: *Peak finding and false discovery rates.*

- Merge overlapping candidate peaks
- Refine the set of candidate peaks based on the count and the spatial distribution of reads of forward and reverse orientation within the peaks

The estimation of the null distribution of coverage and the calculation of the false discovery rates are based on the **Window size** and **Maximum false discovery rate (%)** parameters. The **Window size** specifies the width of the window that is used to count reads both when the null distribution is estimated and for the subsequent scanning for candidate peaks.

The **Maximum false discovery rate** specifies the maximum proportion of false positive peaks that you are willing to accept among your called peaks. A value of 10 % means that you are willing to accept that 10 % of the peaks called are expected to be false discoveries.

To estimate the false discovery rate (FDR) we use the method of [Ji et al., 2008] (see also Supplementary materials of the paper).

In the case where only a ChIP-sample is used, a negative binomial distribution is fitted to the counts from low coverage regions. This distribution is used as a null distribution to obtain the numbers of windows with a particular count of reads that you would expect in the absence of significant binding. By comparing the number of windows with a specific count you expect to see under the null distribution and the number you actually see in your data, you can calculate a false discovery rate for a given read count for a given window size as: 'fraction of windows with read count expected under the null distribution'/'fraction of windows with read count observed'.

In the case where both a ChIP- and a control sample are used, a sampling ratio between the samples is first estimated, using only windows in which the total numbers of reads (that is, the sum of those in the sample and those in the control) is small. The sampling ratio is estimated as the ratio of the cumulated sample read counts ($c^{sample} = \sum_i k_i^{sample}$) to cumulated control read counts ($c^{control} = \sum_i k_i^{control}$) in these windows. The sampling ratio is used to estimate the proportion of the reads that are expected to be ChIP-sample reads under the null distribution, as $p_0 = c^{sample}/(c^{sample} + c^{control})$. For a given total read count, *n*, of a window, the numbers of reads expected in the ChIP-sample under the null distribution can then be estimated from the binomial distribution with parameters *n* and $p_0$. By comparing the expected and observed numbers, a false discovery rate can then be calculated. Note, that when a control sample is used different null-distributions are estimated for different total read counts, *n*.

In both cases, the user can specify whether the null distribution should be estimated separately for each reference sequence by checking the option **Analyze each reference separately**.

### 31.3.2 Read shifting

Because the ChIP-seq experimental protocol selects for sequencing input fragments that are centered around a DNA-protein binding site it is expected that true peaks will exhibit a signature distribution where forward reads are found *upstream* of the binding site and reverse reads are found *downstream* of the binding site leading to reduced coverage at the exact binding site. For this reason, the algorithm allows shifting forward reads towards the 3' end and reverse reads towards the 5' end in order to generate a much more discernible peak around the putative binding site prior to the peak detection step. This is done by checking the **Shift reads based on fragment length** box. To shift the reads you also need to input the expected length of the sequencing input fragments by setting the **Fragment length** parameter, this is the size of the fragment isolated from gel (L in the illustration below).

The illustration below shows a peak where the forward reads are in one window and the reverse reads fall in another window (window 1 and 3).

```
------------------------------------------------------- reference
        ----------------------|------------------         (actual sequenced fragment length = L bp)
        ---->                                             reads
         ---->                                            reads
                                          <---            reads
                                        <---              reads.

|--------------------|--------------------|-------------  window size W
         1                      2                  3
```

If the reads are not shifted, the algorithm will count 2 reads in window 1 and 3. But if the forward reads are shifted $*L/2$ to the right and reverse reads are shifted $L/2$ to left, the algorithm will find 4 reads in window 2 as shown below:

```
------------------------------------------------------- reference
        ----------------------|------------------         (actual sequenced fragment length = L basepairs)
                             ---->                          reads
                              ---->                         reads
                          <---                              reads
                         <---                               reads

|--------------------|--------------------|-------------  window size W
         1                      2                  3
```

After shifting the reads, the number of reads that fall within a peak region is increased and consequently the reads will be more concentrated into fewer windows, which improves the accuracy of the peak detection. So the reported number of reads for a peak-region will be higher than in the original read mapping.

The following peak refinement step, the reporting of the peak and the visualization will use the original position of the reads, so the shifting is only a virtual shift performed as part of the peak detection.

### 31.3.3 Peak refinement

Clicking **Next** will display the dialog shown in figure 31.15.

This dialog presents the parameters and options that can be used to refine the set of candidate peaks discovered when scanning the read mapping. All three refinement options again utilize the fact that coverage around a true DNA-protein binding site is expected to exhibit a signature distribution where forward reads are found upstream of the binding site and reverse reads

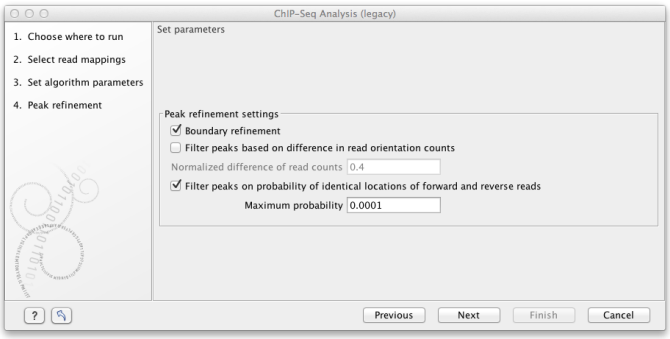



Figure 31.15: *Peak refinement settings.*

are found downstream of the binding site. Peak refinement can be performed both with- and without a control sample but the algorithm only uses information contained in the reads from the ChIP-samples, not the control samples.

If the **Boundary refinement** option is checked, the algorithm will estimate the position of the DNA-protein binding interaction and place the resulting annotations on this region, rather than on the region where a peak in coverage is found. A center of sequencing intensity is defined for all forward reads as the median value of the center points of all forward reads and likewise for all reverse reads. The "refined peak" is thus defined as the region between these two points.

One of the advantages of including this boundary refinement is that shorter regions can be given as input to subsequent pattern discovery analysis.

By checking the **Filter peaks based on difference in read orientation counts** the algorithm will calculate the normalized difference in the number of forward and reverse reads within a peak as

$$\frac{|\text{count forward reads - count reverse reads}|}{\text{count forward reads + count reverse reads}}$$

The desired maximum value of this parameter can be set in the **Normalized difference of read counts** field and any candidate peak with a value above this will then be dismissed. Setting a low value will ensure that peaks are only called if there is a well balanced number of forward and reverse reads.

As an example if you have 15 forward reads and 5 reverse reads, you will end up with a value of 0.5. With the default limit set to 0.4, a peak like that would be excluded.

By checking the **Filter peaks based on spatial distribution of read orientation** the algorithm will evaluate how clearly separated the location of forward and reverse reads are within a peak. This is done via the Wilcoxon rank-sum test (see `http://en.wikipedia.org/wiki/Mann-Whitney-Wilcoxon_test`). The null hypothesis here is that the positions of forward and reverse reads within a peak are drawn from the same distribution i.e. that their locations are not significantly different and the alternative hypothesis is that the forward reads have a sum of ranked positions that is shifted to lower positions than the reverse reads. Peaks will be dismissed if the probability of the null hypothesis exceeds the value set in the **Maximum probability** field.

Setting a low **Maximum probability** will ensure that peaks are only called if there is a clear signature distribution where forward reads are found upstream of reverse reads within the peak.

A general comment about peak filtering is that the relevant statistics are all reported in the peak table that the algorithm outputs. If it is desirable to explore a large set of candidate peaks

it is recommended to use no or relatively loose filtering criteria and then use the advanced table filtering options to explore the effect of the different parameters (see section 8.3). It may be desirable to omit the addition of annotations in this exploratory analysis and rely on the information in the table instead. Once a desired set of parameters is found, the algorithm can be rerun using these as filtering criteria to add annotations to the reference sequence and to produce a final list of peaks.

### 31.3.4 Reporting the results

When you click **Next**, you will be able to specify how the results should be reported (see figure 31.16).

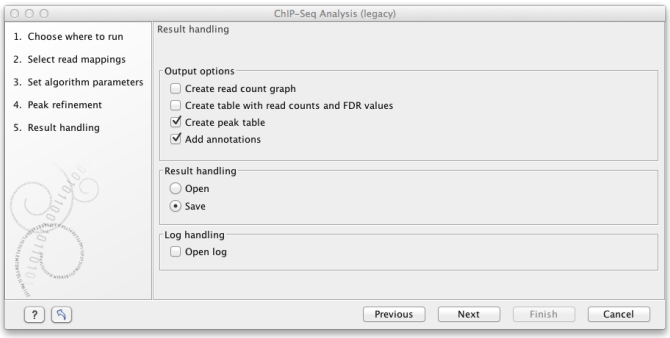



Figure 31.16: *Output options.*

The different output options are described in detail below. Note, that it is not possible to output a graph and table of read counts in the case where a control sample is used. These options are therefore disabled in this case.

#### Graph and table of background distribution and false discovery statistics

An example of a FDR graph based on a single ChIP-sample is shown in figure 31.17.

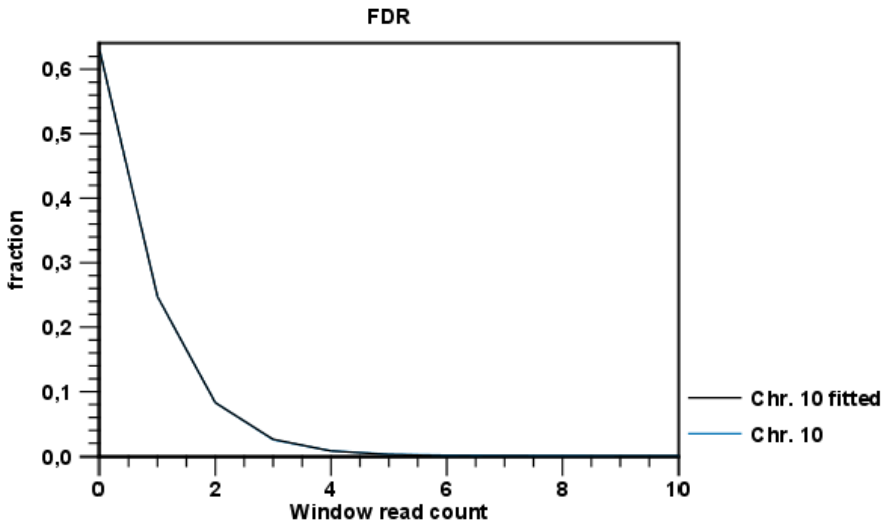


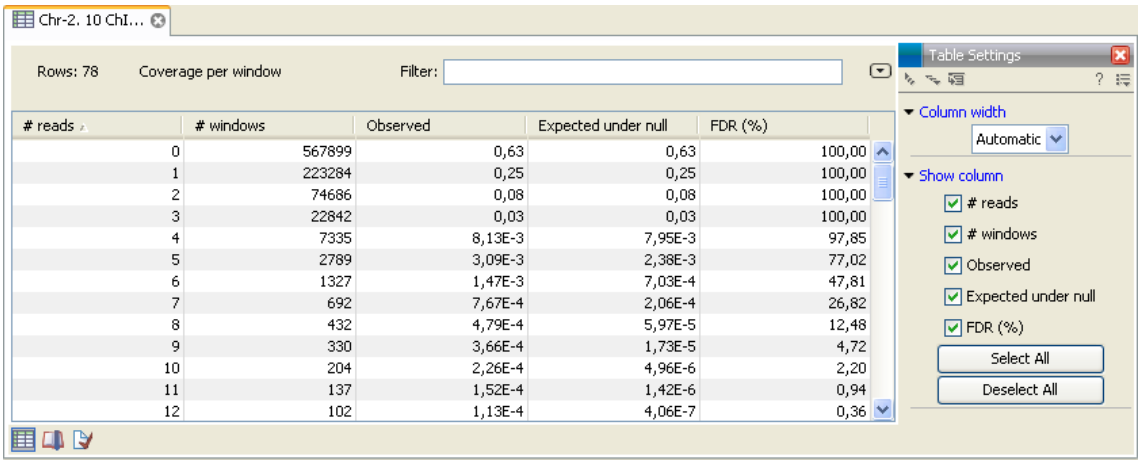

Figure 31.17: *FDR graph.*

The graph shows the estimated background distribution of read counts in discrete windows and the observed counts and can thus be used to inspect how well the estimated distribution fits the observed pattern of coverage.

The FDR table displays the observed and expected fraction of windows with a given read count and also shows the rate of false discovery related to a given level of coverage within a window:

- **# reads** - the number of reads within a window.
- **# windows** - the number of windows with the given read count. A window of a fixed width is slid across the sequence. For every window position the number of reads in that window is recorded and stored as the read count. After this, the windows are counted based on their recorded read counts. # windows of read count x is thus the number of windows that were found to contain x reads during this process. This is done to establish the background distribution of coverage and to evaluate the fit of the estimated distribution.
- **Observed** - the observed faction of windows with the given read count.
- **Expected under null** - the expected fraction of windows with a given read count, under the null distribution.
- **FDR %** - the false discovery rate which is the fraction of the peaks with the given read count that can be expected to be false positives.

An example is shown in figure 31.18.




| # reads | # windows | Observed | Expected under null | FDR (%) |
|---|---|---|---|---|
| 0 | 567899 | 0,63 | 0,63 | 100,00 |
| 1 | 223284 | 0,25 | 0,25 | 100,00 |
| 2 | 74686 | 0,08 | 0,08 | 100,00 |
| 3 | 22842 | 0,03 | 0,03 | 100,00 |
| 4 | 7335 | 8,13E-3 | 7,95E-3 | 97,85 |
| 5 | 2789 | 3,09E-3 | 2,38E-3 | 77,02 |
| 6 | 1327 | 1,47E-3 | 7,03E-4 | 47,81 |
| 7 | 692 | 7,67E-4 | 2,06E-4 | 26,82 |
| 8 | 432 | 4,79E-4 | 5,97E-5 | 12,48 |
| 9 | 330 | 3,66E-4 | 1,73E-5 | 4,72 |
| 10 | 204 | 2,26E-4 | 4,96E-6 | 2,20 |
| 11 | 137 | 1,52E-4 | 1,42E-6 | 0,94 |
| 12 | 102 | 1,13E-4 | 4,06E-7 | 0,36 |

Figure 31.18: *FDR table.*

From this table you can see that less than 5% of the called peaks with 9 reads can be expected to be false discoveries and for peaks with 11 reads the FDR is less than 1%.

### Peak table and annotations

The main result is the table showing the peaks and the annotations added to the reference sequence.

An example of a peak table is shown in figure 31.19.

The table includes information about each peak that has been found:

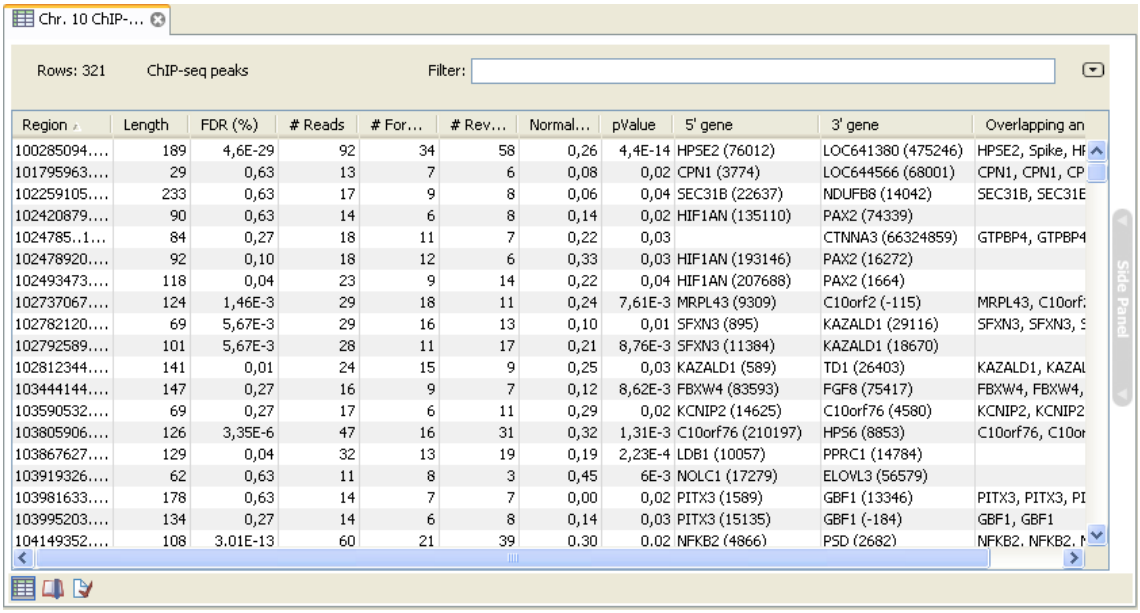



Figure 31.19: *ChIP sequencing peak table.*

- **Name**. If the mapping was based on more than one reference sequence, the name of the reference sequence in question will be shown here.
- **Region**. The position of the peak. To find that position in the ChIP-sample mapping, you can make a split view of the table and the mapping of the table and the mapping (see `http://www.clcsupport.com/clcgenomicsworkbench/current/index.php?manual=Arrange_views_in_View_Area.html`). You will then be able to browse through the peaks by clicking in the table. This will cause the view to jump to the position of the peak region.
- **Length**. The length of the peak.
- **FDR (%)**. The false discovery rate for the peak (learn more in section 31.3.1).
- **# Reads**. The total number of reads covering the peak region. Note that the reported number of reads for a peak-region will be a higher than in the original read mapping. This is a result of shifting forward reads towards the 3' end and reverse reads towards the 5' by the expected length of the sequencing input fragments. See above section 31.3.2
- **# Forward reads**. The number of forward reads covering the peak region.
- **# Reverse reads**. The number of reverse reads covering the peak region. The normalized difference in the count of forward-reverse reads is calculated based on these numbers (see figure 31.15).
- **Normalized difference**. See section 31.3.3.
- **P-value**. The p-value is for the Wilcoxon rank sum test for the equality of location of forward and reverse reads in a peak. See section 31.3.3.
- **Max forward coverage**. The refined region described in section 31.3.3 is calculated based on the maximum coverage of forward and reverse reads.
- **Max reverse coverage**. See previous.

- **Refined region**. The refined region.
- **Refined region length**. The length of the refined region.
- **5' gene**. The nearest gene upstream, based on the start position of the gene. The number in brackets is the distance from the peak to the gene start position.
- **3' gene**. The nearest gene downstream, based on the start position of the gene. The number in brackets is the distance from the peak to the gene start position.
- **Overlapping annotations**. Displays any annotations present on the reference sequence that overlap the peak.

Note that if you make a split view of the table and the mapping of the table and the mapping, you will be able to browse through the peaks by clicking in the table. This will cause the view to jump to the position of the peak.

An example of a peak is shown in figure 31.20.

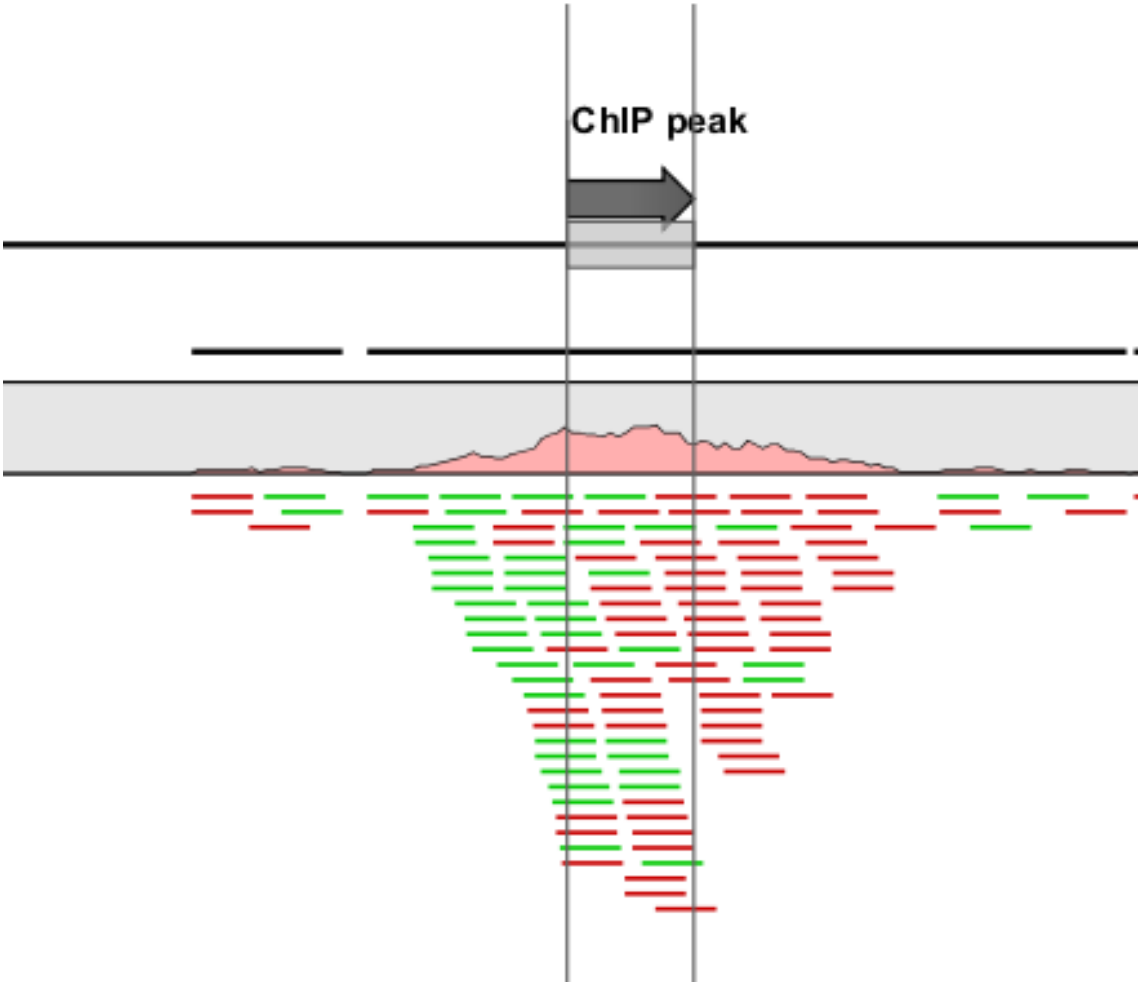



Figure 31.20: *Inspecting an annotated peak. The green lines represent forward reads and the red lines represent reverse reads.*

If you want to extract the sequence of all the peak regions to a list, you can use the **Extract Annotations** tool to extract all annotations of the type "Binding site".

# Part V

# Appendix

# Appendix A

# Comparison of workbenches

Below we list a number of functionalities that differ between CLC Workbenches and the CLC Sequence Viewer:

- CLC Sequence Viewer (■)
- CLC Main Workbench (■)
- CLC Genomics Workbench (■)

| **Data handling** | Viewer | Main | Genomics |
|---|---|---|---|
| Add multiple locations to Navigation Area | | ■ | ■ |
| Share data on network drive | | ■ | ■ |
| Search all your data | | ■ | ■ |

| **Assembly of sequencing data** | Viewer | Main | Genomics |
|---|---|---|---|
| Advanced contig assembly | | ■ | ■ |
| Importing and viewing trace data | | ■ | ■ |
| Trim sequences | | ■ | ■ |
| Assemble without use of reference sequence | | ■ | ■ |
| Map to reference sequence | | ■ | ■ |
| Assemble to existing contig | | ■ | ■ |
| Viewing and edit contigs | | ■ | ■ |
| Tabular view of an assembled contig (easy data overview) | | ■ | ■ |
| Secondary peak calling | | ■ | ■ |
| Multiplexing based on barcode or name | | ■ | ■ |

| **Next-generation Sequencing Data Analysis** | Viewer | Main | Genomics |
|---|---|---|---|
| Import of 454, Illumina Genome Analyzer, SOLiD and Helicos data | | | ■ |
| Reference assembly of human-size genomes | | | ■ |
| De novo assembly | | | ■ |
| SNP/DIP detection | | | ■ |
| Graphical display of large contigs | | | ■ |
| Support for mixed-data assembly | | | ■ |
| Paired data support | | | ■ |
| RNA-Seq analysis | | | ■ |
| Expression profiling by tags | | | ■ |
| ChIP-Seq analysis | | | ■ |

| **Expression Analysis** | Viewer | Main | Genomics |
|---|---|---|---|
| Import of Illumina BeadChip, Affymetrix, GEO data | | ■ | ■ |
| Import of Gene Ontology annotation files | | ■ | ■ |
| Import of Custom expression data table and Custom annotation files | | ■ | ■ |
| Multigroup comparisons | | ■ | ■ |
| Advanced plots: scatter plot, volcano plot, box plot and MA plot | | ■ | ■ |
| Hierarchical clustering | | ■ | ■ |
| Statistical analysis on count-based and gaussian data | | ■ | ■ |
| Annotation tests | | ■ | ■ |
| Principal component analysis (PCA) | | ■ | ■ |
| Hierarchical clustering and heat maps | | ■ | ■ |
| Analysis of RNA-Seq/Tag profiling samples | | ■ | ■ |

| **Molecular cloning** | Viewer | Main | Genomics |
|---|---|---|---|
| Advanced molecular cloning | | ■ | ■ |
| Graphical display of in silico cloning | | ■ | ■ |
| Advanced sequence manipulation | | ■ | ■ |

| **Database searches** | Viewer | Main | Genomics |
|---|---|---|---|
| GenBank Entrez searches | ■ | ■ | ■ |
| UniProt searches (Swiss-Prot/TrEMBL) | | ■ | ■ |
| Web-based sequence search using BLAST | | ■ | ■ |
| BLAST on local database | | ■ | ■ |
| Creation of local BLAST database | | ■ | ■ |
| PubMed lookup | | ■ | ■ |
| Web-based lookup of sequence data | | ■ | ■ |
| Search for structures (at NCBI) | | ■ | ■ |

| **General sequence analyses** | Viewer | Main | Genomics |
|---|---|---|---|
| Linear sequence view | ■ | ■ | ■ |
| Circular sequence view | ■ | ■ | ■ |
| Text based sequence view | | ■ | ■ |
| Editing sequences | ■ | ■ | ■ |
| Adding and editing sequence annotations | | ■ | ■ |
| Advanced annotation table | | ■ | ■ |
| Join multiple sequences into one | ■ | ■ | ■ |
| Sequence statistics | ■ | ■ | ■ |
| Shuffle sequence | ■ | ■ | ■ |
| Local complexity region analyses | | ■ | ■ |
| Advanced protein statistics | | ■ | ■ |
| Comprehensive protein characteristics report | | ■ | ■ |

| **Nucleotide analyses** | Viewer | Main | Genomics |
|---|---|---|---|
| Basic gene finding | ■ | ■ | ■ |
| Reverse complement without loss of annotation | ■ | ■ | ■ |
| Restriction site analysis | ■ | ■ | ■ |
| Advanced interactive restriction site analysis | | ■ | ■ |
| Translation of sequences from DNA to proteins | ■ | ■ | ■ |
| Interactive translations of sequences and alignments | | ■ | ■ |
| G/C content analyses and graphs | | ■ | ■ |

| **Protein analyses** | Viewer | Main | Genomics |
|---|---|---|---|
| 3D molecule view | | ■ | ■ |
| Hydrophobicity analyses | | ■ | ■ |
| Antigenicity analysis | | ■ | ■ |
| Protein charge analysis | | ■ | ■ |
| Reverse translation from protein to DNA | | ■ | ■ |
| Proteolytic cleavage detection | | ■ | ■ |
| Prediction of signal peptides (SignalP) | | ■ | ■ |
| Transmembrane helix prediction (TMHMM) | | ■ | ■ |
| Secondary protein structure prediction | | ■ | ■ |
| PFAM domain search | | ■ | ■ |

| **Sequence alignment** | Viewer | Main | Genomics |
|---|---|---|---|
| Multiple sequence alignments (Two algorithms) | ■ | ■ | ■ |
| Advanced re-alignment and fix-point alignment options | | ■ | ■ |
| Advanced alignment editing options | ■ | ■ | ■ |
| Join multiple alignments into one | | ■ | ■ |
| Consensus sequence determination and management | ■ | ■ | ■ |
| Conservation score along sequences | ■ | ■ | ■ |
| Sequence logo graphs along alignments | | ■ | ■ |
| Gap fraction graphs | | ■ | ■ |
| Copy annotations between sequences in alignments | | ■ | ■ |
| Pairwise comparison | ■ | ■ | ■ |

| **RNA secondary structure** | Viewer | Main | Genomics |
|---|---|---|---|
| Advanced prediction of RNA secondary structure | | ■ | ■ |
| Integrated use of base pairing constraints | | ■ | ■ |
| Graphical view and editing of secondary structure | | ■ | ■ |
| Info about energy contributions of structure elements | | ■ | ■ |
| Prediction of multiple sub-optimal structures | | ■ | ■ |
| Evaluate structure hypothesis | | ■ | ■ |
| Structure scanning | | ■ | ■ |
| Partition function | | ■ | ■ |

| **Dot plots** | Viewer | Main | Genomics |
|---|---|---|---|
| Dot plot based analyses | | ■ | ■ |

| **Phylogenetic trees** | Viewer | Main | Genomics |
|---|---|---|---|
| Neighbor-joining and UPGMA phylogenies | ■ | ■ | ■ |
| Maximum likelihood phylogeny of nucleotides | | ■ | ■ |

| **Pattern discovery** | Viewer | Main | Genomics |
|---|---|---|---|
| Search for sequence match | ■ | ■ | ■ |
| Motif search for basic patterns | | ■ | ■ |
| Motif search with regular expressions | | ■ | ■ |
| Motif search with ProSite patterns | | ■ | ■ |
| Pattern discovery | | ■ | ■ |

| **Primer design** | Viewer | Main | Genomics |
|---|---|---|---|
| Advanced primer design tools | | ■ | ■ |
| Detailed primer and probe parameters | | ■ | ■ |
| Graphical display of primers | | ■ | ■ |
| Generation of primer design output | | ■ | ■ |
| Support for Standard PCR | | ■ | ■ |
| Support for Nested PCR | | ■ | ■ |
| Support for TaqMan PCR | | ■ | ■ |
| Support for Sequencing primers | | ■ | ■ |
| Alignment based primer design | | ■ | ■ |
| Alignment based TaqMan probe design | | ■ | ■ |
| Match primer with sequence | | ■ | ■ |
| Ordering of primers | | ■ | ■ |
| Advanced analysis of primer properties | | ■ | ■ |

| **Molecular cloning** | Viewer | Main | Genomics |
|---|---|---|---|
| Advanced molecular cloning | | ■ | ■ |
| Graphical display of in silico cloning | | ■ | ■ |
| Advanced sequence manipulation | | ■ | ■ |

| **Virtual gel view** | Viewer | Main | Genomics |
|---|---|---|---|
| Fully integrated virtual 1D DNA gel simulator | | ■ | ■ |

# Appendix B

# Use of multi-core computers

Many tools in CLC Workbenches and Servers can make use of multi-core CPUs. This does not necessarily mean that all available CPU cores are used throughout the analysis. It means that these tools benefit from running on computers with multiple CPU cores.

Tools available differ between CLC Workbenches. In the table, the availability of these tools in different CLC Workbench Toolbox menus is indicated with an X.

| **Use of multi-core computers** | Genomics | Drug Discovery | Cancer Research |
|---|---|---|---|
| Basic Variant Detection | X | | X |
| BLAST (will not scale well on many cores) | X | | |
| Create Alignment | X | X | X |
| Create Detailed Mapping Report | X | | X |
| Create Sequencing QC Report (will not scale well on more than four cores) | X | | |
| De Novo Assembly | X | | |
| Dock Ligands | | X | |
| Download Reference Genome Data | X | | |
| Extract and Count | X | | X |
| Fixed Ploidy Variant Detection | X | | X |
| Import Molecules from SMILES or 2D | | X | |
| K-mer Based Tree Construction | X | | |
| Large Gap Read Mapper (currently in beta) | X | | |
| Local Realignment | X | | X |
| Low Frequency Variant Detection | X | | X |
| Map Reads to Contigs | X | | |
| Map Reads to Reference | X | | X |
| Maximum Likelihood Phylogeny | X | | |
| Model Testing | X | | |
| Probabilistic Variant Detection (legacy) | X | | X |
| QC for Sequencing Reads (will not scale well on more than four cores) | | | X |
| Quality-based Variant Detection (legacy) | X | | X |
| RNA-Seq Analysis | X | | X |
| Screen Ligands | | X | |
| Trim Sequences | X | | X |

Please note that a static license has a limitation on the maximum number of cores, see section 1.3.1.

# Appendix C

# 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.
  - Outside
  - Inside
- **Tick lines at.** Choosing Major ticks will show a grid behind the graph.
  - None
  - Major ticks
- **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.
- **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 **Lines and plots** below contains the following settings:

- **Dot type**
  - None
  - Cross
  - Plus
  - Square
  - Diamond
  - Circle
  - Triangle
  - Reverse triangle
  - Dot
- **Dot color.** Allows you to choose between many different colors. Click the color box to select a color.
- **Line width**
  - Thin
  - Medium
  - Wide
- **Line type**
  - None
  - Line
  - Long dash
  - Short dash
- **Line color.** Allows you to choose between many different colors. Click the color box to select a color.

For graphs with multiple data series, you can select which curve the dot and line preferences should apply to. This setting is at the top of the **Side Panel** group.

Note that the graph title and the 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 D

# BLAST databases

Several databases are available at NCBI, which can be selected to narrow down the possible BLAST hits.

## D.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.
- **month.** All new or revised GenBank CDS translations + PDB + SwissProt + PIR + PRF released in the last 30 days..

## D.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.
- **refseq_rna.** mRNA sequences from NCBI Reference Sequence Project.
- **refseq_genomic.** Genomic 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.
- **month.** All new or revised GenBank+EMBL+DDBJ+PDB sequences released in the last 30 days.
- **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.
- **wgs.** Assemblies of Whole Genome Shotgun sequences.
- **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.

## D.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: http://www.clcbio.com/wbsettings/NCBI_BlastNucleotideDatabazip
- Protein databases: http://www.clcbio.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 E

# 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 F

# 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: http://www.clcbio.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.

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 G

# 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 `http://www.clcbio.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 G.1).

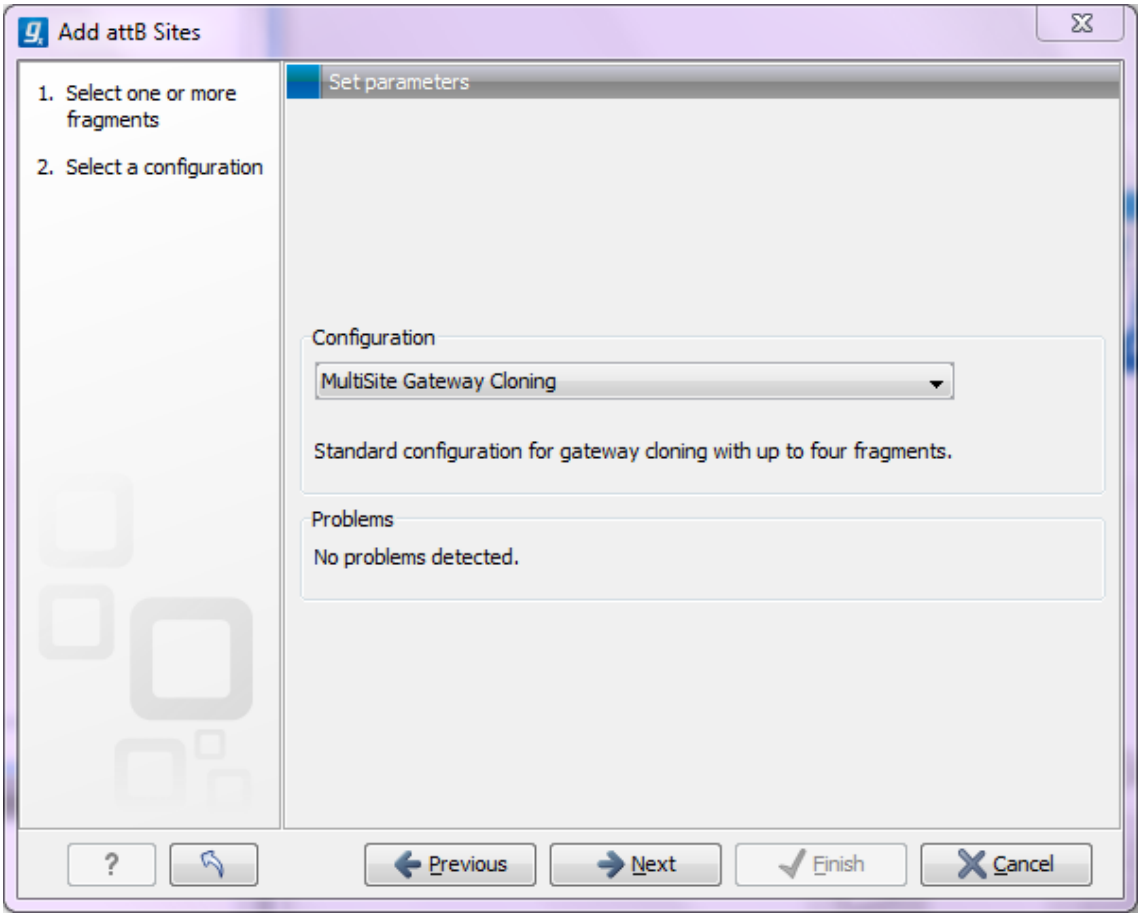



Figure G.1: *Selecting between different gateway cloning configurations.*

# Appendix H

# 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 I

# 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 J

# Formats for import and export

## J.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 and trees.

### J.1.1 Sequence data formats

Note that high-throughput sequencing data formats from Illumina, SOLiD, 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 |
| 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 |
| GCG sequence | .gcg | X | | Rich information incl. annotations |
| GenBank | .gbk/.gb/.gp | X | X | Rich information incl. annotations |
| Gene Construction Kit | .gck | X | | |
| Lasergene | .pro/.seq | X | | |
| Nexus | .nxs/.nexus | X | X | |
| Phred | .phd | X | | Including chromatograms |
| PIR (NBRF) | .pir | 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 1.7.1).

When exporting in fasta format, it is possible to remove sequence ends covered by annotations of type "Trim" (read more in section 18.2).

### J.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 6.3.9 |
| 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 6.3.9 |
| Mapping coverage | .tsv | | X | Detailed per-base info on coverage (see below) |

**Special 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 read 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.

### J.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 | |

### J.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 | |

### J.1.5 Expression data formats

Read about technical details of these data formats in section L.

| 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 |

### J.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 sequence or sequence list using functionality provided by the Annotate with GFF plugin (http://www.clcbio.com/clc-plugin/annotate-sequence-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 note below |
| GFF | .gff | X | X | To import as annotation track, see section 6.2. To annotated sequence or sequence list, see plugin: http://www.clcbio.com/clc-plugin/annotate-sequence-with-gff-file/ |
| 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. |
| COSMIC variation database | .tsv | X | | Special format for COSMIC data |
| BED | .bed | X | | See section 6.2 |
| Wiggle | .wig | X | | See section 6.2 |
| UCSC variant database table dump | .txt | X | | See section 6.2 |
| Complete genomics master var files | masterVar | X | | Complete genomics variant data format |

**Special note on VCF export**

For VCF export, counts from the variant track are put in CLCAD2 tags and coverage in DP tags. 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.

When exporting VCF files, there are three options:

**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.

**Enforce diploid export** The *CLC Genomics Workbench* option will generate 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 are 1/1). This is to ensure compatibility of the exported VCF file with programs for downstream variant analysis that expect strictly diploid genomes. The user can specify that the Enforce diploid option is only applied to certain chromosomes, while others may be reported as haploid. If you export a variant track that has been filtered, there can be situations where there is only one heterozygous variant at a given position. In this case, the *CLC Genomics Workbench* will use a "." to denote an unknown genotype, so the GT field will be "1/.".

*Note:* the "Enforce diploid" option does NOT enforce diploidy for polyploid variant loci. Regardless of this setting, all variant alleles reported during variant calling are included in the exported VCF file.

It is important to note that this **Enforce diploid export** option will create a diploid format of the VCF file, but it is not able to recover any inconsistencies in the variant track used as input. If the variant track has three variants at a given position, three genotypes will be output. Or if the variant track has two variants at the same position that both postulate to be homozygous, they will be output as two heterozygous variants. When exporting data created by the variant callers of *CLC Genomics Workbench*, this is usually not a problem, but when applying this diploid scheme to data that has been imported into the *CLC Genomics Workbench* from other sources, the data can be inconsistent with a diploid model.

**Exceptions** 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.

#### Special note on former VCF export

In CLC Genomics Workbench 6.5 instead of the CLCAD2, the CLCAD field had been reported. The difference between CLCAD and CLCAD2 is that the former is following the order in the GT (genotype) field in VCF, while the latter is following the order of the REF and ALT fields in VCF in is therefore more in line with the AD field reported from GATK and other sources.

**Special notes on VCF import Note!** Please also see section 6.2.

The import process for VCF files into the CLC Genomics Workbench currently work as follows:

1. For VCF rows that are reporting the reference base no variants are imported
2. In cases where GT = 0/0, GT=./., GT=0/. or GT=./0 no variants are imported at all
3. 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
4. In cases where GT = X/X and X is not zero, in Genomics Workbench 6.5 this will result in two independent variants. In version 6.5.1 they will be reported as a single homozygous variant
5. In cases where GT = X/Y and X and Y are different but either one may be zero, two independent variants are created

Please note that some replacements cannot be interpreted in versions that are older than CLC Genomics Workbench 7.0; such replacements will therefore not be imported in previous versions of *CLC Genomics Workbench*.

An example of these types of replacements are the following:

chr2 32843292 . TTTA T,TTT 100 PASS DP=44

Due to the VCF interpretation, the initial T base is removed from all alternatives.

In version 6.5.x, only the reference variant (TTA -> TTA) and the first deletion in ALT (TTA -> -) will be imported. The replacements TTA -> TT will not be imported.

In version 7.0 and later versions, all variants will be imported, but the replacements TTA -> TT will be imported as one deletion A -> - .

To get a variant count as part of your imported variant, one of the following VCF tags have to be present in your VCF file: CLCAD2, AD, or AO.

The import of CLCAD2/AD/AO tags are prioritized in the following order:

1. CLCAD2
2. AD
3. AO

If the CLCAD2 is missing, and only AD is present, then AD is used in the "count" column.

The consequence of this, if the file for example has CLCAD2:AD, and in a sample for three possible variants the values are 2,3,4:5,6,7, then the CLCAD2 tag will be imported as count, so each of the three variants will have just one count value (2, 3, and 4 respectively). At the same time, the AD tag will be imported as an annotation so all of the three variants will have "5,6,7" under the AD column, like for any unknown format tag.

**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.

### J.1.7 Other formats

| File type | Suffix | Import | Export | Description |
|---|---|---|---|---|
| CLC | .clc | X | X | Rich format including all information |
| mmCIF | .cif | X | | 3D structure |
| PDB | .pdb | X | | 3D structure |
| RNA structures | .ct, .col, .rnaml/.xml | x | | Secondary structure for RNA |

### J.1.8 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 J.1.5 **Expression data formats** for special cases of table imports.

### J.1.9 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.

## J.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 6.5 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 K

# 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.

**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.

  This scoring system is based on a recommendation provided in the the SAM FAQ:

  http://sourceforge.net/apps/mediawiki/samtools/index.php?title=SAM_FAQ#How_to_make_my_aligner_work_best_with_samtools.3F

**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.
- The workbench exports color space information in the CS tag.
- The colors of a right mate are incorrect since the colors of a paired read are stored as a single color string.
- For hard clipped sequence reads, the color space is incorrect, since the color space string is not hard clipped.
- SAM files contain sequence quality score and color quality scores. The workbench only have color quality scores and these are stored and exported as sequence quality scores.

## K.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.

| Description of the example | Bits | Flag | Illustration |
|---|---|---|---|
| The first mate of a non-broken paired read | `0x1`, `0x2`, `0x20`, `0x40` | **99** | See Figure K.1 |
| The second mate of a non-broken paired read | `0x1`, `0x2`, `0x10`, `0x80` | **147** | See Figure K.2 |
| A single, forward read (or paired read, where only one mate of the pair is mapped) | No set bits | **0** | see Figure K.3 |
| A single, reversed read (or paired read, where only one mate of the pair is mapped) | `0x10` | **16** | See Figure K.4 |
| The first, forward segment from a broken pair with forward mate | `0x1`, `0x40` | **65** | See Figure K.5 |
| The second, forward segment from broken pair with reversed mate | `0x1`, `0x20`, `0x80` | **161** | See Figure K.6 |
| The first, reversed segment from broken pair with forward mate | `0x1`, `0x10`, `0x40` | **81** | See Figure K.7 |
| The second, reversed segment from broken pair with reversed mate | `0x1`, `0x10`, `0x20`, `0x80` | **177** | See Figure K.8 |

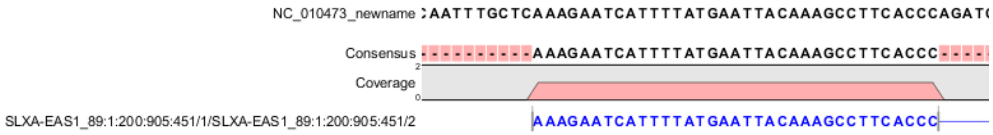



Figure K.1: *The read is paired, both reads are mapped and the mate of this read is reversed*

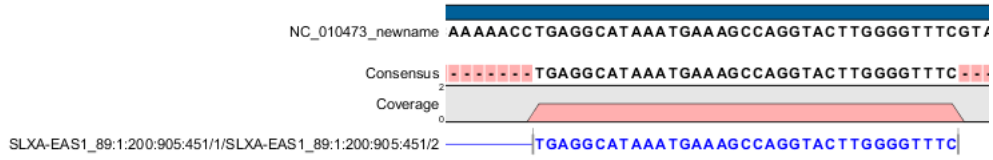



Figure K.2: *The read is paired, both mates are mapped, and this segment is reversed*

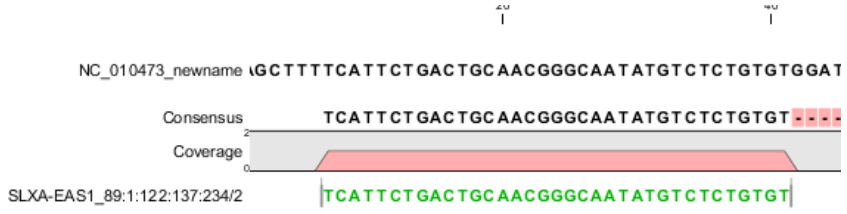



Figure K.3: *A single, forward read, or a paired read where the mate is not mapped*

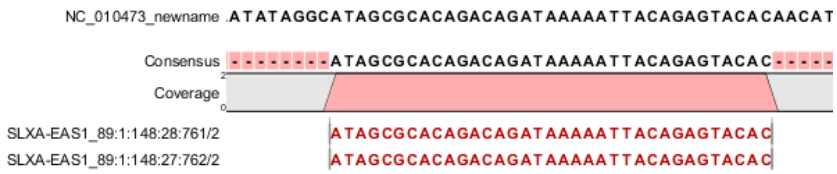



Figure K.4: *The read is a single, reversed read, or a paired read where the mate is not mapped*

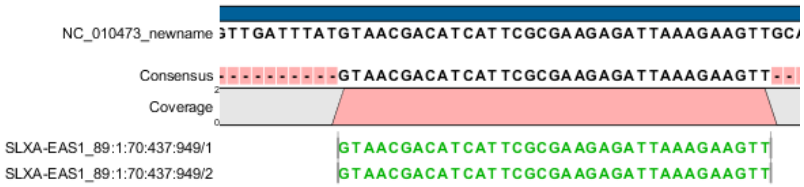



Figure K.5: *These forward reads are paired. They map to the same place, so the pair is broken*

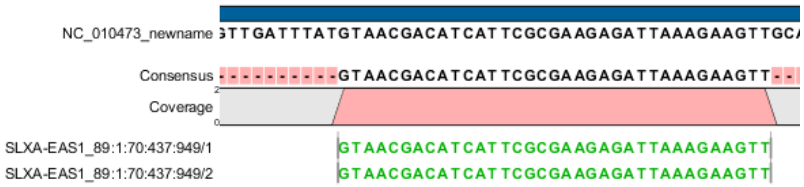



Figure K.6: *Forward read that is part of a broken read where the mate is reversed*

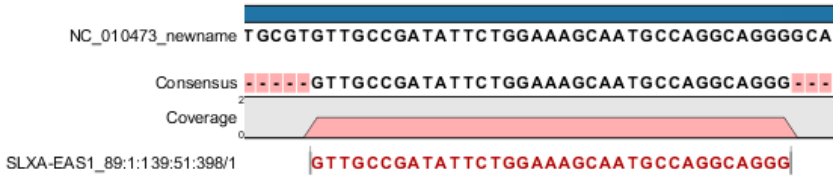



Figure K.7: *Reversed read that is part of a broken pair, where the mate is forward*

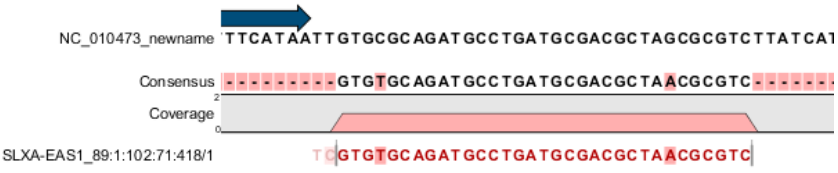



Figure K.8: *Reversed read that is part of a broken pair, where the mate is also reversed.*

# Appendix L

# 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 L.5).

The workbench assumes that expression values are given at the gene level, thus probe-level analysis of e.g. 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 L.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 L.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.

## L.1 GEO (Gene Expression Omnibus)

The GEO (Gene Expression Omnibus) sample and series formats are supported. Figure L.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`,

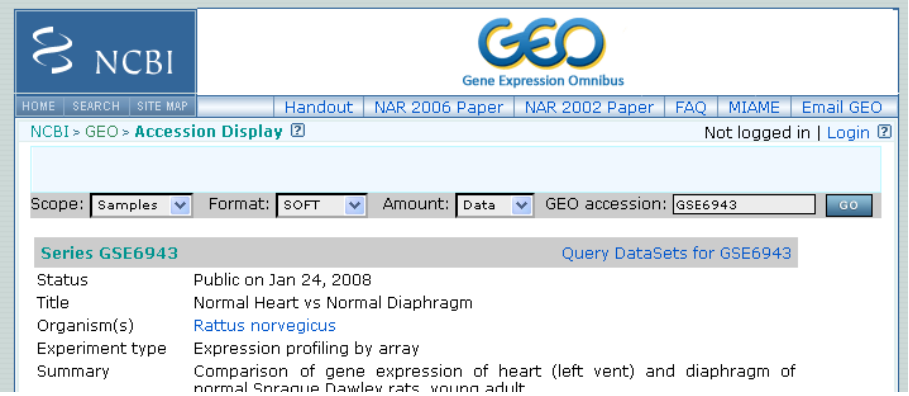



Figure L.1: *Selecting Samples, SOFT and Data before clicking go will give you the format supported by the* **CLC Genomics Workbench**.

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:
http://www.clcbio.com/madata/GEOSampleFilesConcatenated.txt

Below you can find examples of the formatting of the GEO formats.

### L.1.1 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:
http://www.clcbio.com/madata/GEOSampleFileSimple.txt

### L.1.2 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:
http://www.clcbio.com/madata/GEOSampleFileAbsentPresent.txt

### L.1.3 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:
http://www.clcbio.com/madata/GEOSampleFileAbsentPresentCallAndPValue.txt

### L.1.4 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:
http://www.clcbio.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:
http://www.clcbio.com/madata/GEOSampleFileSimpleSequenceTagAndProbe.txt

### L.1.5 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:
http://www.clcbio.com/madata/GEOSeriesFile.txt

## L.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 gene-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 gene 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 L.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.

### L.2.1 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):
http://www.clcbio.com/madata/AffymetrixCHPandPSI.zip

### L.2.2 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:
http://www.clcbio.com/madata/AffymetrixMetrics.txt

### L.2.3 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 28.4.4.

Download a small example annotation file here which includes header information:
http://www.clcbio.com/madata/AffymetrixNetAffxAnnotationFile.csv

## L.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.

### L.3.1 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:
http://www.clcbio.com/madata/IlluminaBeadChipCompact.txt

### L.3.2 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:
http://www.clcbio.com/madata/IlluminaBeadChipExtended.txt

### L.3.3 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:
http://www.clcbio.com/madata/IlluminaBeadStudioWithAnnotations.txt

### L.3.4 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:
http://www.clcbio.com/madata/IlluminaBeadStudioMultipleSamples.txt

This file contains data for 18 samples. Each sample has an expression value (the value in the `AVG_Signal` column), a detection p-value, a bead standard deviation and an average bead number column. The workbench recognizes the 18 samples and their columns.

### L.3.5 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 28.4.4.

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:
`http://www.clcbio.com/madata/IlluminaBeadChipAnnotation.txt`

## L.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*. This can be used to annotate experiments as shown in section 28.4.4. See the complete list including download links at `http://www.geneontology.org/GO.current.annotations.shtml`.

This is an easy way to annotate your experiment with GO categories.

## L.5 Generic expression and annotation data file formats

If you have your expression or annotation data in e.g. 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.

### L.5.1 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 a this example as a file here:
http://www.clcbio.com/madata/CustomExpressionData.txt

### L.5.2 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:
http://www.clcbio.com/madata/SimpleCustomAnnotation.csv
http://www.clcbio.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 M

# 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.

Restart the Workbench to have the changes take effect.

Please note that when updating the Workbench to a new version, this information is not preserved. This means that you should keep this information in a separate place as back-up. (The ability to change the tables is mainly aimed at centrally deployed installations of the Workbench).

# Appendix N

# 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 |
|---|---|---|---|
| Compare Samples within Group (see section 27.14.1) | 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 27.14.4) | 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 | - |
| Fisher Exact Test (see section 27.14.3) | 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 |
|---|---|---|---|
| Filter Against Control Reads (see section 27.14.5) | 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 patient | - |
| Filter against known variants (see section 27.13.1) | 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 27.13.2) | 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 24.5.4) | 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 | - |
| Compare Sample Variant Tracks (see section 27.14.2) | Removes variants called in one sample that are also called (or not called) in another sample for comparison | Sample/normal comparison | Please note that this tool has the same functionality as the Annotate against known variants tool, but should be used for called variants only and not for comparison or filtering using databases |

# Bibliography


[Akmaev and Wang, 2004] Akmaev, V. R. and Wang, C. J. (2004). Correction of sequence-based artifacts in serial analysis of gene expression. *Bioinformatics*, 20(8):1254–1263.

[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 and Gish, 1996] Altschul, S. F. and Gish, W. (1996). Local alignment statistics. *Methods Enzymol*, 266:460–480.

[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.

[Altshuler et al., 2000] Altshuler, D., Pollara, V. J., Cowles, C. R., Etten, W. J. V., Baldwin, J., Linton, L., and Lander, E. S. (2000). An snp map of the human genome generated by reduced representation shotgun sequencing. *Nature*, 407(6803):513–516.

[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.

[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.

[Bendtsen et al., 2004a] Bendtsen, J. D., Jensen, L. J., Blom, N., Heijne, G. V., and Brunak, S. (2004a). Feature-based prediction of non-classical and leaderless protein secretion. *Protein Eng Des Sel*, 17(4):349–356.

[Bendtsen et al., 2005] Bendtsen, J. D., Kiemer, L., Fausbï¿½ll, A., and Brunak, S. (2005). Non-classical protein secretion in bacteria. *BMC Microbiol*, 5:58.

[Bendtsen et al., 2004b] Bendtsen, J. D., Nielsen, H., von Heijne, G., and Brunak, S. (2004b). Improved prediction of signal peptides: SignalP 3.0. *J Mol Biol*, 340(4):783–795.

[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.

[Blobel, 2000] Blobel, G. (2000). Protein targeting (Nobel lecture). *Chembiochem.*, 1:86–102.

[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.

[Brockman et al., 2008] Brockman, W., Alvarez, P., Young, S., Garber, M., Giannoukos, G., Lee, W. L., Russ, C., Lander, E. S., Nusbaum, C., and Jaffe, D. B. (2008). Quality scores and snp detection in sequencing-by-synthesis systems. *Genome Res*, 18(5):763–770.

[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., Ferrï¿$\frac{1}{2}$, 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., Bï¿½tner, 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., Wï¿½nning, 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.

[Hein, 2001] Hein, J. (2001). An algorithm for statistical alignment of sequences related by a binary tree. In *Pacific Symposium on Biocomputing*, page 179.

[Hein et al., 2000] Hein, J., Wiuf, C., Knudsen, B., Mï¿½ller, M. B., and Wibling, G. (2000). Statistical alignment: computational properties, homology testing and goodness-of-fit. *J Mol Biol*, 302(1):265–279.

[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.

[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.

[Hï¿½hl et al., 2007] Hï¿½hl, M., Rigoutsos, I., and Ragan, M. A. (2007). Pattern-based phylogenetic distance estimation and tree reconstruction. *Evolutionary Bioinformatics*, 2:0–0.

[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.

[Ji et al., 2008] Ji, H., Jiang, H., Ma, W., Johnson, D., Myers, R., and Wong, W. (2008). An integrated software system for analyzing ChIP-chip and ChIP-seq data. *Nature Biotechnology*, 26(11):1293–1300.

[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*.

[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.

[Klee and Ellis, 2005] Klee, E. W. and Ellis, L. B. M. (2005). Evaluating eukaryotic secreted protein prediction. *BMC Bioinformatics*, 6:256.

[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.

[Krogh et al., 2001] Krogh, A., Larsson, B., von Heijne, G., and Sonnhammer, E. L. (2001). Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *J Mol Biol*, 305(3):567–580.

[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.

[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., 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.

[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.

[Maeda et al., 2008] Maeda, N., Nishiyori, H., Nakamura, M., Kawazu, C., Murata, M., Sano, H., Hayashida, K., Fukuda, S., Tagami, M., Hasegawa, A., Murakami, K., Schroder, K., Irvine, K., Hume, D., Hayashizaki, Y., Carninci, P., and Suzuki, H. (2008). Development of a dna barcode tagging method for monitoring dynamic changes in gene expression by using an ultra high-throughput sequencer. *Biotechniques*, 45(1):95–97.

[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.

[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.

[Menne et al., 2000] Menne, K. M., Hermjakob, H., and Apweiler, R. (2000). A comparison of signal sequence prediction methods using a test set of signal peptides. *Bioinformatics*, 16(8):741–742.

[Meyer et al., 2007] Meyer, M., Stenzel, U., Myles, S., Prï¿½fer, 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.

[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.

[Nguyen et al., 2011] Nguyen, P., Ma, J., Pei, D., Obert, C., Cheng, C., and Geiger, T. (2011). Identification of errors introduced during high throughput sequencing of the t cell receptor repertoire. *BMC genomics*, 12(1):106.

[Nielsen et al., 1997] Nielsen, H., Engelbrecht, J., Brunak, S., and von Heijne, G. (1997). Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein Eng*, 10(1):1–6.

[Nielsen, 2007] Nielsen, K. L., editor (2007). *Serial Analysis of Gene Expression (SAGE): Methods and Protocols*, volume 387 of *Methods in Molecular Biology*. Humana Press.

[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.

[Reinhardt and Hubbard, 1998] Reinhardt, A. and Hubbard, T. (1998). Using neural networks for prediction of the subcellular location of proteins. *Nucleic Acids Res*, 26(9):2230–2236.

[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 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., Sætrom, P., and Drabløs, 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.

['t Hoen et al., 2008] 't Hoen, P. A. C., Ariyurek, Y., Thygesen, H. H., Vreugdenhil, E., Vossen, R. H. A. M., de Menezes, R. X., Boer, J. M., van Ommen, G.-J. B., and den Dunnen, J. T. (2008). Deep sequencing-based expression analysis shows major advances in robustness, resolution and inter-lab portability over five microarray platforms. *Nucleic Acids Res*, 36(21):e141.

[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.

[von Ahsen et al., 2001] von Ahsen, N., Wittwer, C. T., and Schï¿½tz, 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.

[von Heijne, 1986] von Heijne, G. (1986). A new method for predicting signal sequence cleavage sites. *Nucl. Acids Res.*, 14:4683–4690.

[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.

[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 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.

# Part VI

# Index

# Index


mapping
    extract from selection, 389
3D Molecule Viewer, 245
3D molecule view
    navigate, 250
    rotate, 250
    styles, 252
    zoom, 250
3D structure, 247, 248
454 sequencing data, 884

AB1, file format, 904
Abbreviations
    amino acids, 901
ABI, file format, 904
About CLC Workbenches, 34
Accession number, display, 67
.ace, file format, 910
ACE, file format, 905
Actions Drugdiscovery, 258
Adapter trimmming, 516
Add
    annotations, 193, 885
    sequences to alignment, 448
    sequences to contig, 383
    Structure Prediction Constraints, 490
Adjust selection, 185
Adjust trim, 384
Advanced preferences, 87
Advanced RNA options
    Apply base pairing constraints, 490
    Avoid isolated base pairs, 490, 502
    Coaxial stacking, 490, 502
    GAIL rule, 490, 502
Advanced search, 78
Affymetrix arrays, 916
Affymetrix NetAffx, file format, 906
Affymetrix, file format, 906
Affymetrix, supported file formats, 919
Algorithm
    alignment, 436
Align
    alignments, 439
    sequences, 886
Alignment, *see* Alignments
Alignment Primers
    Degenerate primers, 359, 360
    PCR primers, 358
    Primers with mismatches, 359, 360
    Primers with perfect match, 359, 360
    TaqMan Probes, 358
Alignment-based primer design, 358
Alignments, 436, 886
    add sequences to, 448
    compare, 450
    create, 437
    design primers for, 358
    edit, 446
    fast algorithm, 438
    join, 449
    multiple, Bioinformatics explained, 453
    remove sequences from, 448
    view, 442
    view annotations on, 189
Aliphatic index, 289
.aln, file format, 910
Alphabetical sorting of folders, 65
Ambiguities, reverse translation, 334
Amino acid composition, 291
Amino acids
    abbreviations, 901
    UIPAC codes, 901
Analyze primer properties, 363
Annotate tag experiment, 757
Annotation
    select, 185
Annotation Layout, in Side Panel, 189
Annotation level, 766
Annotation tests, 817
    Gene set enrichment analysis (GSEA), 819
    GSEA, 819
    Hypergeometric test, 817

Annotation types
define your own, 193
Annotation Types, in Side Panel, 189
Annotations
add, 193
add to experiment, 769
copy to other sequences, 447
edit, 193, 195
expression analysis, 769
extract, 268
in alignments, 447
introduction to, 189
links, 216
overview of, 192
show/hide, 189
table of, 192
trim, 375
types of, 189
view on sequence, 189
viewing, 189
Annotations, add links to, 195
Antigenicity, 319, 886
Append wildcard, search, 203, 206, 209
Arrange
views in View Area, 47
Array data formats, 916
Array platforms, 916
Assemble
de novo, 830
report, 589
sequences, 375
to existing contig, 383
to reference sequence, 380, 572
Assembly, 884
variance table, 391
Atomic composition, 290
attB sites, add, 408
Attributes, 70
Audit, 83
Automation, 148

Back-up, attribute, 71
Backup, 126
BAM format, 116
BAM, export format specification, 912
BAM, file format, 905
Base pairs
required for mispriming, 351
Batch edit element properties, 69
Batch processing, 139
log of, 143
Bibliography, 938
Binding site for primer, 364
Bioinformatic data
export, 120
formats, 96, 903
bl2seq, *see* Local BLAST
BLAST, 885
against a local Database, 222
against NCBI, 219
contig, 389
create database from file system, 233
create database from Navigation Area, 233
create local database, 233
database management, 235
graphics output, 228
list of databases, 893
parameters, 220, 224
search, 218, 219
sequencing data, assembled, 389
specify server URL, 87
table output, 229
URL, 87
BLAST database index, 234
BLAST DNA sequence
BLASTn, 220, 223
BLASTx, 220, 223
tBLASTx, 220, 223
BLAST Protein sequence
BLASTp, 220, 223
tBLASTn, 220, 223
BLAST result
search in, 231
BLAST search
Bioinformatics explained, 236
BLAST search, Protein Data Bank, 248
BLOSUM, scoring matrices, 281
Bootstrap tests, 470
Borrow network license, 30
Box plot, 784
BP reaction, Gateway cloning, 413
Broken pair coloring, 387, 599
Broken pairs, find mates, 600
Browser,import sequence from, 97
Bug reporting, 35

C/G content, 182
CAS, file format, 905

CASAVA1.8, paired data, 105
CDS, translate to protein, 185
Chain flexibility, 182
Cheap end gaps, 438
ChIP sequencing, 875
ChIP-Seq analysis, 884
ChIP-seq analysis, 850, 853
ChIP-seq Quality Control, 851
Chromatin immunoprecipitation, *see* ChIP sequencing
Chromatogram traces
    scale, 371
.cif, file format, 910
Circular view of sequence, 186, 885
.clc, file format, 126, 910
CLC Standard Settings, 90
CLC Workbenches, 34
CLC, file format, 904–906, 910
    associating with *CLC Genomics Workbench*, 16
Cleavage, 335
    the Peptidase Database, 339
Cloning, 396, 885, 888
    insert fragment, 406
Close view, 46
Clustal, file format, 906
Cluster linkage
    Average linkage, 790
    Complete linkage, 790
    Single linkage, 790
Coding sequence, translate to protein, 185
Codon
    frequency tables, reverse translation, 333
    usage, 334
.col, file format, 910
Color residues, 443
Color space
    tag profiling, 749
Comments, 197
Common name
    batch edit, 69
Compare workbenches, 884
Compatible ends, 423
Complete Genomics data, 115
Complexity plot, 284
Configure network, 40
Conflicting enzymes, 429
Conflicts, overview in assembly, 391
Consensus sequence, 442, 886
    extract, 595
    open, 442, 595
Consensus sequence, extract, 389, 610
Conservation, 442
    graphs, 886
Contact information, 15
Contig, 884
    ambiguities, 391
    BLAST, 389
    create, 375
    reverse complement, 385
    view and edit, 383
Contigs
    map reads, 846
Copy, 134
    annotations in alignments, 447
    elements in Navigation Area, 65
    into sequence, 186
    search results, GenBank, 205
    search results, structure search, 211
    search results, UniProt, 208
    sequence, 198
    sequence selection, 304
    text selection, 198
Cores, maximum limit, 19
Cores, using multiple, 889
Count
    small RNAs, 731
    tag profiling, 749
Coverage, definition of, 584
.cpf, file format, 87
.chp, file format, 910
CPU cores, maximum limit, 19
CPU usage and multiple cores, 889
Create
    alignment, 437
    dot plots, 274
    enzyme list, 433
    local BLAST database, 233
    new folder, 64
    workspace, 57
Create a workflow, 149
Create index file, BLAST database, 234
Create tree, 458
Create Trees, 456
Create virtual tag list, 753
csfasta, file format, 108

CSV
export graph data points, 133
formatting of decimal numbers, 123
.csv, file format, 910
CSV, file format, 906, 907, 910
.ct, file format, 910
Custom annotation types, 193
Custom fields, 70
Customizing visualization, 3D structure, 252

Dark, color of broken pairs, 387, 599
Data
storage location, 63
Data formats
bioinformatic, 903
graphics, 911
Data preferences, 86
Data sharing, 63
Data structure, 62
Database
GenBank, 202
local, 62
NCBI, 232
nucleotide, 893
peptide, 893
shared BLAST database, 232
structure, 208
UniProt, 206
Db source, 197
db_xref references, 216
De novo, assembly, 830
de-multiplexing, 524
Delete
element, 68
residues and gaps in alignment, 447
workspace, 58
Description, 197
batch edit, 69
DGE, 885
Digital gene expression, 885
Digital gene expression(DGE), 713
tag-based, 749
DIP detection, 884
Dipeptide distribution, 291
Discovery studio
file format, 904
Distance based reconstruction methods
neighbor joining, 468
UPGMA, 468
Distance measure, 790
Distance, pairwise comparison of sequences in alignments, 452
DNA translation, 305
DNAstrider, file format, 904
Dot plots, 887
Bioinformatics explained, 277
create, 274
print, 276
Double cutters, 418
Double stranded DNA, 178
Download 3D Protein Structure Database, 710
Download and open
search results, GenBank, 205, 211
search results, UniProt, 208
Download and save
search results, GenBank, 205, 211
search results, UniProt, 208
Download of *CLC Genomics Workbench*, 15
Download, 3D Protein Structure Database, 710
Drag and drop
folder editor, 69
Navigation Area, 65
search results, GenBank, 205, 211
search results, UniProt, 208
DS Gene
file format, 904
Dual screen support, 50

E-PCR, 364
Edit
alignments, 446, 886
annotations, 193, 195, 885
enzymes, 419
sequence, 186
sequences, 885
single bases, 186
Element
delete, 68
rename, 67
.embl, file format, 910
Embl, file format, 904
Encapsulated PostScript, export, 131
End gap cost, 438
End gap costs
cheap end caps, 438
free end gaps, 438
Entry clone, creating, 413
Enzyme list, 432

create, 433
edit, 434
view, 434
Epigenomics, ChIP sequencing, 875
.eps-format, export, 131
Error reports, 35
Example data, import, 37
Excel 2007, file format, 907
Excel 2010, file format, 907
Excel, export file format, 910
Expand selection, 185
Expect, BLAST search, 229
Experiment
set up, 760
**Experiment**, 760
Export
bioinformatic data, 120
dependent objects, 124
folder, 124
graph in csv format, 133
graphics, 128
history, 125
list of formats, 903
mapping coverage, 905
preferences, 87
Side Panel Settings, 85
table, 127
tables, 906, 910
workflow output, 127
Export visible area, 129
Export whole view, 129
Expression analysis, 885
Expression clone, creating, 415
Expression data
annotation files, 916
Extensions, 37
External files, import and export, 97
Extinction coefficient, 290
Extract
Consensus sequence, 610
part of a mapping, 389
Extract and count small RNAs, 731
Extract and count tags, 749
Extract consensus sequences, from mapping table, 595
Extract sequences, 270

FASTA, file format, 904
FASTQ, file format, 104, 904
Favorite tools, 56
Feature clustering, 809
K-means clustering, 813
K-medoids clustering, 813
Feature request, 35
Feature table, 291
Feature, for expression analysis, 760
Features, *see* Annotations
File name, sort sequences based on, 377
File system, local BLAST database, 233
Filtering restriction enzymes, 420, 422, 425, 434
Find
in GenBank file, 198
in sequence, 183
results from a finished process, 54
Find open reading frames, 307
Find, in tracks, 551
Fit to pages, print, 93
Fixpoints, for alignments, 440
Folder editor
drag and drop, 69
Follow selection, 178
Footer, 94
Format, of the manual, 41
FormatDB, 233
Fragment table, 429
Fragment, select, 185
Fragments, separate on gel, 430
Free end gaps, 438
Freezer position, 70
Frequently used tools, 56
.fsa, file format, 910

G/C content, 182, 886
G/C restrictions
3' end of primer, 346
5' end of primer, 346
End length, 346
Max G/C, 346
Gap
compare number of, 452
delete, 447
extension cost, 438
fraction, 443, 886
insert, 446
open cost, 438
Gateway cloning
add attB sites, 408

create entry clones, 413
create expression clones, 415
Gb Division, 197
.gbk, file format, 910
GC content, 345
GCG Alignment, file format, 906
GCG Sequence, file format, 904
.gck, file format, 910
GCK, Gene Construction Kit file format, 904
Gel
separate sequences without restriction enzyme digestion, 431
tabular view of fragments, 429
Gel electrophoresis, 430, 888
marker, 432
view, 431
view preferences, 431
when finding restriction sites, 427
GenBank
view sequence in, 198
file format, 904
search, 202, 885
search sequence in, 215
Gene Construction Kit, file format, 904
Gene expression analysis, 885
Gene expression, sequencing-based, 713
Gene expression, sequencing-by tag, 749
Gene finding, 307
General preferences, 82
General Sequence Analyses, 268
Genetic code, reverse translation, 334
Genome browser, 544
GEO, file format, 906
GFF, 99
.gff, file format, 910
GO, import annotation file, 922
Google sequence, 215
GOstats, *see* Hypergeometric tests on annotations
Graph
export data points in csv format, 133
Graph Side Panel, 891
Graphics
data formats, 911
export, 128
Groups, define, 760
.gzip, file format, 910
Gzip, file format, 910
Half-life, 289
Handling of results, 142
Header, 94
Heat map, 885
clustering of features, 811
clustering of samples, 791
Help, 36
Heterozygotes, discover via secondary peaks, 394
Hide/show Toolbox, 54
Hierarchical clustering
of features, 809
of samples, 788
High-throughput sequencing, 884
Histogram, 823
Distributions, 823
History, 136
export, 125
preserve when exporting, 137
source elements, 137
Homology, pairwise comparison of sequences in alignments, 452
Hotelling observer, 852
Hydrophobicity, 321, 886
Bioinformatics explained, 323
Chain Flexibility, 325
Cornette, 182, 324
Eisenberg, 182, 324
Emini, 182
Engelman (GES), 182, 324
Hopp-Woods, 182, 324
Janin, 182, 324
Karplus and Schulz, 182
Kolaskar-Tongaonkar, 182, 325
Kyte-Doolittle, 182, 324
Rose, 324
Surface Probability, 325
Welling, 182, 324
Hypergeometric tests on annotations, 817

ID, license, 23
Illumina Genome Analyzer, 884
Import
bioinformatic data, 96, 97
from a web page, 97
High-throughput sequencing data, 100
list of formats, 903
Next-Generation Sequencing data, 100
NGS data, 100

preferences, 87
raw sequence, 97
Side Panel Settings, 85
using copy paste, 97
Import issues, 249
Import protein structure, BLAST, 248
Import protein structure, from file, 247
Import protein structure, Protein Data Bank, 247
Improvements, 41
In silico PCR, 364
Index for searching, 80
Information point, primer design, 343
Insert
gaps, 446
Insert restriction site, 407
Inspecting results, 856
Installation, 15
Invert sequence, 304
Isoelectric point, 288
Isoschizomers, 423
IUPAC codes
nucleotides, 902

Join
alignments, 449
sequences, 291
.jpg-format, export, 131

K-means clustering, 813
K-medoids clsutering, 813
K-mer based distance estimation, 467
K-mer Based Tree Construction, 457
Keywords, 197

Label
of sequence, 178
Landscape, Print orientation, 93
Lasergene sequence
file format, 904
Latin name
batch edit, 69
Length, 197
License, 19
ID, 23
non-networked machine, 32
starting without a license, 34
License server, 29
License server: access offline, 30
Limited mode, 34
Link Variants to 3D Protein Structure, 700
Linker trimming, 516
Links, from annotations, 195
Linux
installation, 17
installation with RPM-package, 18
List of restriction enzymes, 432
List of sequences, 198
Load enzyme list, 419
Local BLAST, 222
Local BLAST Database, 233
Local BLAST database management, 235
Local BLAST Databases, 232
Local complexity plot, 284, 885
Local Database, BLAST, 222
Locale setting, 83
Location
search in, 79
path to, 63
Locations
multiple, 884
Log of batch processing, 143
Logo, sequence, 444, 886
LR reaction, Gateway cloning, 415

MA plot, 825
.ma4, file format, 910
Mac OS X installation, 16
Manage BLAST databases, 235
Manipulate sequences, 885, 888
Manual editing, auditing, 83
Manual format, 40
Map
to coding regions, 573
Map reads to reference
select reference sequences, 572
Map reads to reference
masking, 573
Map reads, contigs, 846
Mapping
report, 582
Mapping reads to a reference sequence, 572
Mapping table, 595
Mappings
merge, 609
Marker, in gel view, 432
Mask, reference sequence, 573
Match weight, 747

Mates, locate from broken pairs, 600
Maximize size of view, 49
Maximum likelihood, 887
Maximum Likelihood Phylogeny, 462
Maximum likelihood reconstruction methods, 469
Melting temperature
    DMSO concentration, 345
    dNTP concentration, 345
    Magnesium concentration, 345
Melting temperature, 345
    Cation concentration, 345, 364
    Cation concentration, 365
    Inner, 345
    Primer concentration, 345, 364
    Primer concentration, 365
Menu Bar, illustration, 43
Merge mapping results, 609
Meta data, 70
MFold, 887
Microarray data formats, 916
Microarray platforms, 916
microRNA analysis, 730
Mixed data, 609
mmCIF, file format, 910
Model testing, 460
Modification date, 197
Modify enzyme list, 434
Modules, 37
Molecular weight, 288
Monitors, supporting multiple monitors, 50
Motif list, 300
Motif search, 294, 300, 887
Mouse modes, 52
Move
    elements in Navigation Area, 65
    sequence to top, 447
    sequences in alignment, 447
mRNA sequencing
    by tag, 749
.msf, file format, 910
Multi-group experiment, 761
Multiple alignments, 453, 886
Multiple testing
    Benjamini-Hochberg corrected p-values, 806
    Benjamini-Hochberg FDR, 806
    Bonferroni, 806
    Correction of p-values, 806
    FDR, 806
Multiplexing, 524
    by name, 377
Multiselecting, 65

N50, 582
Name, 197
Navigate, 3D structure, 250
Navigation Area, 62
    create local BLAST database, 233
    illustration, 43
NCBI, 202
    search for structures, 208
    search sequence in, 215
NCBI BLAST
    add more databases, 894
Negatively charged residues, 290
Neighbor joining, 468
Neighbor-joining, 887
Nested PCR primers, 887
NetAffx annotation files, 920
Network configuration, 40
Network drive, shared BLAST database, 232
Network license, 29
Network license: use offline, 30
Never show this dialog again, 83
New
    feature request, 35
    folder, 64
New sequence
    create from a selection, 185
Newick, file format, 906
Next-Generation Sequencing, 884
.nexus, file format, 910
Nexus, file format, 904, 906
NGS, 884
.nhr, file format, 910
Non-coding RNA analysis, 730
Non-specific matches, 578
Non-standard residues, 180
Normalization, 782
    Quantile normlization, 782
    Scaling, 782
Nucleotide
    info, 180
    sequence databases, 893
Nucleotides
    UIPAC codes, 902
Numbers on sequence, 178

.nwk, file format, 910
.nxs, file format, 910

.oa4, file format, 910
Open
    consensus sequence, 442
    from clipboard, 97
Open reading frame determination, 307
Open-ended sequence, 308
Order primers, 368, 887
ORF, 307
Organism, 197
Originates from, 137
Overhang
    of fragments from restriction digest, 429
Overhang, find restriction enzymes based on, 420, 422, 425, 433
Overlapping paired reads, 599

.pa4, file format, 910
Page heading, 94
Page number, 94
Page setup, 93
Paired data, 110, 113, 116
Paired reads
    combined with single reads, 609
    overlapping, 599
Paired samples, expression analysis, 761
Paired status, 116
Pairwise comparison, 450
PAM, scoring matrices, 281
Parallelization, 889
Parameters
    search, 203, 206, 209
Partition function, 489, 887
Partitioning around medoids (PAM), *see* K-medoids clustering
Paste
    text to create a new sequence, 97
Paste/copy, 134
Pattern Discovery, 292
Pattern discovery, 887
Pattern Search, 294
PCA, 793
PCR primers, 887
PCR, perform virtually, 364
.pdb, file format, 246, 910
.seq, file format, 910
PDB, file format, 910
.pdf-format, export, 131
Peak finding, ChIP sequencing, 875
Peak shape, 852
Peak shape score, 853
Peak, call secondary, 394
Peptidase, 335
Peptide sequence databases, 893
Percent identity, pairwise comparison of sequences in alignments, 452
Personal information, 35
Pfam domain search, 326, 886
.phr, file format, 910
Phred, file format, 904
.phy, file format, 910
Phylip, file format, 906
Phylogenetic tree, 458, 887
Phylogenetic tree methods, 465
Phylogenetic trees
    add or modify metadata, 481
    background settings, 475
    bootstrap settings, 477
    bootstrap tests, 470
    branch layout, 475
    create tree, 458
    create trees, 456
    features, 455
    K-mer based distance estimation, 467
    K-mer based tree construction, 457
    label settings, 474
    maximum likelihood phylogeny, 462
    maximum likelihood reconstruction methods, 469
    metadata, 477, 480
    minimap, 471
    model testing, 460
    neighbor joining, 468
    node right click menu, 478
    node settings, 473
    selection of nodes, 483
    substitution models and distance estimation, 466
    table settings and filtering, 481
    tree layout, 471
    tree settings, 470
    UPGMA, 468
Pipeline, 148
.pir, file format, 910
PIR (NBRF), file format, 904

Plot
dot plot, 274
local complexity, 284
Plugins, 37
.png-format, export, 131
Polarity colors, 180
Portrait, Print orientation, 93
Positively charged residues, 290
PostScript, export, 131
Preference group, 88
Preferences, 82
advanced, 87
Data, 86
export, 87
General, 82
import, 87
style sheet, 88
toolbar, 84
View, 84
view, 50
Primer, 364
analyze, 363
based on alignments, 358
Buffer properties, 345
design, 887
design from alignments, 887
display graphically, 347
length, 345
mode, 346
nested PCR, 346
order, 368
sequencing, 346
standard, 346
TaqMan, 346
Primers
find binding sites, 364
Principal component analysis, 793
Scree plot, 796
Print, 91
dot plots, 276
preview, 94
visible area, 92
whole view, 92
.pro, file format, 910
Problems when starting up, 35
Processes, 54
Properties, batch edit, 69
Protease, cleavage, 335
Protein
charge, 317, 886
cleavage, 335
hydrophobicity, 323
Isoelectric point, 288
report, 329, 885
report, output, 331
signal peptide, 311
statistics, 288
structure prediction, 328
translation, 332
Protein Data Bank, 247
Proteolytic cleavage, 335, 886
Bioinformatics explained, 338
Proteolytic enzymes cleavage patterns, 896
Proxy server, 40
.ps-format, export, 131
.psi, file format, 910
PubMed references, search, 216
PubMed references,search, 885

QC, 784
QSEQ,file format, 104
Quality control
MA plot, 825
Quality of chromatogram trace, 371
Quality of trace, 373, 514
Quality score of trace, 373, 514
Quality scores, 181
Quick start, 36

Rasmol colors, 180
Read mapping, 572
Reading frame, 307
Realign alignment, 886
Reassemble contig, 393
Rebase, restriction enzyme database, 433
Rebuild index, 80
Recognition sequence
insert, 407
Recover removed attribute, 71
Recycle Bin, 68
Redo alignment, 439
Redo/Undo, 47
Reference assembly, 572
Reference sequence, 884
References, 938
Region
types, 186

Remove
annotations, 196
sequences from alignment, 448
terminated processes, 55
Rename element, 67
Repeat masking, 573
Report
of assembly, 589
Report program errors, 35
Report, protein, 885
Request new feature, 35
Residue coloring, 180
Restore
deleted elements, 68
size of view, 49
Restriction enzmyes
filter, 420, 422, 425, 434
from certain suppliers, 420, 422, 425, 434
Restriction enzyme list, 432
Restriction enzyme, star activity, 433
Restriction enzymes, 416
compatible ends, 423
cutting selection, 421
isoschizomers, 423
methylation, 420, 422, 425, 433
number of cut sites, 418
overhang, 420, 422, 425, 433
separate on gel, 430
sorting, 418
Restriction sites, 416, 886
enzyme database Rebase, 433
select fragment, 185
number of, 426
on sequence, 179, 417
parameters, 424
Results handling, 142
Reverse complement, 304, 886
Reverse complement mapping, 385
Reverse sequence, 304
Reverse translation, 332, 886
Bioinformatics explained, 333
Right-click on Mac, 40
RNA secondary structure, 887
RNA structure
partition function, 489
RNA structure prediction by minimum free energy minimization
Bioinformatics explained, 506
RNA translation, 305
RNA-Seq analysis, 713, 884
.rnaml, file format, 910
Rotate, 3D structure, 250
RPKM,definition, 730

Safe mode, 35
SAGE
tag-based mRNA sequencing, 749
SageScreen, tag profiling by, 749
SAM format, 116
SAM, export format specification, 912
SAM, file format, 905
Sample, for expression analysis, 760
Save
changes in a view, 46
style sheet, 88
view preferences, 88
workspace, 57
Save enzyme list, 419
Scale bar, 470
Scale traces, 371
SCARF, file format, 104
Scatter plot, 828
SCF2, file format, 904
SCF3, file format, 904
Score, BLAST search, 229
Scoring matrices
Bioinformatics explained, 281
BLOSUM, 281
PAM, 281
Scree plot, 796
Screen, multiple screen support, 50
Scripting, 148
Scroll wheel
to zoom in, 53
to zoom out, 53
Search, 78
in one location, 79
BLAST, 218, 219
for structures at NCBI, 208
GenBank, 202
GenBank file, 198
handle results from GenBank, 204
handle results from NCBI structure DB, 210
handle results from UniProt, 207
hits, number of, 83
in a sequence, 183
in annotations, 183

in Navigation Area, 75
Local BLAST, 222
local data, 884
options, GenBank, 203
options, GenBank structure search, 209
options, UniProt, 206
own motifs, 300
parameters, 203, 206, 209
patterns, 292, 294
Pfam domains, 326
PubMed references, 216
sequence in UniProt, 216
sequence on Google, 215
sequence on NCBI, 215
sequence on web, 215
TrEMBL, 206
troubleshooting, 80
UniProt, 206
Search, in tracks, 551
Secondary peak calling, 394
Secondary structure
predict RNA, 887
Secondary structure prediction, 328, 886
Secondary structure, for primers, 346
Select
exact positions, 183
in sequence, 185
parts of a sequence, 185
workspace, 58
Select annotation, 185
Selection mode in the toolbar, 54
Selection, adjust, 185
Selection, expand, 185
Self annealing, 345
Self end annealing, 346
Separate sequences on gel, 431
using restriction enzymes, 430
Sequence
alignment, 436
analysis, 268
display different information, 67
extract from sequence list, 270
find, 183
information, 197
join, 291
layout, 178
lists, 198
logo, 886
logo Bioinformatics explained, 444
region types, 186
search, 183
select, 185
shuffle, 272
statistics, 285
view, 177
view as text, 198
view circular, 186
view format, 67
web info, 215
Sequence comma separated values, file format, 904
Sequence logo, 444
Sequencing data, 884
Sequencing primers, 887
Share data, 63, 884
Share Side Panel Settings, 85
Shared BLAST database, 232
Shortcuts, 58
Show
enzymes cutting selection, 421
results from a finished process, 54
Show dialogs, 83
Show enzymes with compatible ends, 423
Show Side Panel, 84
Show/hide Toolbox, 54
Shuffle sequence, 272, 885
Side Panel Settings
export, 85
import, 85
share with others, 85
Side Panel, show, 84
Signal peptide, 311, 312, 886
SignalP, 311
Bioinformatics explained, 312
Single base editing
in mapping, 388
in sequences, 186
Single cutters, 418
Small RNA analysis, 730
Small RNAs
extract and count, 731
trim, 731
Snippets, 163
SNP detection, 860, 884
SNV
detect, 860

SNV detection, 860
Solexa, *see* Illumina Genome Analyzer
SOLiD data, 884
Sort
    sequences alphabetically, 447
    sequences by similarity, 447
Sort sequences by name, 377
Sort, folders, 65
Source element, 137
Species, display name, 67
Staden, file format, 904
Standard Settings, CLC, 90
Star activity, 433
Start Codon, 308
Start-up problems, 35
Statistical analysis, 798
    ANOVA, 798
    Corrected of p-values, 806
    Paired t-test, 798
    Repeated measures ANOVA, 798
    t-test, 798
    Volcano plot, 807
Statistics
    about sequence, 885
    protein, 288
    sequence, 285
Status Bar, 54, 57
    illustration, 43
.str, file format, 910
Structure editor, 250
Structure scanning, 887
Structure, prediction, 328
Style sheet, preferences, 88
Subcontig, extract part of a mapping, 389
Substitution models and distance estimation, 466
Support, 35
Surface probability, 182
.svg-format, export, 131
Swiss-Prot, 206
    search, *see* UniProt
Swiss-Prot, file format, 904
Swiss-Prot/TrEMBL, 885
.swp, file format, 910
System requirements, 18

Tab delimited, file format, 906, 910
Tab, file format, 904
Table of fragments, 429
Tabs, use of, 45
Tag profiling, 749
    annotate tag experiment, 757
    create virtual tag list, 753
Tag-based expression profiling, 884
Tags
    extract and count, 749
Tags, insert into sequence, 407
TaqMan primers, 887
.tar, file format, 910
Tar, file format, 910
Taxonomy
    batch edit, 69
tBLASTn, 220, 223
tBLASTx, 220, 223
Terminated processes, 55
Text format, 184
    user manual, 41
    view sequence, 198
Text, file format, 910
.tif-format, export, 131
TMHMM, 318
Toolbar
    illustration, 43
    preferences, 84
Toolbox, 54, 55
    illustration, 43
    show/hide, 54
Trace colors, 180
Trace data, 371, 884
    quality, 373, 514
Traces
    scale, 371
Track list
    in workflow, 555
Track Lists, 544
Tracks, 542
Transcriptome sequencing, 713
    tag-based, 749
Transcriptomics, 713
    tag-based, 749
Transformation, 781
Translate
    a selection, 181
    along DNA sequence, 180
    annotation to protein, 185
    CDS, 307
    coding regions, 307

DNA to RNA, 302
nucleotide sequence, 305
ORF, 307
protein, 332
RNA to DNA, 303
to DNA, 886
to protein, 305, 886
Translation
of a selection, 181
show together with DNA sequence, 180
Transmembrane helix prediction, 318, 886
Tree generation, methods, 465
TrEMBL, search, 206
Trim, 372, 513, 884
small RNAs, 731
Trimmed regions
adjust manually, 384
TSV, file format, 904
Two-color arrays, 916
Two-group experiment, 761
.txt, file format, 910

UIPAC codes
amino acids, 901
Undo limit, 83
Undo/Redo, 47
UniProt, 206
search, 206, 885
search sequence in, 216
UniVec, trimming, 373, 514
UPGMA, 468
UPGMA algorithm, 887
Urls, Navigation Area, 97
User defined view settings, 84
User interface, 43

Variance table, assembly, 391
Variant
detect, 628
Variant callers, 628
Variant data, 672
Variant detection, 628
Variant tracks, 672
Variants, Link to 3D Protein Structure, 700
VCF, 907
Vector
see cloning, 396
Vector contamination, find automatically, 373, 514
Vector design, 396
Vector graphics, export, 131
Vector NTI import, 98
VectorNTI
file format, 904
View, 44
alignment, 442
dot plots, 275
GenBank format, 198
preferences, 50
save changes, 46
sequence, 177
sequence as text, 198
View Area, 44
illustration, 43
View preferences, 84
show automatically, 84
style sheet, 88
View settings
user defined, 84
Virtual gel, 888
Virtual tag list
create, 753
how to annotate, 757
Visualization styles, 3D structure, 252
Volcano plot, 807
.vsf, file format for settings, 85

Web page, import sequence from, 97
Wildcard, append to search, 203, 206, 209
Windows installation, 15
Workflow, 148
adding elements to existing workflow, 163
configure elements, 150
connect elements, 152
create, 149
input modifying tools, 158
layout, 158
lock and unlock parameters, 151
reusing elements from workflow, 163
snippets, 163
validation, 160
Workspace, 57
create, 57
delete, 58
save, 57
select, 58
Wrap sequences, 178

.xls, file format, 910
.xlsx, file format, 910
.xml, file format, 910
XSQ, file format, 108

Zip, file format, 910
Zoom, 52
Zoom In, 53
Zoom Out, 53
Zoom, 3D structure, 250