CLC Phylogeny Module

User manual

User manual for

Phylogeny Module 1.0

Windows, Mac OS X and Linux

September 13, 2013

This software is for research purposes only.

CLC bio
Silkeborgvej 2
Prismet
DK-8000 Aarhus C
Denmark

Contents

Chapter 1

Introduction to the Phylogeny Module

1.1 Phylogeny

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.

1.2 Features

The Phylogeny Module comes with a greatly enhanced viewer for visualizing and working with phylogenetic trees. The viewer allows the user to rapidly create high-quality, publication-ready figures of phylogenetic trees. Large trees are made easy to explore using different zoom functionalities and a small minimap of the entire tree. The viewer also comes with two alternative tree layouts, namely circular layouts and radial layouts, which are great for visualizing very large trees. Finally, the new viewer supports importing, editing and visualization of metadata associated with nodes in phylogenetic trees. The Phylogeny Module is scheduled to become part of the CLC Main Workbench and the CLC Genomics Workbench in the near future.

In addition to the new viewer, two new tools are included in the module. The first tool can reconstruct phylogenetic trees based on k-mers. This approach avoids the computationally intensive step of constructing a multiple alignment of the input sequences. However the accuracy of the constructed tree might not be as high as the other reconstruction methods. The k-mer based reconstruction 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. The second new tool implements a statistic evaluation of different substitution models to be used with maximum likelihood tree construction, similar to that produced by the tool Model Testing [Posada and Crandall, 1998]. The output of this tool is a report that lists the recommended settings to be used when constructing phylogenetic trees based on maximum likelihood. Below is an overview of these tools and the main features of the new editor. Further details can be found in the subsequent sections.

- **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.
- **K-mer based tree construction**.
  - Construction of phylogenetic trees without a time consuming multiple alignment of the input sequences [Blaisdell, 1989].
- **Model Testing**.
  - Tool for selecting a substitution model for use with maximum likelihood tree construction.
  - Supports comparison of five substitution models (Jukes-Cantor, Felsenstein 81, Kimura 80, Hasegawa-Kishino-Yano, General Time Reversible), optional rate variation and topology variation.
  - Compares models based on hierarchical likelihood ratio tests, Bayesian information criterion and Akaike minimum theoretical information criterion.

Chapter 2

System requirements and installation of the Phylogeny Module

2.1 System requirements

The system requirements of the Phylogeny Module are:

- Windows XP, Windows Vista, or Windows 7, Windows Server 2003 or Windows Server 2008
- Mac OS X 10.6 or later. However, Mac OS X 10.5.8 is supported on 64-bit Intel systems.
- Linux: Red Hat 5.0 or later. SUSE 10.2 or later. Fedora 6 or later.
- 32 or 64 bit
- 1 GB RAM required
- 2 GB RAM recommended
- 1024 x 768 display recommended
- CLC Genomics Workbench

2.2 How to install a plug-in

Plug-ins are installed using the plug-in manager[1]:

Help in the Menu Bar | Plug-ins and Resources... ()

or **Plug-ins () in the Toolbar**

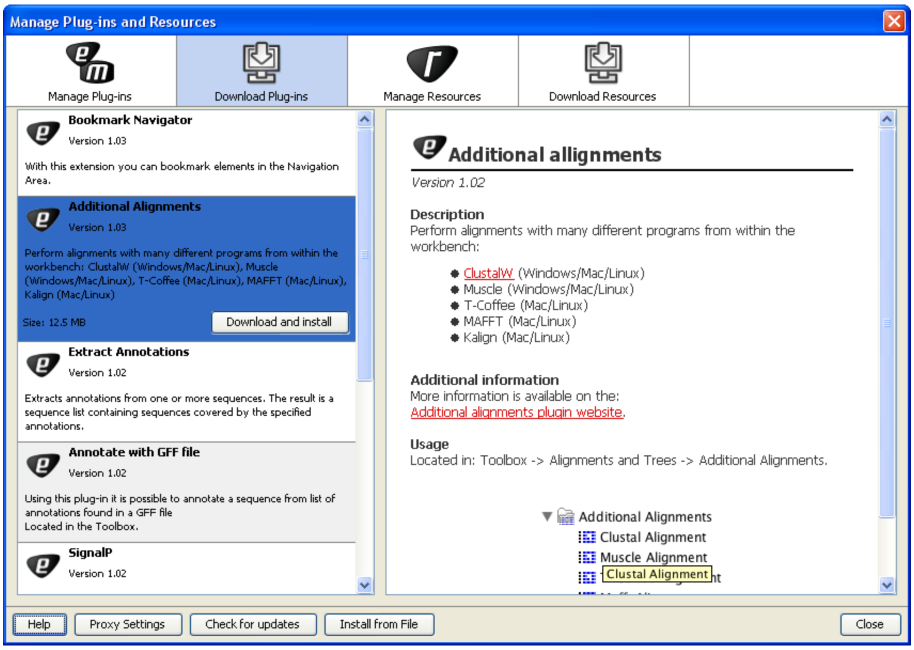
The plug-in manager has four tabs at the top:

- **Manage Plug-ins.** This is an overview of plug-ins that are installed.
- **Download Plug-ins.** This is an overview of available plug-ins on CLC bio's server.

[1] In order to install plug-ins 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.

- **Manage Resources.** This is an overview of resources that are installed.
- **Download Resources.** This is an overview of available resources on CLC bio's server.

To install a plug-in, click the **Download Plug-ins** tab. This will display an overview of the plug-ins that are available for download and installation (see figure 2.1).



Bookmark Navigator
Version 1.03
With this extension you can bookmark elements in the Navigation Area.

Additional Alignments
Version 1.03
Perform alignments with many different programs from within the workbench: ClustalW (Windows/Mac/Linux), Muscle (Windows/Mac/Linux), T-Coffee (Mac/Linux), MAFFT (Mac/Linux), Kalign (Mac/Linux)
Size: 12.5 MB Download and install

Extract Annotations
Version 1.02
Extracts annotations from one or more sequences. The result is a sequence list containing sequences covered by the specified annotations.

Annotate with GFF file
Version 1.02
Using this plug-in it is possible to annotate a sequence from list of annotations found in a GFF file
Located in the Toolbox.

SignalP
Version 1.02

Additional allignments
Version 1.02

Description
Perform alignments with many different programs from within the workbench:

- ClustalW (Windows/Mac/Linux)
- Muscle (Windows/Mac/Linux)
- T-Coffee (Mac/Linux)
- MAFFT (Mac/Linux)
- Kalign (Mac/Linux)

Additional information
More information is available on the:
Additional alignments plugin website.

Usage
Located in: Toolbox -> Alignments and Trees -> Additional Alignments.

Additional Alignments
Clustal Alignment
Muscle Alignment
Clustal Alignment

Figure 2.1: *The plug-ins that are available for download.*

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

Click the Phylogeny Module and press **Download and Install**. A dialog displaying progress is now shown, and the plug-in is downloaded and installed.

If the Phylogeny Module Plug-in 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 plug-in. The plug-in 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 plug-in will not be ready for use before you have restarted.

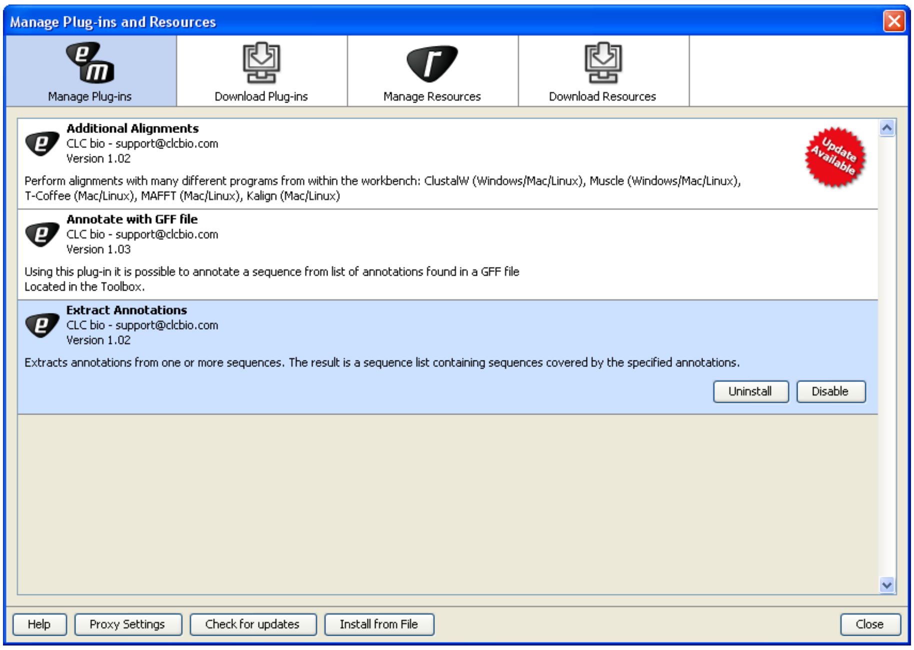
2.3 How to uninstall a plug-in

Plug-ins are uninstalled using the plug-in manager:

Help in the Menu Bar | Plug-ins and Resources... ( )

or **Plug-ins ( ) in the Toolbar**

This will open the dialog shown in figure 2.2.



Additional Alignments
CLC bio - support@clcbio.com
Version 1.02

Perform alignments with many different programs from within the workbench: ClustalW (Windows/Mac/Linux), Muscle (Windows/Mac/Linux), T-Coffee (Mac/Linux), MAFFT (Mac/Linux), Kalign (Mac/Linux)

Annotate with GFF file
CLC bio - support@clcbio.com
Version 1.03

Using this plug-in it is possible to annotate a sequence from list of annotations found in a GFF file
Located in the Toolbox.

Extract Annotations
CLC bio - support@clcbio.com
Version 1.02

Extracts annotations from one or more sequences. The result is a sequence list containing sequences covered by the specified annotations.

Figure 2.2: *The plug-in manager with plug-ins installed.*

The installed plug-ins are shown in this dialog. To uninstall:

Click the Phylogeny Module | **Uninstall**

If you do not wish to completely uninstall the plug-in 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 plug-in will not be uninstalled before the workbench is restarted.

Chapter 3

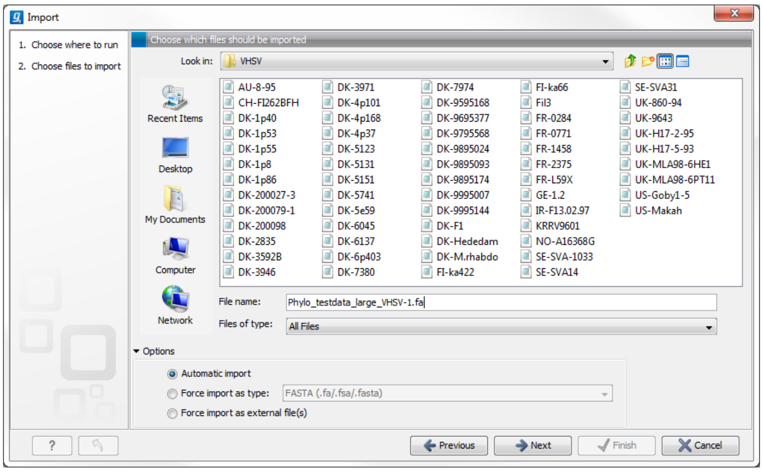
Alignment of sequences

Sequences that are not already in the Workbench can be imported in fasta format using the standard import function.

To import a fasta file:

> **File** | **Import** (📂) | **Standard Import** (📂)

Use "Automatic Import" to import the sequences (figure 3.1):



Import

1. Choose where to run
2. Choose files to import

Choose which files should be imported

Look in: VHSV

AU-8-95
CH-FI262BFH
DK-1p40
DK-1p53
DK-1p55
DK-1p8
DK-1p86
DK-200027-3
DK-200079-1
DK-200098
DK-2835
DK-3592B
DK-3946
DK-3971
DK-4p101
DK-4p168
DK-4p37
DK-5123
DK-5131
DK-5151
DK-5741
DK-5e59
DK-6045
DK-6137
DK-6p403
DK-7380
DK-7974
DK-9595168
DK-9695377
DK-9795568
DK-9895024
DK-9895093
DK-9895174
DK-9995007
DK-9995144
DK-F1
DK-Hededam
DK-M.rhabdo
FI-ka422
FI-ka66
Fil3
FR-0284
FR-0771
FR-1458
FR-2375
FR-L59X
GE-1.2
IR-F13.02.97
KRRV9601
NO-A16368G
SE-SVA-1033
SE-SVA14
SE-SVA31
UK-860-94
UK-9643
UK-H17-2-95
UK-H17-5-93
UK-MLA98-6HE1
UK-MLA98-6PT11
US-Goby1-5
US-Makah

File name: Phylo_testdata_large_VHSV-1.fa

Files of type: All Files

Options

● Automatic import
○ Force import as type: FASTA (.fa/.fsa/.fasta)
○ Force import as external file(s)

Figure 3.1: *Specify parameters for model testing.*

3.1 Create an alignment

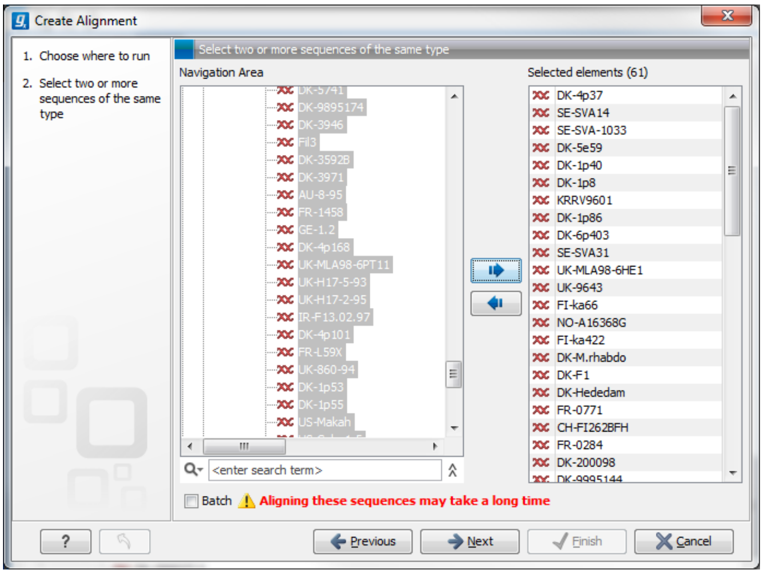
Alignments can be created from nucleotide or protein sequences, sequence lists, existing alignments and from any combination of the three.

To create an alignment in CLC Genomics Workbench:

> **Select Sequences to Align** | **Toolbox in the Menu Bar** | **Classical Sequence Analysis** (📂) | **Alignments and Trees** (📂)| **Create Alignment** (☷)

or **Select Sequences to Align** | **Right-click any selected sequence** | **Toolbox** | **Classical Sequence Analysis** (📂) | **Alignments and Trees** (📂)| **Create Alignment** (☷)

This opens the dialog shown in figure 3.2.



Create Alignment
1. Choose where to run
2. Select two or more sequences of the same type
Select two or more sequences of the same type
?
Previous
Next
Finish
Cancel

Navigation Area

DK-5741
DK-9895174
DK-3946
Fil3
DK-3592B
DK-3971
AU-8-95
FR-1458
GE-1.2
DK-4p168
UK-MLA98-6PT11
UK-H17-5-93
UK-H17-2-95
IR-F13.02.97
DK-4p101
FR-L59X
UK-860-94
DK-1p53
DK-1p55
US-Makah

<enter search term>

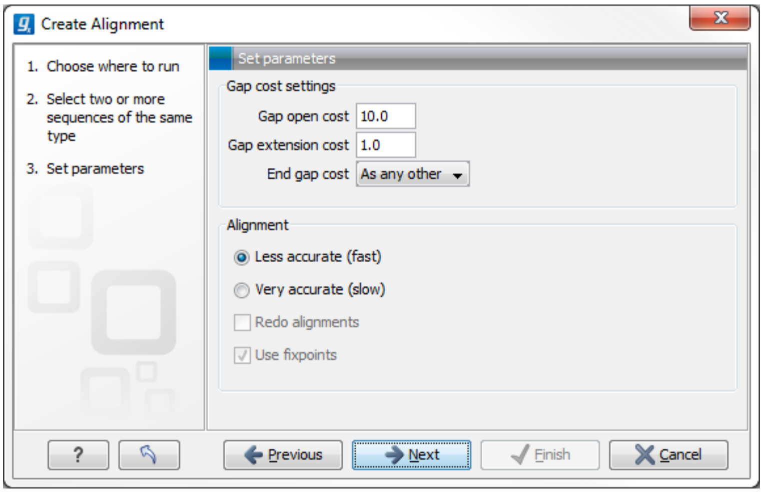
Selected elements (61)

DK-4p37
SE-SVA14
SE-SVA-1033
DK-5e59
DK-1p40
DK-1p8
KRRV9601
DK-1p86
DK-6p403
SE-SVA31
UK-MLA98-6HE1
UK-9643
FI-ka66
NO-A16368G
FI-ka422
DK-M.rhabdo
DK-F1
DK-Hededam
FR-0771
CH-FI262BFH
FR-0284
DK-200098
DK-9995144

□ Batch ⚠ Aligning these sequences may take a long time

Figure 3.2: *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 3.3.



Gap cost settings

Gap open cost 10.0

Gap extension cost 1.0

End gap cost As any other

Alignment

● Less accurate (fast)

○ Very accurate (slow)

□ Redo alignments

☑ Use fixpoints

Figure 3.3: *Adjusting alignment algorithm parameters.*

3.1.1 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 one decimal place.

- **Gap open cost** The penalty for introducing gaps in an alignment.
- **Gap extension cost** The penalty 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 set the Gap open cost higher than the Gap extension cost. The default values are 10.0 and 1.0 for the two parameters, respectively.

- **End gap cost** The penalty 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.

3.1.2 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.

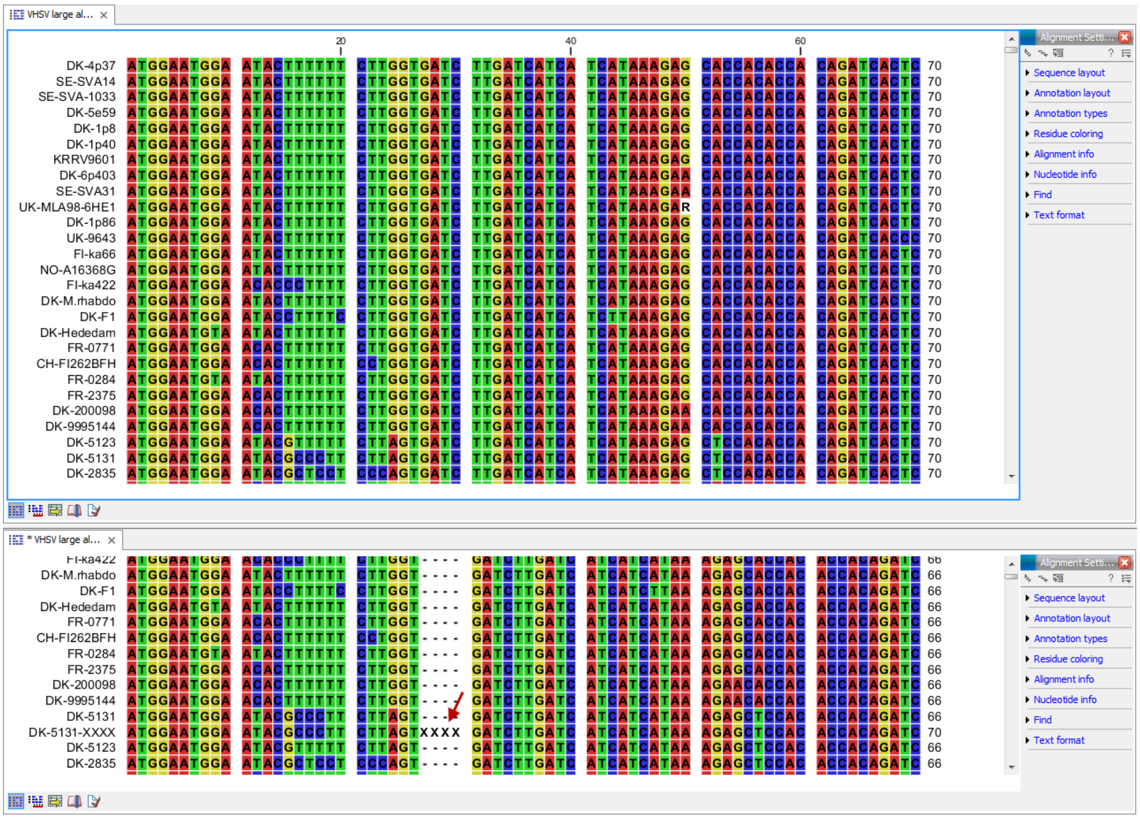
Both algorithms use progressive alignment. The faster algorithm builds the initial tree by doing more approximate pairwise alignments than the slower option.

3.1.3 Aligning alignments

If you have selected an existing alignment in the first step (figure 3.2), 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 3.4.

The top of figure 3.4 shows the original alignment. In the lower part of the figure, 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.



```
                                     20                    40                    60
                                      |                     |                     |
      DK-4p37 ATGGAATGGA ATACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAG CACCACACCA CAGATCACTC 70
     SE-SVA14 ATGGAATGGA ATACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAG CACCACACCA CAGATCACTC 70
  SE-SVA-1033 ATGGAATGGA ATACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAG CACCACACCA CAGATCACTC 70
      DK-5e59 ATGGAATGGA ATACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAG CACCACACCA CAGATCACTC 70
       DK-1p8 ATGGAATGGA ATACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAG CACCACACCA CAGATCACTC 70
      DK-1p40 ATGGAATGGA ATACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAG CACCACACCA CAGATCACTC 70
     KRRV9601 ATGGAATGGA ATACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAG CACCACACCA CAGATCACTC 70
     DK-6p403 ATGGAATGGA ATACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAA CACCACACCA CAGATCACTC 70
     SE-SVA31 ATGGAATGGA ATACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAA CACCACACCA CAGATCACTC 70
UK-MLA98-6HE1 ATGGAATGGA ATACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAR CACCACACCA CAGATCACTC 70
      DK-1p86 ATGGAATGGA ATACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAG CACCACACCA CAGATCACTC 70
      UK-9643 ATGGAATGGA ATACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAG CACCACACCA CAGATCACCC 70
      FI-ka66 ATGGAATGGA ATACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAG CACCACACCA CAGATCACTC 70
   NO-A16368G ATGGAATGGA ATACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAG CACCACACCA CAGATCACTC 70
     FI-ka422 ATGGAATGGA ACACCCTTTT CTTGGTGATC TTGATCATCA TCATAAAGAG CACCACACCA CAGATCACTC 70
  DK-M.rhabdo ATGGAATGGA ATACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAG CACCACACCA CAGATCACTC 70
        DK-F1 ATGGAATGGA ATACCTTTTC CTTGGTGATC TTGATCATCA TCTTAAAGAG CACCACACCA CAGATCACTC 70
   DK-Hededam ATGGAATGTA ATACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAG CACCACACCA CAGATCACTC 70
      FR-0771 ATGGAATGGA ACACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAG CACCACACCA CAGATCACTC 70
  CH-FI262BFH ATGGAATGGA ACACTTTTTT CCTGGTGATC TTGATCATCA TCATAAAGAG CACCACACCA CAGATCACTC 70
      FR-0284 ATGGAATGTA ATACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAG CACCACACCA CAGATCACTC 70
      FR-2375 ATGGAATGGA ACACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAG CACCACACCA CAGATCACTC 70
    DK-200098 ATGGAATGGA ACACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAA CACCACACCA CAGATCACTC 70
   DK-9995144 ATGGAATGGA ACACTTTTTT CTTGGTGATC TTGATCATCA TCATAAAGAA CACCACACCA CAGATCACTC 70
      DK-5123 ATGGAATGGA ATACGTTTTT CTTAGTGATC TTGATCATCA TCATAAAGAG CTCCACACCA CAGATCACTC 70
      DK-5131 ATGGAATGGA ATACGCCCTT CTTAGTGATC TTGATCATCA TCATAAAGAG CTCCACACCA CAGATCACTC 70
      DK-2835 ATGGAATGGA ATACGCTCCT CCCAGTGATC TTGATCATCA TCATAAAGAG CTCCACACCA CAGATCACTC 70
```

```
     FI-ka422 ATGGAATGGA ACACCCTTTT CTTGGT---- GATCTTGATC ATCATCATAA AGAGCACCAC ACCACAGATC 66
  DK-M.rhabdo ATGGAATGGA ATACTTTTTT CTTGGT---- GATCTTGATC ATCATCATAA AGAGCACCAC ACCACAGATC 66
        DK-F1 ATGGAATGGA ATACCTTTTC CTTGGT---- GATCTTGATC ATCATCTTAA AGAGCACCAC ACCACAGATC 66
   DK-Hededam ATGGAATGTA ATACTTTTTT CTTGGT---- GATCTTGATC ATCATCATAA AGAGCACCAC ACCACAGATC 66
      FR-0771 ATGGAATGGA ACACTTTTTT CTTGGT---- GATCTTGATC ATCATCATAA AGAGCACCAC ACCACAGATC 66
  CH-FI262BFH ATGGAATGGA ACACTTTTTT CCTGGT---- GATCTTGATC ATCATCATAA AGAGCACCAC ACCACAGATC 66
      FR-0284 ATGGAATGTA ATACTTTTTT CTTGGT---- GATCTTGATC ATCATCATAA AGAGCACCAC ACCACAGATC 66
      FR-2375 ATGGAATGGA ACACTTTTTT CTTGGT---- GATCTTGATC ATCATCATAA AGAGCACCAC ACCACAGATC 66
    DK-200098 ATGGAATGGA ACACTTTTTT CTTGGT---- GATCTTGATC ATCATCATAA AGAACACCAC ACCACAGATC 66
   DK-9995144 ATGGAATGGA ACACTTTTTT CTTGGT---- GATCTTGATC ATCATCATAA AGAACACCAC ACCACAGATC 66
      DK-5131 ATGGAATGGA ATACGCCCTT CTTAGT---- GATCTTGATC ATCATCATAA AGAGCTCCAC ACCACAGATC 66
 DK-5131-XXXX ATGGAATGGA ATACGCCCTT CTTAGTXXXX GATCTTGATC ATCATCATAA AGAGCTCCAC ACCACAGATC 70
      DK-5123 ATGGAATGGA ATACGTTTTT CTTAGT---- GATCTTGATC ATCATCATAA AGAGCTCCAC ACCACAGATC 66
      DK-2835 ATGGAATGGA ATACGCTCCT CCCAGT---- GATCTTGATC ATCATCATAA AGAGCTCCAC ACCACAGATC 66
```

Figure 3.4: *The top figures shows the original alignment. In the bottom panel a single sequence with four inserted X's is aligned to the original alignment. This introduces a gap 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.

3.1.4 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.

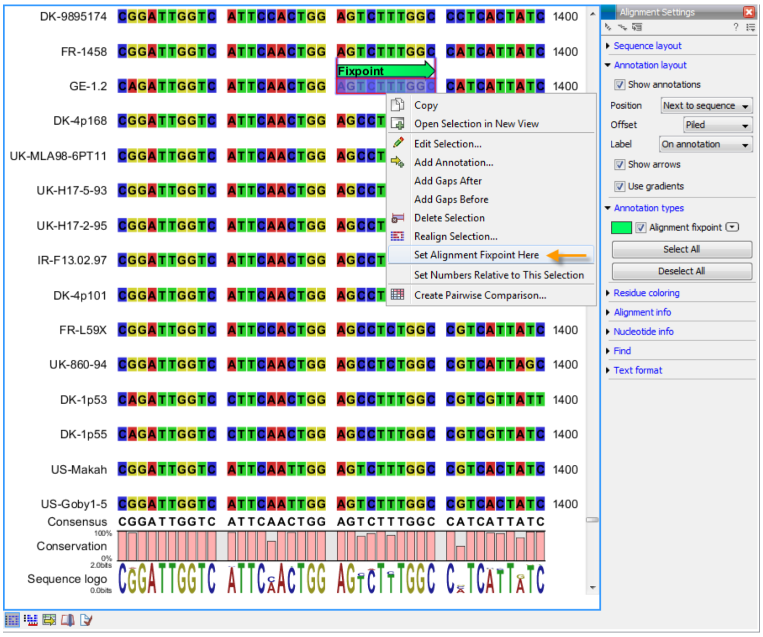
Fixpoints are added to sequences or alignments before clicking "Create alignment". To add a fixpoint, open the sequence or alignment and:

> **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 3.5). 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 3.6 the result of an alignment using fixpoints is illustrated.



☑ Show annotations
Position: Next to sequence
Offset: Piled
Label: On annotation
☑ Show arrows
☑ Use gradients
🟩 ☑ Alignment fixpoint

Figure 3.5: *Adding a fixpoint to a sequence in an existing alignment. At the top you can see a fixpoint that has already been added.*

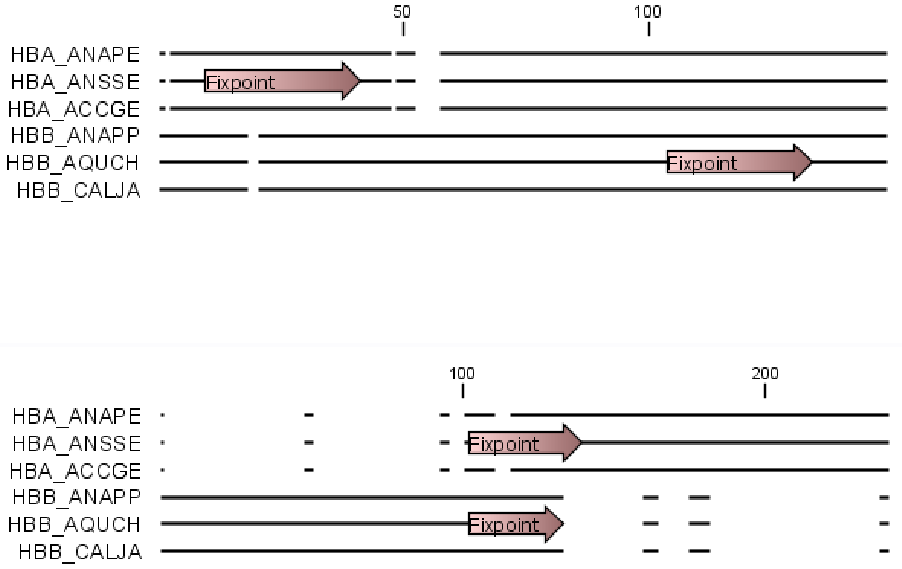


Figure 3.6: *Realigning using fixpoints. In the top view, fixpoints have been added to two of the sequences. In the view below, the alignment has been realigned using the fixpoints. The three top sequences are very similar, and therefore they follow the one sequence (number two from the top) that has a fixpoint.*

You can add multiple fixpoints, e.g. adding two fixpoints to the sequences that are aligned will force their first fixpoints to be aligned to each other, and their second fixpoints will also be aligned to each other.

Advanced use of fixpoints

Fixpoints with the same names will be aligned to each other, which gives the opportunity for great control over the alignment process. It is only necessary to change any fixpoint names in very special cases.

One example would be three sequences A, B, and C where sequences A and B has one copy of a domain while sequence C has two copies of the domain. You can now force sequence A to align to the first copy and sequence B to align to the second copy of the domains in sequence C. This is done by inserting fixpoints in sequence C for each domain, and naming them 'fp1' and 'fp2' (for example). Now, you can insert a fixpoint in each of sequences A and B, naming them 'fp1' and 'fp2', respectively. Now, when aligning the three sequences using fixpoints, sequence A will align to the first copy of the domain in sequence C, while sequence B would align to the second copy of the domain in sequence C.

You can name fixpoints by:

> **right-click the Fixpoint annotation | Edit Annotation () | type the name in the 'Name' field**

3.2 Join alignments

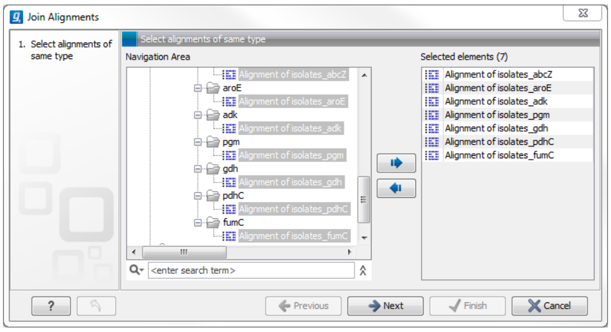
CLC Genomics Workbench can join several alignments into one. This feature can for example be used to construct "supergenes" for phylogenetic inference by joining alignments of several disjoint genes into one spliced alignment. Note, that when alignments are joined, all their annotations are carried over to the new spliced alignment.

Alignments can be joined by:

> **select alignments to join | Toolbox in the Menu Bar | Classical Sequence Analysis () | Alignments and Trees ()| Join Alignments ()**

or **select alignments to join | right-click either selected alignment | Toolbox | Classical Sequence Analysis () | Alignments and Trees ()| Join Alignments ()**

This opens the dialog shown in figure 3.7.



Join Alignments

1. Select alignments of same type

Select alignments of same type

Navigation Area

- Alignment of isolates_abcZ
- aroE
  - Alignment of isolates_aroE
- adk
  - Alignment of isolates_adk
- pgm
  - Alignment of isolates_pgm
- gdh
  - Alignment of isolates_gdh
- pdhC
  - Alignment of isolates_pdhC
- fumC
  - Alignment of isolates_fumC

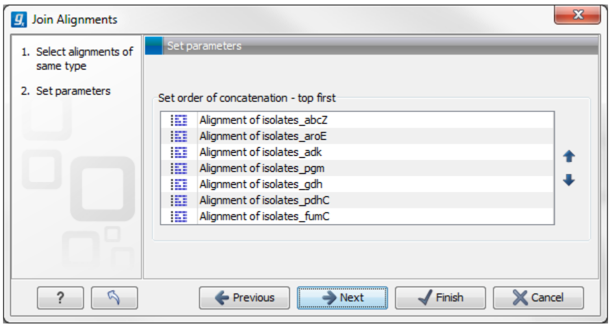
<enter search term>

Selected elements (7)

- Alignment of isolates_abcZ
- Alignment of isolates_aroE
- Alignment of isolates_adk
- Alignment of isolates_pgm
- Alignment of isolates_gdh
- Alignment of isolates_pdhC
- Alignment of isolates_fumC

Figure 3.7: *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 3.8.



Join Alignments

1. Select alignments of same type
2. Set parameters

Set parameters

Set order of concatenation - top first

- Alignment of isolates_abcZ
- Alignment of isolates_aroE
- Alignment of isolates_adk
- Alignment of isolates_pgm
- Alignment of isolates_gdh
- Alignment of isolates_pdhC
- Alignment of isolates_fumC

Figure 3.8: *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 3.9.

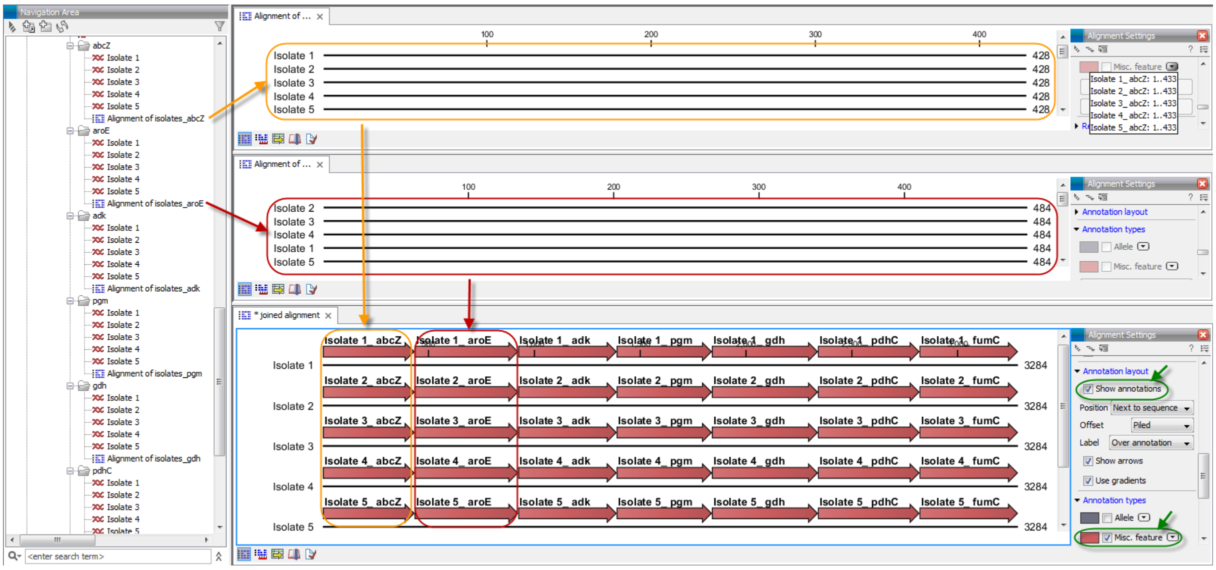


Figure 3.9: *The upper part of the figure shows two of the seven alignments for the genes "abcZ" and "aroE" respectively. Each alignment consists of sequences from one gene from five different isolates. The lower part of the figure shows the result of "Join Alignments". Seven genes have been joined to 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.*

3.2.1 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 3.9 "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.

3.3 Pairwise comparison

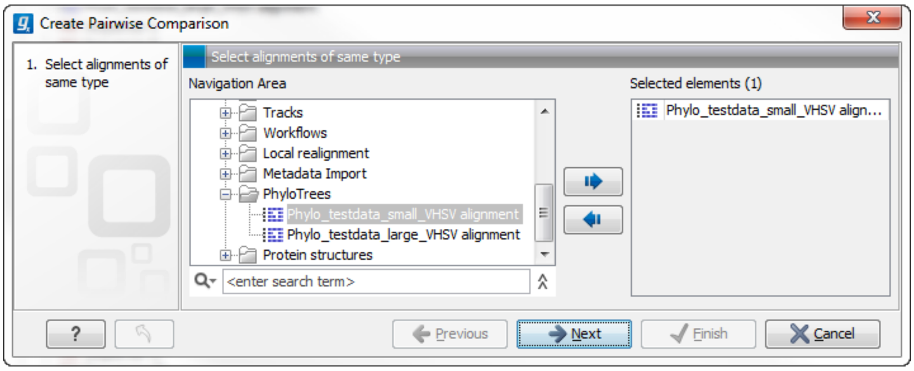
For a given set of aligned sequences it is possible to make a pairwise comparison in which each pair of sequences are compared to each other. This provides an overview of the diversity among the sequences in the alignment.

In CLC Genomics Workbench this is done by creating a comparison table:

> **Toolbox in the Menu Bar | Classical Sequence Analysis () | Alignments and Trees ()| Pairwise Comparison ()**

or **right-click alignment in Navigation Area | Toolbox | Classical Sequence Analysis () | Alignments and Trees ()| Pairwise Comparison ()**

This opens the dialog displayed in figure 3.10:



Create Pairwise Comparison

1. Select alignments of same type

Select alignments of same type

Navigation Area

Selected elements (1)

Phylo_testdata_small_VHSV align...

Figure 3.10: *Creating a pairwise comparison table.*

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.

3.3.1 Pairwise comparison on alignment selection

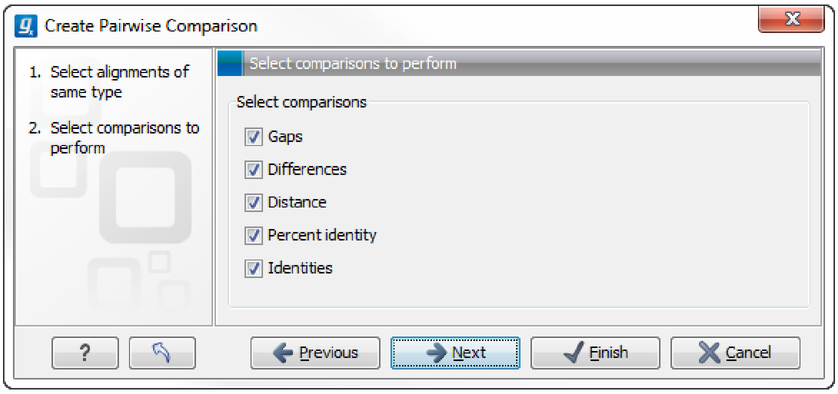
A pairwise comparison can also be performed for a selected part of an alignment:

> **right-click on an alignment selection | Pairwise Comparison ()**

This leads directly to the dialog described in the next section.

3.3.2 Pairwise comparison parameters

There are five kinds of comparison that can be made between the sequences in the alignment, as shown in figure 3.11.


Create Pairwise Comparison
1. Select alignments of same type
2. Select comparisons to perform
Select comparisons to perform
Select comparisons
?
Previous
Next
Finish
Cancel

☑ Gaps
☑ Differences
☑ Distance
☑ Percent identity
☑ Identities

Figure 3.11: *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.

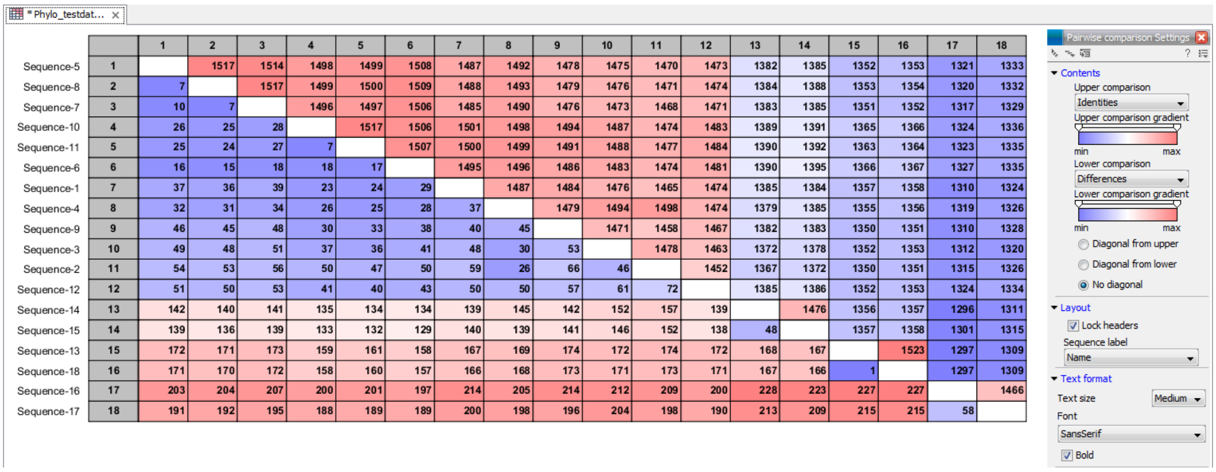
- **Differences** Calculates the number of alignment positions where one sequence is different from the other. This includes gap differences as in the Gaps comparison.

- **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.

3.3.3 The pairwise comparison table

The table shows the results of selected comparisons (see an example in figure 3.12). 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.


* Phylo_testdat...
Pairwise comparison Settings
Contents
Layout
Text format
min
max
Upper comparison gradient
Lower comparison gradient

		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	

Upper comparison: Identities
Lower comparison: Differences
○ Diagonal from upper
○ Diagonal from lower
● No diagonal
☑ Lock headers
Sequence label: Name
Text size: Medium
Font: SansSerif
☑ Bold

Figure 3.12: *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 Diagona.** 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.

Chapter 4

Create Trees

For a given set of aligned sequences (see section 3.1) 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 4.5). 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 4.5.3). 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.

- **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 4.5.3).

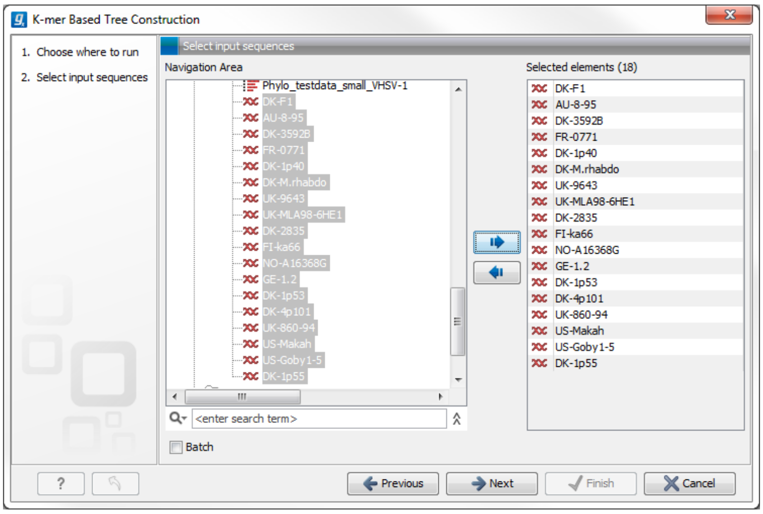
- **Maximum Likelihood Phylogeny** ( ) The most advanced and time consuming method of the three mentioned. The maximum likelihood tree estimation is performed under the assumption of one of five substitution models: the Jukes-Cantor, the Kimura 80, the HKY and the GTR (also known as the REV model) models (see section 4.4 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 4.3) in order to identify the best suitable models for creating a tree.

4.1 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 4.1):

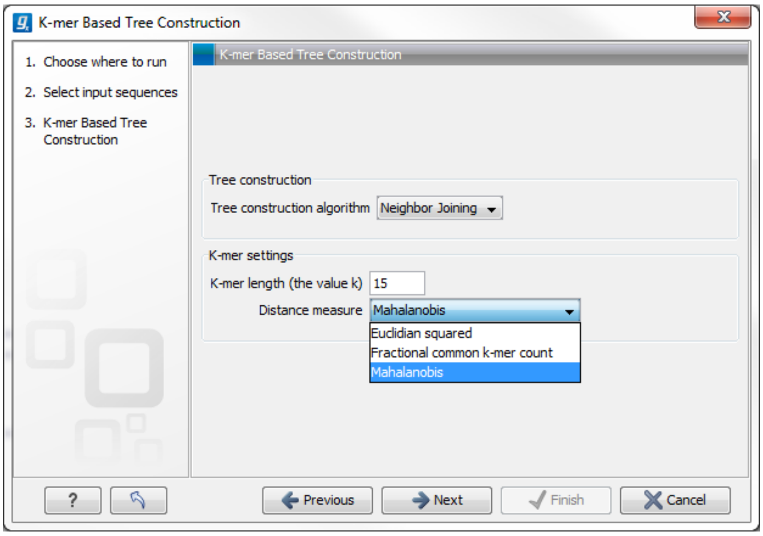


Selected elements (18)

DK-F1
AU-8-95
DK-3592B
FR-0771
DK-1p40
DK-M.rhabdo
UK-9643
UK-MLA98-6HE1
DK-2835
FI-ka66
NO-A16368G
GE-1.2
DK-1p53
DK-4p101
UK-860-94
US-Makah
US-Goby1-5
DK-1p55

Figure 4.1: *Creating a tree with K-mer based tree construction. Select sequences.*

Next, select the reconstruction method, specify the k-mer length and select a distance measure for tree construction (figure 4.2):


K-mer Based Tree Construction
1. Choose where to run
2. Select input sequences
3. K-mer Based Tree Construction
K-mer Based Tree Construction
Euclidian squared
Fractional common k-mer count
Mahalanobis
?
Previous
Next
Finish
Cancel

Tree construction

Tree construction algorithm: Neighbor Joining

K-mer settings

K-mer length (the value k): 15

Distance measure: Mahalanobis

Figure 4.2: *Creating a tree with K-mer based tree construction. Select reconstruction method, specify the k-mer length and select a distance measure.*

- **Tree construction**
  - **Tree construction algorithm** The user is asked to specify which distance-based method to use for tree reconstruction. There are two options (see section 4.5.3):
    - 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.

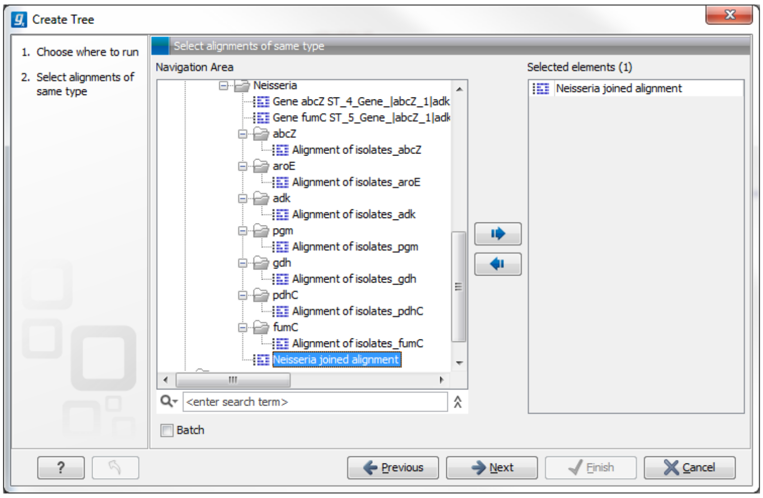
- **Distance measure** The distance measure is used to compute the distances between two counts of k-mers. Three options exist: Euclidian squared, Mahalanobis, and Fractional common K-mer count. See 4.5.2 for further details.

4.2 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 4.3:



Create Tree

1. Choose where to run
2. Select alignments of same type

Select alignments of same type

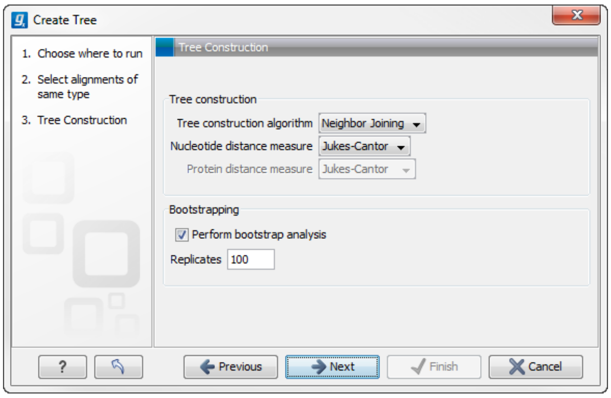
Selected elements (1): Neisseria joined alignment

☐ Batch

Navigation Area
Neisseria
Gene abcZ ST_4_Gene_|abcZ_1|adk
Gene fumC ST_5_Gene_|abcZ_1|adk
abcZ
Alignment of isolates_abcZ
aroE
Alignment of isolates_aroE
adk
Alignment of isolates_adk
pgm
Alignment of isolates_pgm
gdh
Alignment of isolates_gdh
pdhC
Alignment of isolates_pdhC
fumC
Alignment of isolates_fumC
Neisseria joined alignment
<enter search term>
?
Previous
Next
Finish
Cancel

Figure 4.3: *Creating a tree.*

If an alignment was selected before choosing the Toolbox action, this alignment is now listed in the **Selected Elements** window of the dialog. Use the arrows to add or remove elements from the **Navigation Area**. Click **Next** to adjust parameters.



Create Tree

1. Choose where to run
2. Select alignments of same type
3. Tree Construction

Tree Construction

Tree construction

Tree construction algorithm: Neighbor Joining

Nucleotide distance measure: Jukes-Cantor

Protein distance measure: Jukes-Cantor

Bootstrapping

☑ Perform bootstrap analysis

Replicates: 100

Figure 4.4: *Adjusting parameters for distance-based methods.*

Figure 4.4 shows the parameters that can be set for this distance-based tree creation:

- Tree construction (see section 4.5.3)

  - Tree construction algorithm
    - 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 4.5.5). 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 4.5.

4.3 Model Testing

As the "Model Testing" tool can help identify the best substitution model (4.5.1) 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:

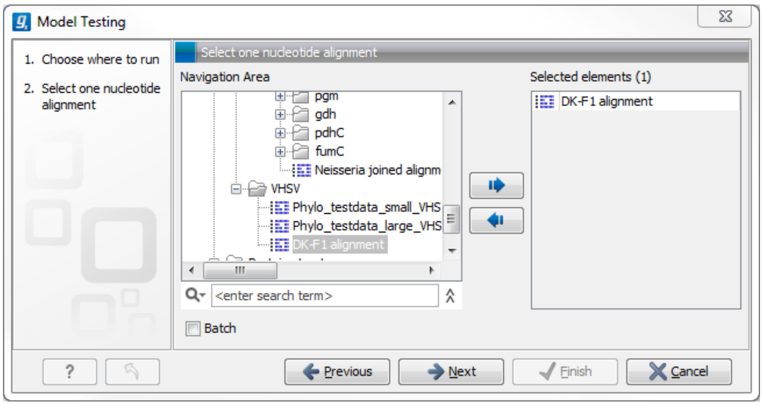
- Jukes-Cantor [Jukes and Cantor, 1969]
- Felsenstein 81 [Felsenstein, 1981]

- 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 4.5):



Model Testing

1. Choose where to run
2. Select one nucleotide alignment

Select one nucleotide alignment

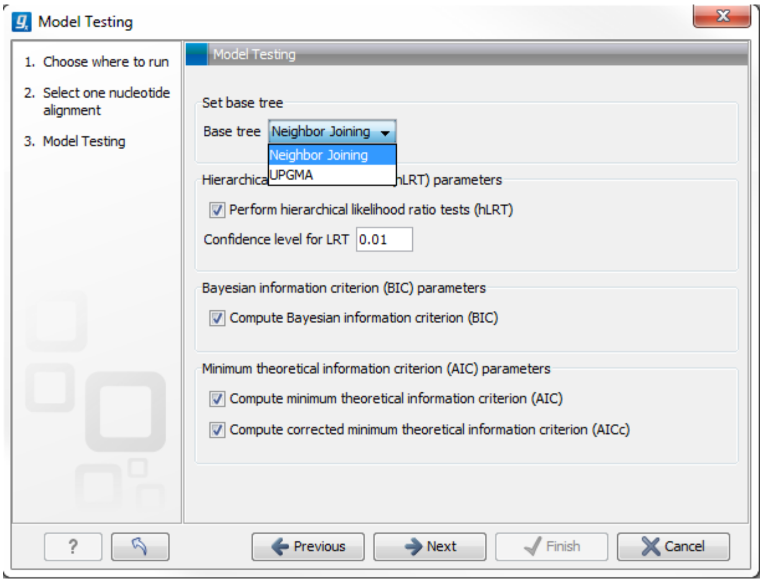
Navigation Area

Selected elements (1)

DK-F1 alignment

Figure 4.5: *Select alignment for model testing.*

Specify the parameters to be used for model testing (figure 4.6):



Model Testing

1. Choose where to run
2. Select one nucleotide alignment
3. Model Testing

Model Testing

Set base tree

Base tree: Neighbor Joining

Hierarchica hLRT) parameters

☑ Perform hierarchical likelihood ratio tests (hLRT)

Confidence level for LRT: 0.01

Bayesian information criterion (BIC) parameters

☑ Compute Bayesian information criterion (BIC)

Minimum theoretical information criterion (AIC) parameters

☑ Compute minimum theoretical information criterion (AIC)

☑ Compute corrected minimum theoretical information criterion (AICc)

Figure 4.6: *Specify parameters for model testing.*

- **Set base tree**

  Creates a base tree using either the Neighbor-Joining method or the UPGMA 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.

  - **Base tree** Two options exist. A base tree can be created automatically using the methods from the "Create Tree" tools:
    - 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.

4.4 Maximum Likelihood Phylogeny

To generate a maximum likelihood based phylogenetic tree:

Toolbox | **Classical Sequence Analysis ()** | **Alignments and Trees ()**| **Maximum Likelihood Phylogeny ()**

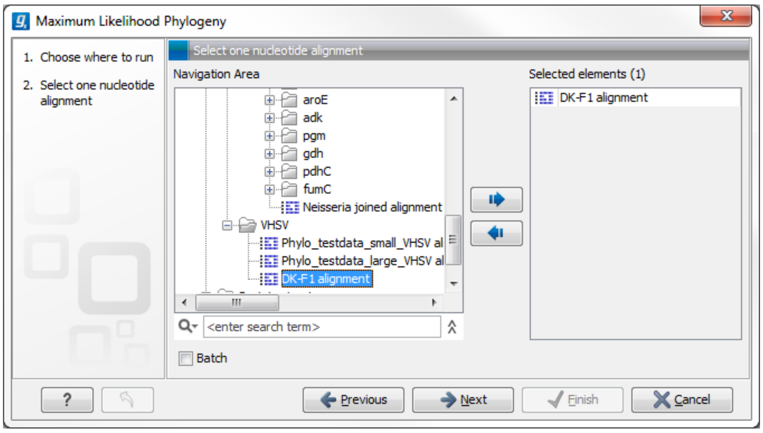
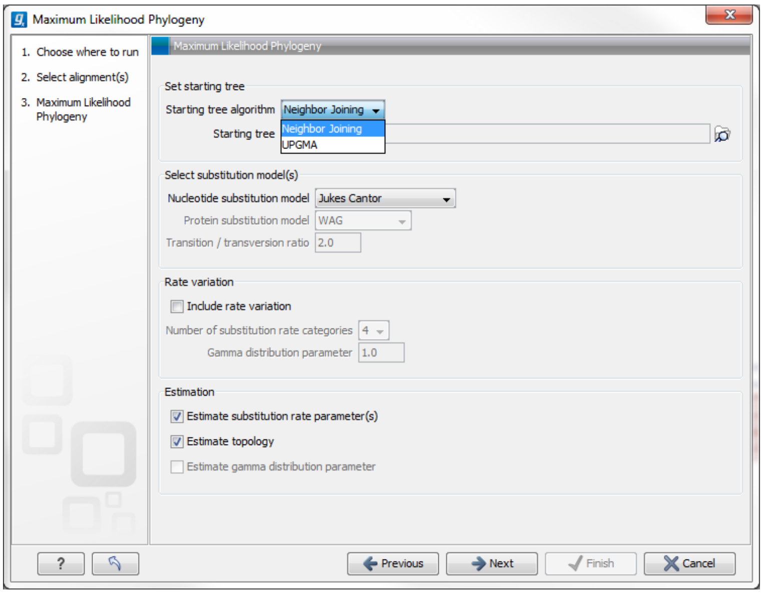

Maximum Likelihood Phylogeny
1. Choose where to run
2. Select one nucleotide alignment
Select one nucleotide alignment
Navigation Area
aroE
adk
pgm
gdh
pdhC
fumC
Neisseria joined alignment
VHSV
Phylo_testdata_small_VHSV al
Phylo_testdata_large_VHSV al
DK-F1 alignment
<enter search term>
Batch
Selected elements (1)
DK-F1 alignment
?
Previous
Next
Finish
Cancel

Figure 4.7: *Select the alingment for tree construction*



Set starting tree

Starting tree algorithm: Neighbor Joining

Starting tree:

Select substitution model(s)

Nucleotide substitution model: Jukes Cantor

Protein substitution model: WAG

Transition / transversion ratio: 2.0

Rate variation

☐ Include rate variation

Number of substitution rate categories: 4

Gamma distribution parameter: 1.0

Estimation

☑ Estimate substitution rate parameter(s)

☑ Estimate topology

☐ Estimate gamma distribution parameter

Figure 4.8: *Adjusting parameters for maximum likelihood phylogeny*

The following parameters can be set for the maximum likelihood based phylogenetic tree (see figure 4.8):

- **Set starting tree**
  - **Starting tree algorithm** Specify the method which should be used to create the initial tree. There are two possibilities:
    - Neighbor Joining
    - UPGMA
  - **Starting 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.
  - Neighbor Joining
  - UPGMA

- **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 4.5.1.

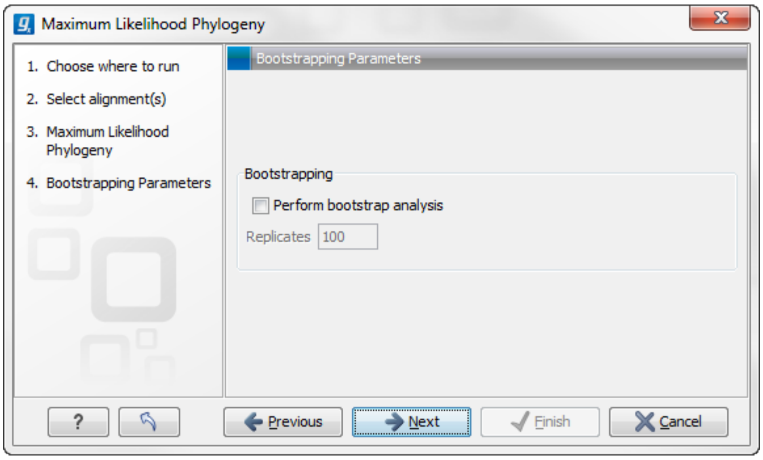
  - **Protein substitution model** CLC Genomics Workbench allows maximum likelihood tree estimation to be performed under the assumption of one of four protein substitution models:
    - Bishop-Friday [Bishop and Friday, 1985]
    - Dayhoff (PAM) [Dayhoff et al., 1978]
    - JTT [Jones et al., 1992]
    - WAG [Whelan and Goldman, 2001]

  The Bishop-Friday substitution model is similar to the Jukes-Cantor model for nucleotide sequences, i.e. it assumes equal amino acid frequencies and substitution rates. This is an unrealistic assumption and we therefore recommend using one of the remaining three models. The Dayhoff, JTT and WAG substitution models are all based on large scale experiments where amino acid frequencies and substitution rates have been estimated by aligning thousands of protein sequences. For these models, the maximum likelihood tool does not estimate parameters, but simply uses those determined from these experiments.

  - **Rate variation**
    To enable variable substitution rates among individual nucleotide sites in the alignment, select the **include rate variation** box. When selected, the discrete gamma model of Yang [Yang, 1994b] is used to model rate variation among sites. The number of categories used in the discretization of the gamma distribution as well as the gamma distribution parameter may be adjusted by the user (as the gamma distribution is restricted to have mean 1, there is only one parameter in the distribution).
  - **Estimation**
    Estimation is done according to the maximum likelihood principle, that is, a search is performed for the values of the free parameters in the model assumed that results in the highest likelihood of the observed alignment [Felsenstein, 1981]. By ticking the **estimate substitution rate parameters** box, maximum likelihood values of the free parameters in the rate matrix describing the assumed substitution model are found. If the **Estimate topology** box is selected, a search in the space of tree topologies for that which best explains the alignment is performed. If left un-ticked, the starting topology is kept fixed at that of the starting tree.

The **Estimate Gamma distribution parameter** is active if rate variation has been included in the model and in this case allows estimation of the Gamma distribution parameter to be switched on or off. If the box is left un-ticked, the value is fixed at that given in the **Rate variation** part. In the absence of rate variation estimation of substitution parameters and branch lengths are carried out according to the expectation maximization algorithm [Dempster et al., 1977]. With rate variation the maximization algorithm is performed. The topology space is searched according to the PHYML method [Guindon and Gascuel, 2003], allowing efficient search and estimation of large phylogenies. **Branch lengths are given in terms of expected numbers of substitutions per nucleotide site**.

In the next step of the wizard it is possible to perform bootstrapping (figure 4.9).



Maximum Likelihood Phylogeny

1. Choose where to run
2. Select alignment(s)
3. Maximum Likelihood Phylogeny
4. Bootstrapping Parameters

Bootstrapping Parameters

Bootstrapping

☐ Perform bootstrap analysis

Replicates 100

Figure 4.9: *Adjusting parameters for ML phylogeny*

- **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 4.5.5). 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.

4.5 Bioinformatics explained

4.5.1 Substitution models and distance estimation

When estimating the evolutionary distance between organisms, one needs a model of how frequently different mutations occur in the DNA. Such models are known as substitution models. Our Model Testing and Maximum Likelihood Phylogeny tools currently support the five nucleotide substitution models listed here:

- Jukes-Cantor [Jukes and Cantor, 1969]

- Felsenstein 81 [Felsenstein, 1981]
- Kimura 80 [Kimura, 1980]
- HKY [Hasegawa et al., 1985]
- GTR (also known as the REV model) [Yang, 1994a]

Common to all these models is that they assume mutations at different sites in the genome occur independently and that the mutations at each site follow the same common probability distribution. Thus all five models provide relative frequencies for each of the 16 possible DNA substitutions (e.g. $C \rightarrow A$, $C \rightarrow C$, $C \rightarrow G$,...).

The Jukes-Cantor and Kimura 80 models assume equal base frequencies and the HKY and GTR models allow the frequencies of the four bases to differ (they will be estimated by the observed frequencies of the bases in the alignment). In the Jukes-Cantor model all substitutions are assumed to occur at equal rates, in the Kimura 80 and HKY models transition and transversion rates are allowed to differ (substitution between two purines $(A \leftrightarrow G)$ or two pyrimidines $(C \leftrightarrow T)$ are transitions and purine - pyrimidine substitutions are transversions). The GTR model is the general time reversible model that allows all substitutions to occur at different rates. For the substitution rate matrices describing the substitution models we use the parametrization of Yang [Yang, 1994a].

For protein sequences, our Maximum Likelihood Phylogeny tool supports four substitution models:

- Bishop-Friday [Bishop and Friday, 1985]
- Dayhoff (PAM) [Dayhoff et al., 1978]
- JTT [Jones et al., 1992]
- WAG [Whelan and Goldman, 2001]

As with nucleotide substitution models, it is assumed that mutations at different sites in the genome occur independently and according to the same probability distribution.

The Bishop-Friday model assumes all amino acids occur with same frequency and that all substitutions are equally likely. This is the simplest model, but also the most unrealistic. The remaining three models use amino acid frequencies and substitution rates which have been determined from large scale experiments where huge sets of protein sequences have been aligned and rates have been estimated. These three models reflect the outcome of three different experiments. We recommend using WAG as these rates where estimated from the largest experiment.

4.5.2 K-mer based distance estimation

K-mer based distance estimation is an alternative to estimating evolutionary distance based on multiple alignments. At a high level, the distance between two sequences is defined by first collecting the set of k-mers (subsequences of length k) occuring in the two sequences. From these two sets, the evolutionary distance between the two organisms is now defined by measuring how different the two sets are. The more the two sets look alike, the smaller is the evolutionary distance. The main motivation for estimating evolutionary distance based on k-mers, is that it is computationally much faster than first constructing a multiple alignment. Experiments show that phylogenetic tree reconstruction using k-mer based distances can produce results comparable to the slower multiple alignment based methods [Blaisdell, 1989].

All of the k-mer based distance measures completely ignores the ordering of the k-mers inside the input sequences. Hence, if the selected k value (the length of the sequences) is too small, very distantly related organisms may be assigned a small evolutionary distance (in the extreme case where k is 1, two organisms will be treated as being identical if the frequency of each nucleotide/amino-acid is the same in the two corresponding sequences). In the other extreme, the k-mers should have a length (k) that is somewhat below the average distance between mismatches if the input sequences were aligned (in the extreme case of k=the length of the sequences, two organisms have a maximum distance if they are not identical). Thus the selected k value should not be too large and not too small. A general rule of thumb is to only use k-mer based distance estimation for organisms that are not too distantly related.

Formal definition of distance. In the following, we give a more formal definition of the three supported distance measures: Euclidian-squared, Mahalanobis and Fractional common k-mer count. For all three, we first associate a point $p(s)$ to every input sequence s. Each point $p(s)$ has one coordinate for every possible length k sequence (e.g. if s represent nucleotide sequences, then $p(s)$ has 4^k coordinates). The coordinate corresponding to a length k sequence x has the value: ''number of times x occurs as a subsequence in s''. Now for two sequences s_1 and s_2, their evolutionary distance is defined as follows:

- **Euclidian squared**: For this measure, the distance is simply defined as the (squared Euclidian) distance between the two points $p(s_1)$ and $p(s_2)$, i.e.

  $$\text{dist}(s_1, s_2) = \sum_i (p(s_1)_i - p(s_2)_i)^2.$$

- **Mahalanobis**: This measure is essentially a fine-tuned version of the Euclidian squared distance measure. Here all the counts $p(s_j)_i$ are ''normalized'' by dividing with the standard deviation σ_j of the count for the k-mer. The revised formula thus becomes:

  $$\text{dist}(s_1, s_2) = \sum_i (p(s_1)_i/\sigma_i - p(s_2)_i/\sigma_i)^2.$$

  Here the standard deviations can be computed directly from a set of equilibrium frequencies for the different bases, see [Gentleman and Mullin, 1989].

- **Fractional common k-mer count**: For the last measure, the distance is computed based on the minimum count of every k-mer in the two sequences, thus if two sequences are very different, the minimums will all be small. The formula is as follows:

  $$\text{dist}(s_1, s_2) = \log(0.1 + \sum_i (\min(p(s_1)_i, p(s_2)_i)/(\min(n, m) - k + 1))).$$

  Here n is the length of s_1 and m is the length of s_2. This method has been described in [Edgar, 2004].

In experiments performed in [Höhl et al., 2007], the Mahalanobis distance measure seemed to be the best performing of the three supported measures.

4.5.3 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 4.5.1). 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 Phylogeny Module 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 neighbour 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.

4.5.4 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 4.3) 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 neighbour-joining to produce a tree based on a multiple alignment.

4.5.5 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.

Chapter 5

Tree Settings

The Tree Settings Side Panel found in the left side of the view area can be used to adjust the tree layout and to visualize metadata that is associated with the tree nodes.

The preferred tree layout settings (user defined tree settings) can be saved and applied via the top right **Save Tree Settings** (figure 5.1). Settings can either be saved **For This Tree Only** or for all saved phylogenetic trees (**For Tree View in General**). The fist option will save the layout of the tree for that tree only and it ensures that the layout is preserved even if it is exported and opened by a different user. The second option stores the layout globally in the Workbench and makes it available to other trees through the **Apply Saved Settings** option.

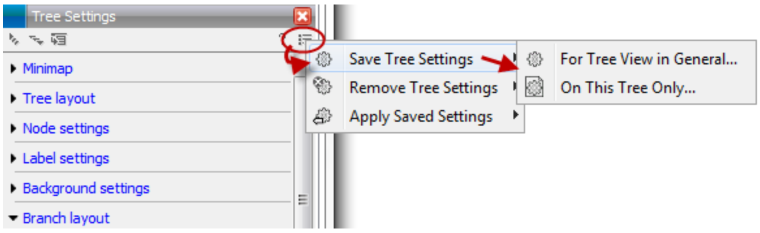


Figure 5.1: *Save, remove or apply preferred layout settings.*

The **Tree Settings** have eight different categories:

- Minimap
- Tree layout
- Node settings
- Label settings
- Background settings
- Branch layout
- Bootstrap settings
- Metadata

5.1 Minimap

The Minimap is a navigation tool that shows a small version of the tree. A grey square indicates the specific part of the tree that is visible in the View Area (figure 5.2). To navigate the tree using the Minimap, click on the Minimap with the mouse and move the grey square around within the Minimap.

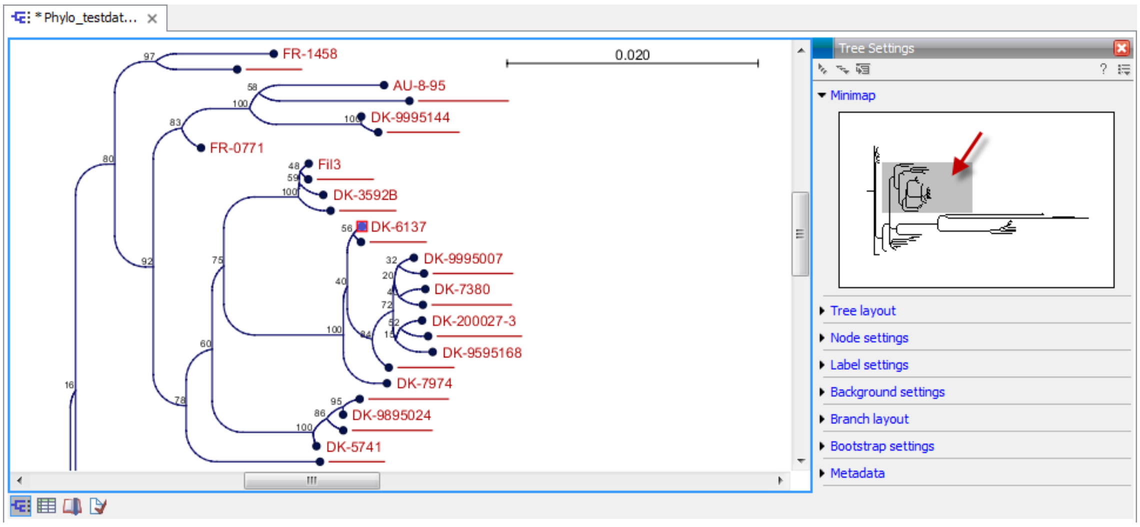
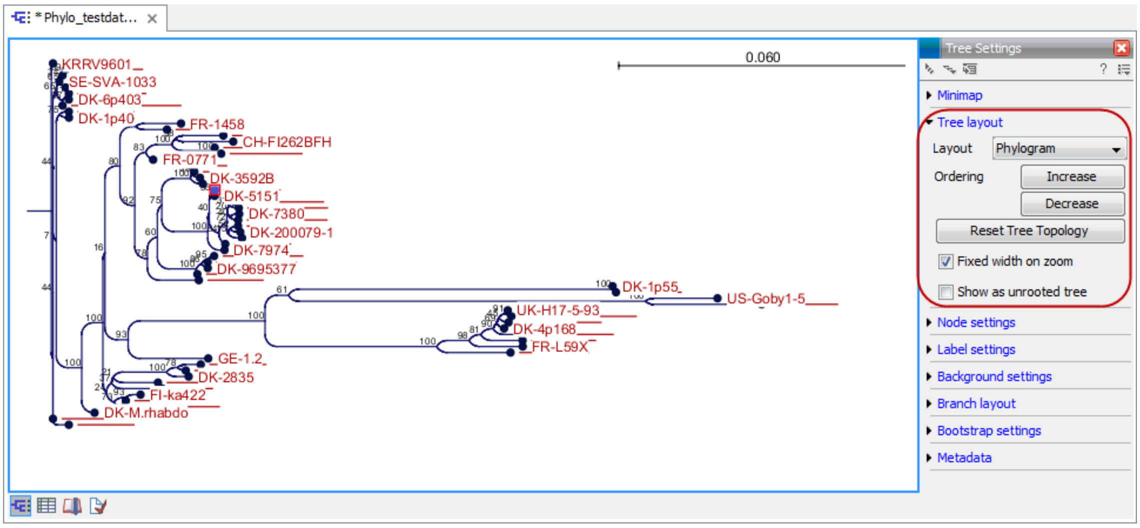


Figure 5.2: *Visualization of a phylogenetic tree. The grey square in the Minimap shows the part of the tree that is shown in the View Area.*

5.2 Tree layout

The **Tree Layout** can be adjusted in the Side Panel (figure 5.3).



Layout Phylogram

Ordering

☑ Fixed width on zoom

□ Show as unrooted tree

Figure 5.3: *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 the overall outline of the five layout types: Phylogram, Cladogram, Circular Phylogram, Circular Cladogram or Radial.

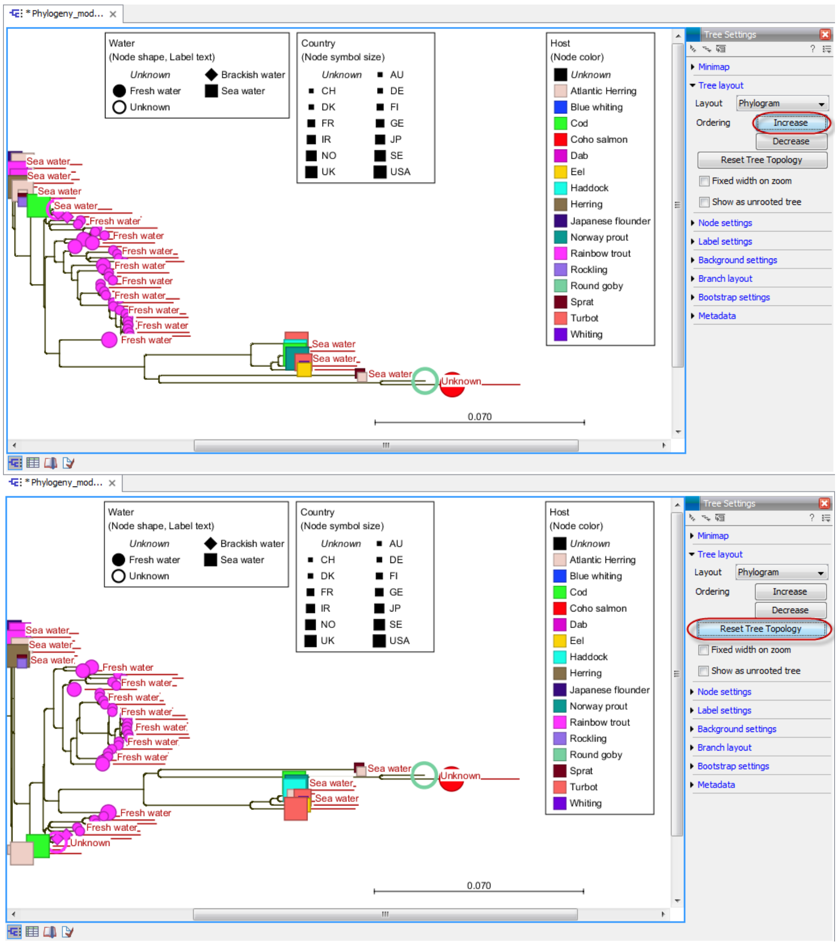
- **Phylogram** is a rooted tree where the edges have "lengths", usually proportional to the inferred amount of evolutionary change to have occurred along each branch.
- **Cladogram** is a rooted tree without branch lengths which is useful for visualizing the topology of trees.
- **Circular Phylogram** 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 5.4) or **Decreasing**.

- **Reset Tree Topology** Resets to the default tree topology and node order (see figure 5.4).

- **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.



Tree Settings

Minimap

Tree layout

Layout: Phylogram

Ordering: Increase / Decrease

Reset Tree Topology

□ Fixed width on zoom

□ Show as unrooted tree

Node settings

Label settings

Background settings

Branch layout

Bootstrap settings

Metadata

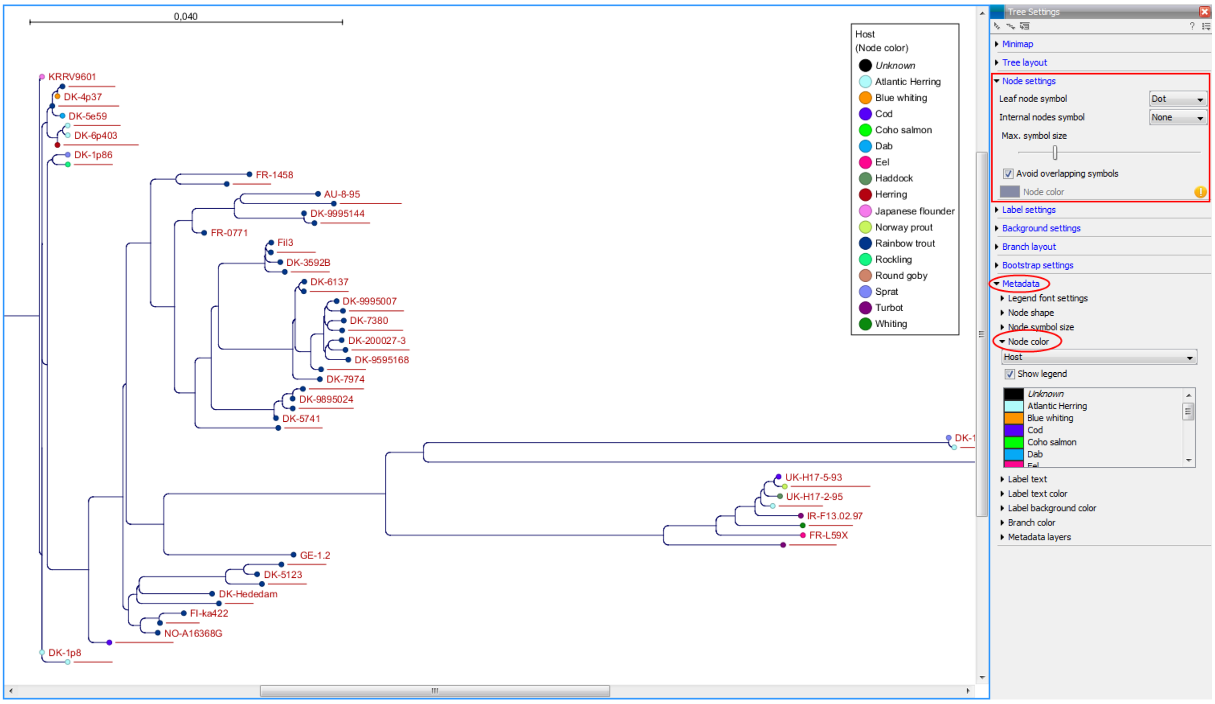
Figure 5.4: *The tree layout can be adjusted in the Side Panel. The top part of the figure shows a tree with increasing node order. In the bottom part of the figure the tree has been reverted to the original tree topology.*

5.3 Node settings

The nodes can be manipulated in several ways. This is relevant when visualizing the associated metadata.

- **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 5.5.



Leaf node symbol: Dot

Internal nodes symbol: None

Max. symbol size

☑ Avoid overlapping symbols

Node color

Node color: Host

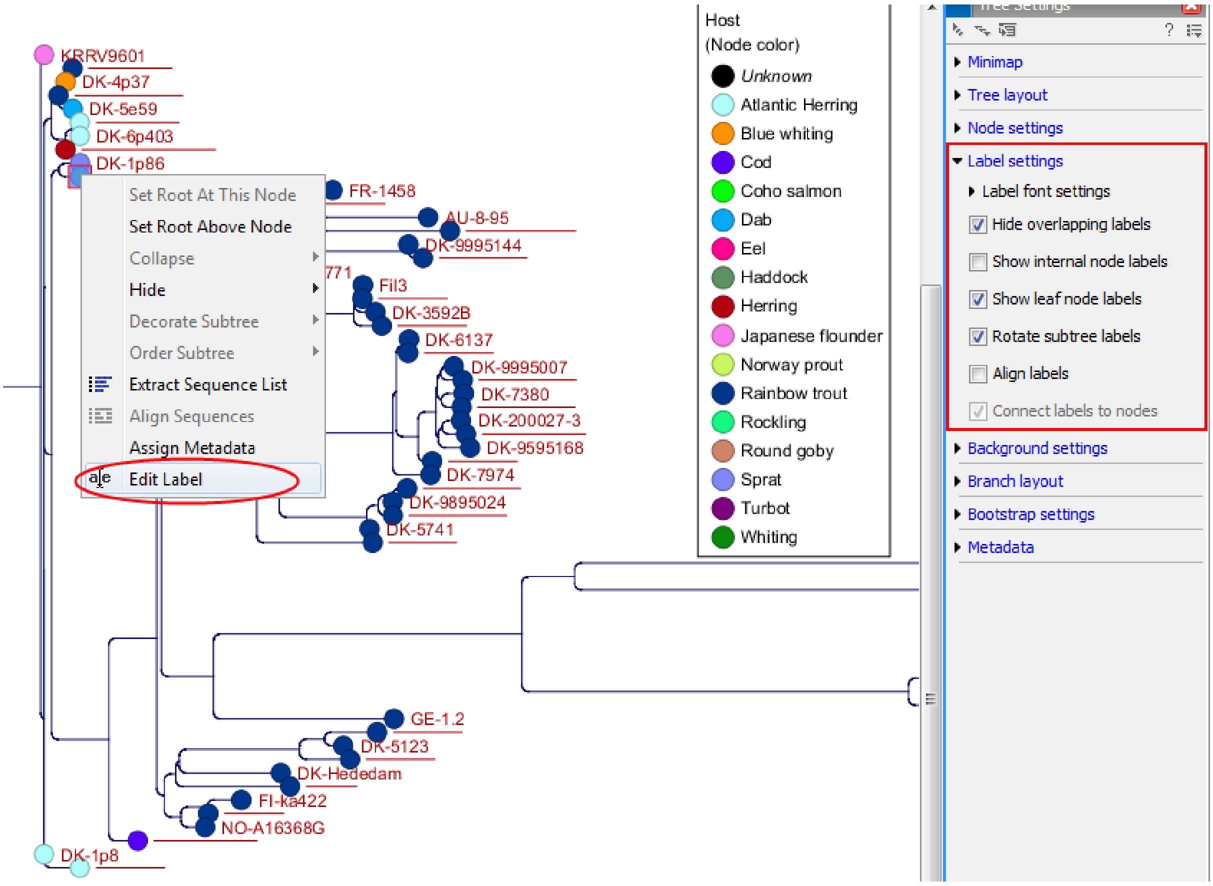
☑ Show legend

Figure 5.5: *The Node Layout settings. Node color is specified by metadata and is therefore inactive in this example.*

5.4 Label settings

- **Label font settings** Can be used to specify/adjust font type, size and typography (Bold, Italic or normal).
- **Hide overlapping labels** Disable automatic hiding of overlapping labels and display all labels even if they overlap.

- **Show internal node labels** Labels for internal nodes of the tree (if any) can be displayed. Please note that subtrees and nodes can be labeled with a custom text. This is done by right clicking the node and selecting **Edit Label** (see figure 5.6).
- **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 5.9).
- **Align labels** Align labels to the node furthest from the center of the tree so that all labels are positioned next to each other. The exact behavior depends on the selected tree layout.
- **Connect labels to nodes** Adds a thin line from the leaf node to the aligned label. Only possible when Align labels option is selected.



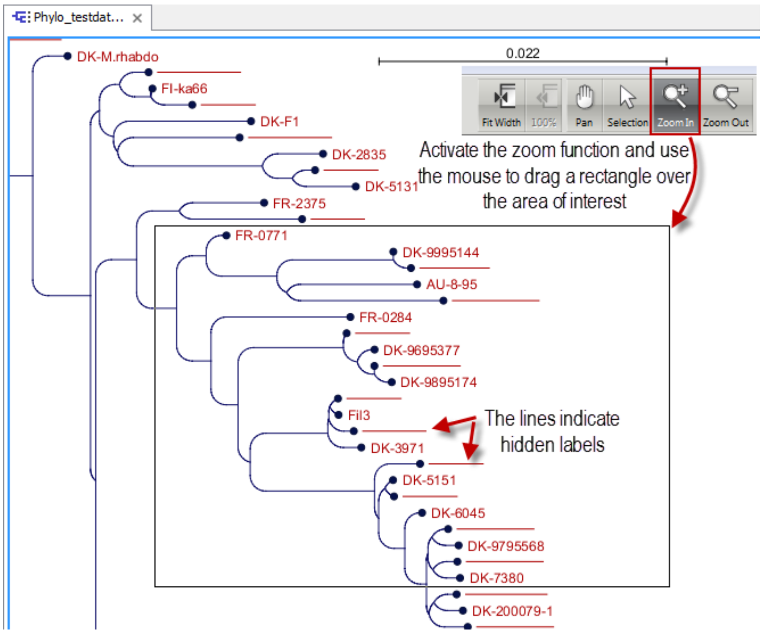
☑ Hide overlapping labels
□ Show internal node labels
☑ Show leaf node labels
☑ Rotate subtree labels
□ Align labels
☑ Connect labels to nodes

Figure 5.6: *"Edit label" in the right click menu can be used to customize the label text. The way node labels are displayed can be controlled through the labels settings in the right side panel.*

When working with big trees there is typically not enough space to show all labels. As illustrated in figure 5.6, only some of the labels are shown. The hidden labels are illustrated with thin horizontal lines (figure 5.7).

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 5.7 and figure 5.8). 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 5.5).

Note! When working with a tree with hidden labels, it is possible to make the hidden label text appear by moving the mouse over the node with the hidden label.



Activate the zoom function and use the mouse to drag a rectangle over the area of interest

The lines indicate hidden labels

Figure 5.7: *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.*

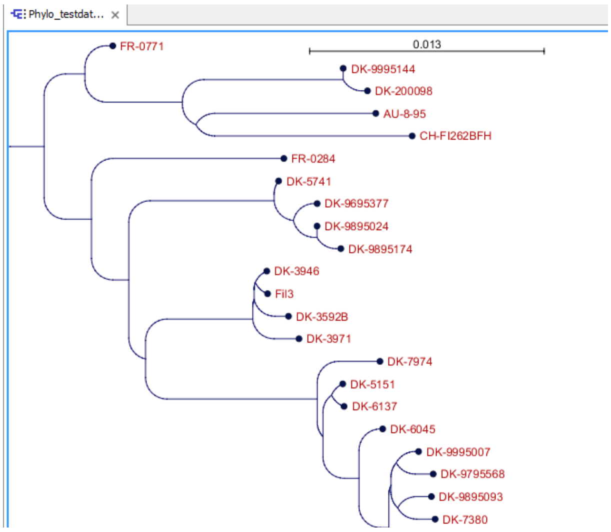


Figure 5.8: *After zooming in on a region of interest more labels become visible. In this example all labels are now visible.*

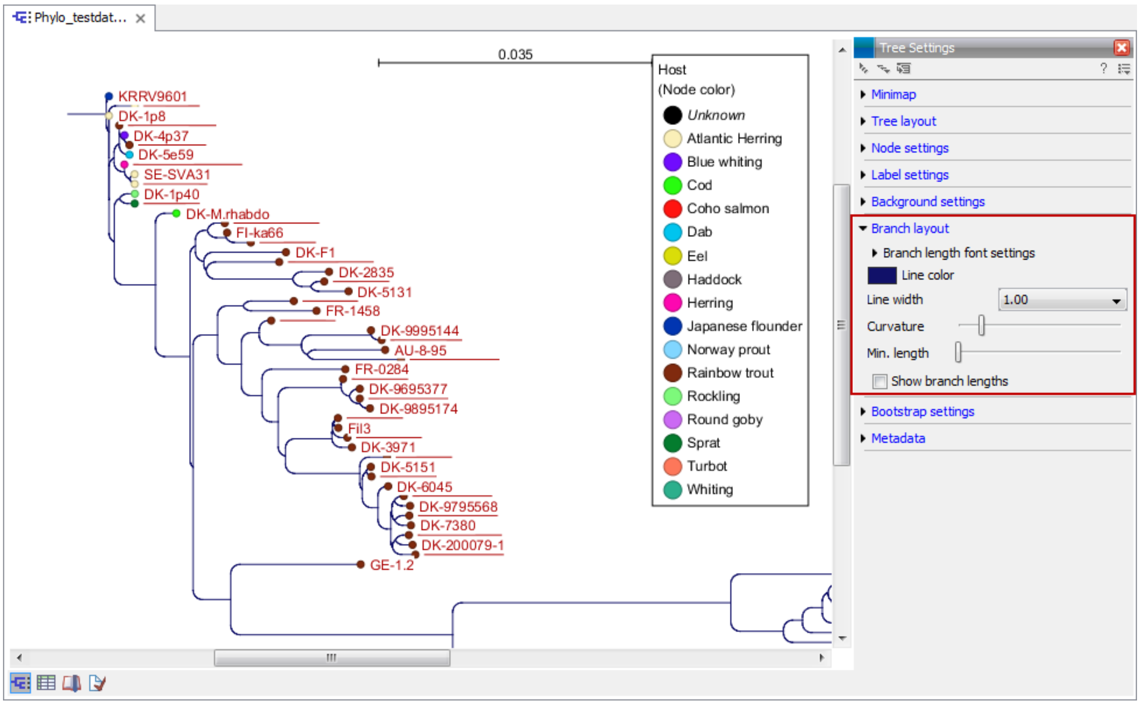
5.5 Background settings

- **Show label background** Show a background color for each label. Once ticked, it is possible to specify whether to use a fixed color or to use the color that is associated with the selected metadata category.

5.6 Branch layout

- **Branch length font settings** Specify/adjust font type, size and typography (Bold, Italic or normal).
- **Line color** Select the default line color.
- **Line width** Select the width of branches (1.0-3.0 pixels).
- **Curvature** Adjust the degree of branch curvature to get branches with round corners.
- **Min. length** Select a minimum branch length. This option can be used to prevent nodes connected with a short branch to cluster at the parent node.
- **Show branch lengths** Show or hide the branch lengths.

The branch layout settings in the Side Panel are shown in figure 5.9.



Line width 1.00

□ Show branch lengths

Figure 5.9: *Branch Layout settings.*

5.7 Bootstrap settings

Bootstrap values can be shown on the internal nodes. The bootstrap values are shown in percent and can be interpreted as confidence levels where a bootstrap value close to 100 indicate a clade, which is strongly supported by the data from which the tree was reconstructed. Bootstrap values are useful for identifying clades in the tree where the topology (and branch lengths) should not be trusted.

- **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.

5.8 Metadata

Metadata associated with a phylogenetic tree (described in detail in section 6) can be visualized in a number of different ways:

- **Node shape** Different node shapes are available to visualize metadata.
- **Node symbol size** Change the node symbol size to visualize metadata.
- **Node color** Change the node color to visualize metadata.
- **Label text** The metadata can be shown directly as text labels as shown in figure 5.10.
- **Label text color** The label text can be colored and used to visualize metadata (see figure 5.10).
- **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 5.5).

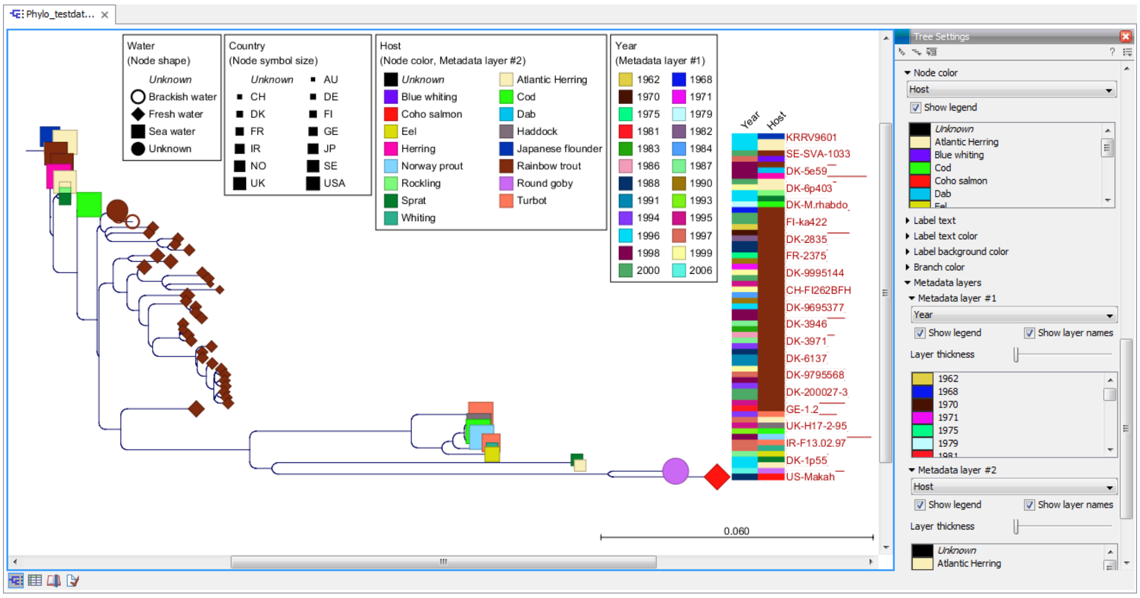


Figure 5.10: *Different types of metadata kan be visualized by adjusting node size, shape, and color. Two color-code metadata layers (Year and Host) are shown in the right side of the tree.*

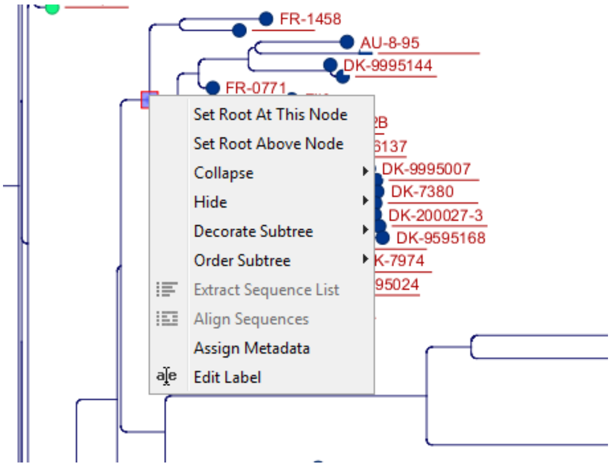


Figure 5.11: *The right click menu that appears when right clicking on a node.*

5.9 Node right click menu

Additional options for layout and extraction of subtree data are available when right clicking the nodes (figure 5.11):

- **Set Root At This Node** Re-root the tree using the selected node as root. Please note that re-rooting will change the tree topology.
- **Set Root Above Node** Re-root the tree by inserting a node between the selected node and its parent. Useful for rooting trees using an outgroup.
- **Collapse** Branches associated with a selected node can be collapsed with or without the associated labels. Collapsed branches can be uncollapsed using the *Uncollapse* option in the same menu.
- **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 5.12). When hiding nodes, a new button appears labeled "Show X hidden nodes" in the Side Panel under "Tree Layout" (figure 5.13). When pressing this button, all hidden nodes are shown again.

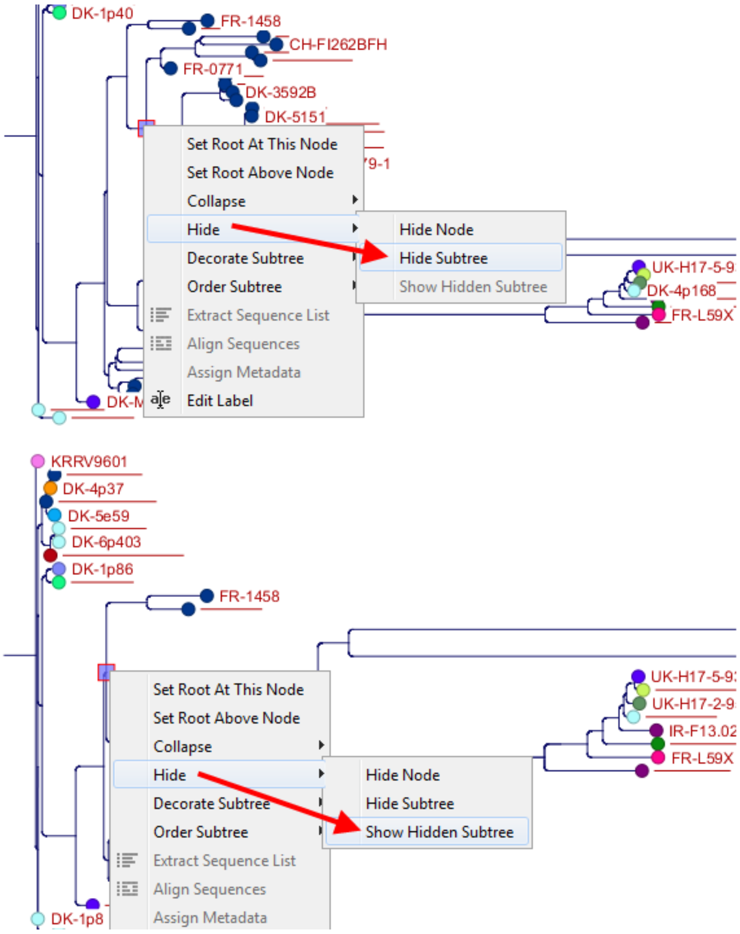
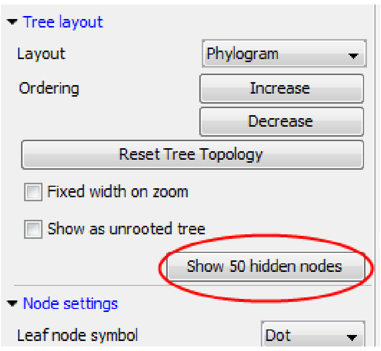


Figure 5.12: *A subtree can be hidden by selecting "Hide Subtree" and is shown again when selecting "Show Hidden Subtree" on a parent node.*



Tree layout

Layout: Phylogram

Ordering: Increase | Decrease

Reset Tree Topology

□ Fixed width on zoom

□ Show as unrooted tree

Show 50 hidden nodes

Node settings

Leaf node symbol: Dot

Figure 5.13: *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.*

- **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**

Chapter 6

Metadata and Phylogenetic Trees

When a tree is reconstructed, some mandatory metadata will be added to nodes in the tree. These metadata are special in the sense that the tree viewer has specialized features for visualizing the data and some of them cannot be edited. The mandatory metadata include:

- **Node name** The node name.
- **Branch length** The length of the branch, which connects a node to the parent node.
- **Bootstrap value** The bootstrap value for internal nodes.
- **Size** The length of the sequence which corresponds to each leaf node. This only applies to leaf nodes.
- **Start of sequence** The first 50bp of the sequence corresponding to each leaf node.

To view metadata associated with a phylogenetic tree, click on the table icon (▦) at the bottom of the tree. If you hold down the Ctrl key (or (⌘) on Mac) while clicking on the table icon (▦), you will be able to see both the tree and the table in a split view (figure 6.1).

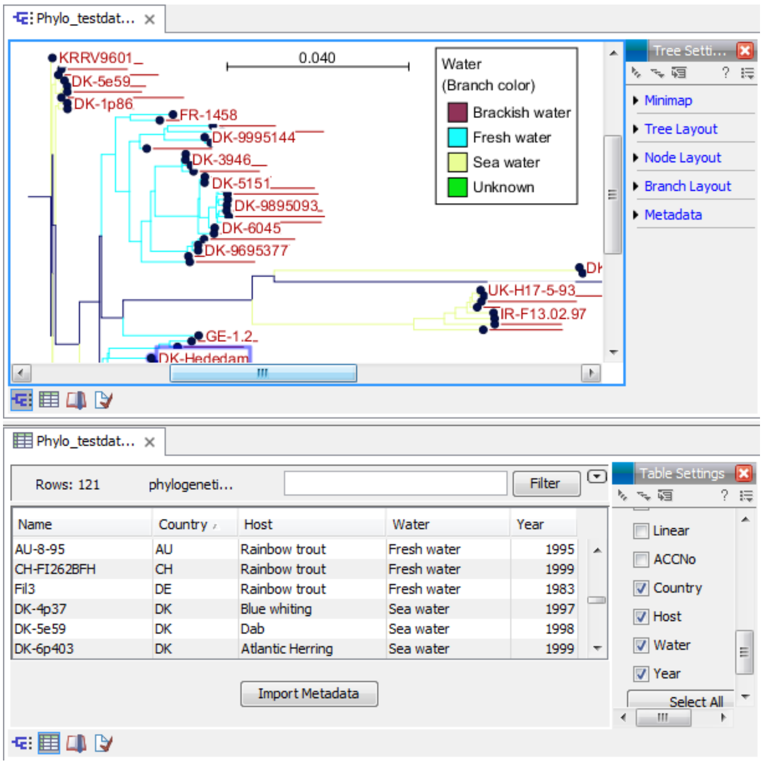
Additional metadata can be associated with a tree by clicking the **Import Metadata** button. This will open up the dialog shown in figure 6.2.

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 6.2. 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.

6.1 Table Settings and Filtering

How to use the metadata table (see figure 6.3):



Water
(Branch color)
▨ Brackish water
▨ Fresh water
▨ Sea water
▨ Unknown

Rows: 121 phylogeneti...

Name	Country	Host	Water	Year
AU-8-95	AU	Rainbow trout	Fresh water	1995
CH-FI262BFH	CH	Rainbow trout	Fresh water	1999
Fil3	DE	Rainbow trout	Fresh water	1983
DK-4p37	DK	Blue whiting	Sea water	1997
DK-5e59	DK	Dab	Sea water	1998
DK-6p403	DK	Atlantic Herring	Sea water	1999

□ Linear
□ ACCNo
☑ Country
☑ Host
☑ Water
☑ Year

Figure 6.1: *Tabular metadata that is associated with an existing tree shown in a split view.*

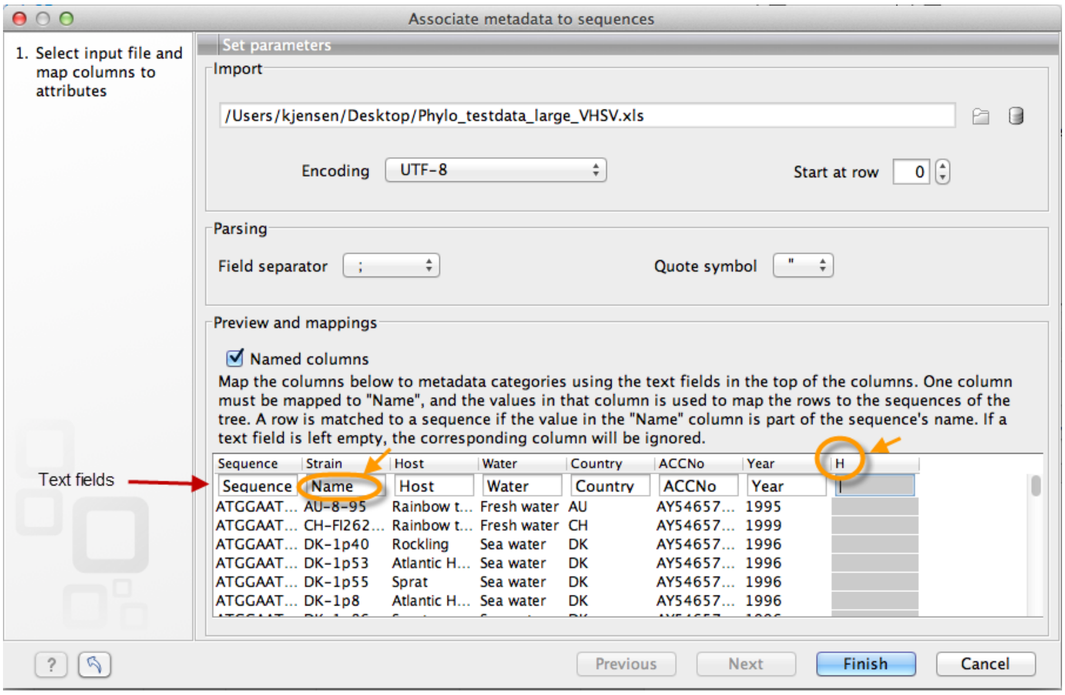
- **Column width** The column width can be adjusted in two ways; *Manually* or *Automatically*.
- **Show column** Selects which metadata categories that are shown in the table layout.
- **Filtering Metadata information** Metadata information in a table can be filtered by a simple- or advanced mode (this is described in the CLC Genomics Workbench manual, Appendix D, Filtering tables).

6.2 Add or modify metadata

It is possible to add and modify metadata from both the tree view and the table view.

Metadata can be added and edited in the metadata table by using the following right click options (see figure 6.4):

- **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.



/Users/kjensen/Desktop/Phylo_testdata_large_VHSV.xls

Encoding: UTF-8 Start at row: 0

Field separator: ; Quote symbol: "

☑ Named columns

Map the columns below to metadata categories using the text fields in the top of the columns. One column must be mapped to "Name", and the values in that column is used to map the rows to the sequences of the tree. A row is matched to a sequence if the value in the "Name" column is part of the sequence's name. If a text field is left empty, the corresponding column will be ignored.

Sequence	Strain	Host	Water	Country	ACCNo	Year	H
Sequence	Name	Host	Water	Country	ACCNo	Year	
ATGGAAT...	AU-8-95	Rainbow t...	Fresh water	AU	AY54657...	1995	
ATGGAAT...	CH-FI262...	Rainbow t...	Fresh water	CH	AY54657...	1999	
ATGGAAT...	DK-1p40	Rockling	Sea water	DK	AY54657...	1996	
ATGGAAT...	DK-1p53	Atlantic H...	Sea water	DK	AY54657...	1996	
ATGGAAT...	DK-1p55	Sprat	Sea water	DK	AY54657...	1996	
ATGGAAT...	DK-1p8	Atlantic H...	Sea water	DK	AY54657...	1996	

Figure 6.2: *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.*

  - 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*".

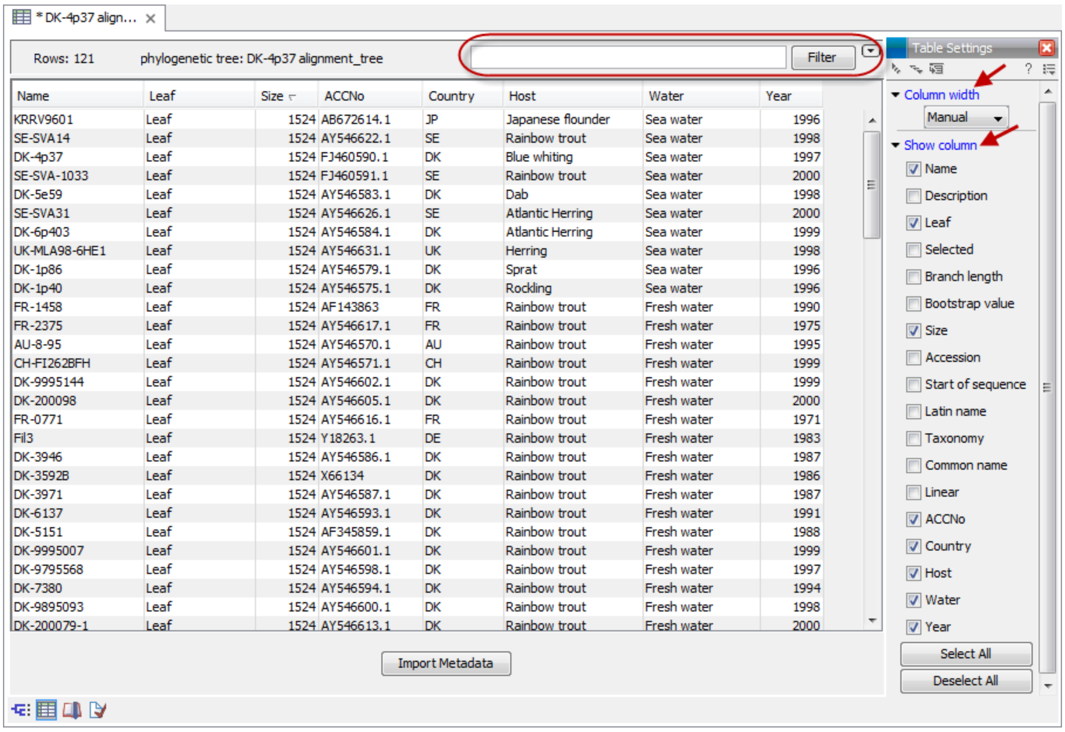
- **Edit "column header"** To modify existing metadata point, right click on a cell in the table and select the "Edit *column header*" (see an example in figure 6.5). To edit multiple entries at once, select multiple rows in the table, right click a selected cell in the column you want to edit and choose "Edit *column header*". This will change values in all selected rows in the column that was clicked.

6.3 Selection of specific nodes

Selection of nodes in a tree is automatically synchronized to the metadata table and the other way around. Nodes in a tree can be selected in three ways:

- *Selection of a single node* Click once on a single node. Additional nodes can be added by holding down Ctrl (or (⌘) for Mac) and clicking on them (see figure 6.6).

- *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.



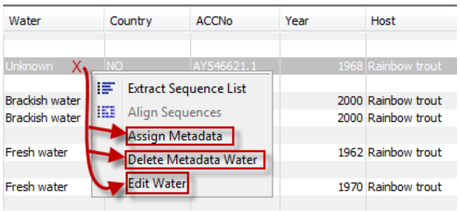
Rows: 121 phylogenetic tree: DK-4p37 alignment_tree

Name	Leaf	Size	ACCNo	Country	Host	Water	Year
KRRV9601	Leaf	1524	AB672614.1	JP	Japanese flounder	Sea water	1996
SE-SVA14	Leaf	1524	AY546622.1	SE	Rainbow trout	Sea water	1998
DK-4p37	Leaf	1524	FJ460590.1	DK	Blue whiting	Sea water	1997
SE-SVA-1033	Leaf	1524	FJ460591.1	SE	Rainbow trout	Sea water	2000
DK-5e59	Leaf	1524	AY546583.1	DK	Dab	Sea water	1998
SE-SVA31	Leaf	1524	AY546626.1	SE	Atlantic Herring	Sea water	2000
DK-6p403	Leaf	1524	AY546584.1	DK	Atlantic Herring	Sea water	1999
UK-MLA98-6HE1	Leaf	1524	AY546631.1	UK	Herring	Sea water	1998
DK-1p86	Leaf	1524	AY546579.1	DK	Sprat	Sea water	1996
DK-1p40	Leaf	1524	AY546575.1	DK	Rockling	Sea water	1996
FR-1458	Leaf	1524	AF143863	FR	Rainbow trout	Fresh water	1990
FR-2375	Leaf	1524	AY546617.1	FR	Rainbow trout	Fresh water	1975
AU-8-95	Leaf	1524	AY546570.1	AU	Rainbow trout	Fresh water	1995
CH-FI262BFH	Leaf	1524	AY546571.1	CH	Rainbow trout	Fresh water	1999
DK-9995144	Leaf	1524	AY546602.1	DK	Rainbow trout	Fresh water	1999
DK-200098	Leaf	1524	AY546605.1	DK	Rainbow trout	Fresh water	2000
FR-0771	Leaf	1524	AY546616.1	FR	Rainbow trout	Fresh water	1971
Fil3	Leaf	1524	Y18263.1	DE	Rainbow trout	Fresh water	1983
DK-3946	Leaf	1524	AY546586.1	DK	Rainbow trout	Fresh water	1987
DK-3592B	Leaf	1524	X66134	DK	Rainbow trout	Fresh water	1986
DK-3971	Leaf	1524	AY546587.1	DK	Rainbow trout	Fresh water	1987
DK-6137	Leaf	1524	AY546593.1	DK	Rainbow trout	Fresh water	1991
DK-5151	Leaf	1524	AF345859.1	DK	Rainbow trout	Fresh water	1988
DK-9995007	Leaf	1524	AY546601.1	DK	Rainbow trout	Fresh water	1999
DK-9795568	Leaf	1524	AY546598.1	DK	Rainbow trout	Fresh water	1997
DK-7380	Leaf	1524	AY546594.1	DK	Rainbow trout	Fresh water	1994
DK-9895093	Leaf	1524	AY546600.1	DK	Rainbow trout	Fresh water	1998
DK-200079-1	Leaf	1524	AY546613.1	DK	Rainbow trout	Fresh water	2000

Column width: Manual

Show column
☑ Name
☐ Description
☑ Leaf
☐ Selected
☐ Branch length
☐ Bootstrap value
☑ Size
☐ Accession
☐ Start of sequence
☐ Latin name
☐ Taxonomy
☐ Common name
☐ Linear
☑ ACCNo
☑ Country
☑ Host
☑ Water
☑ Year

Figure 6.3: *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).*



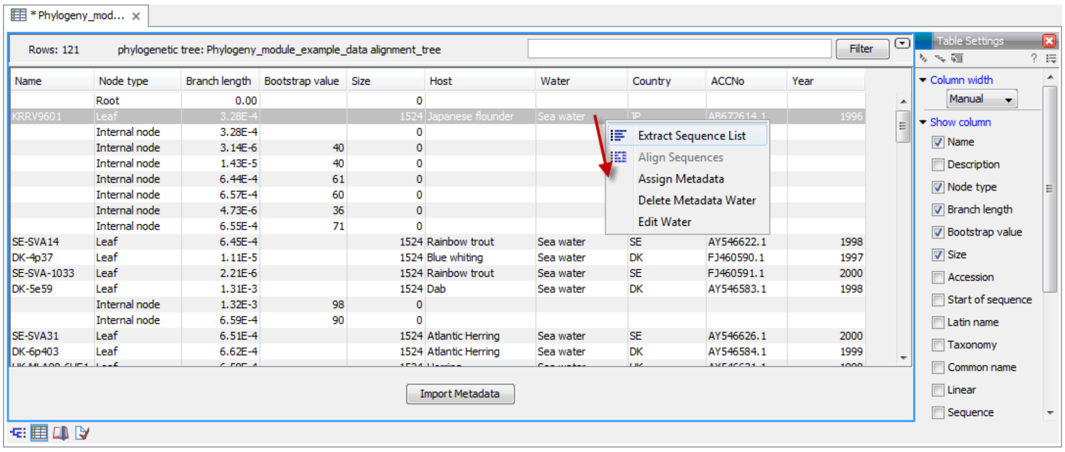
Water	Country	ACCNo	Year	Host
Unknown	NO	AY546621.1	1968	Rainbow trout
Brackish water			2000	Rainbow trout
Brackish water			2000	Rainbow trout
Fresh water			1962	Rainbow trout
Fresh water			1970	Rainbow trout

Figure 6.4: *Right click options in the metadata table.*

- *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 3). Next, change relevant option in the multiple alignment wizard that pops up



* Phylogeny_mod...

Rows: 121 phylogenetic tree: Phylogeny_module_example_data alignment_tree

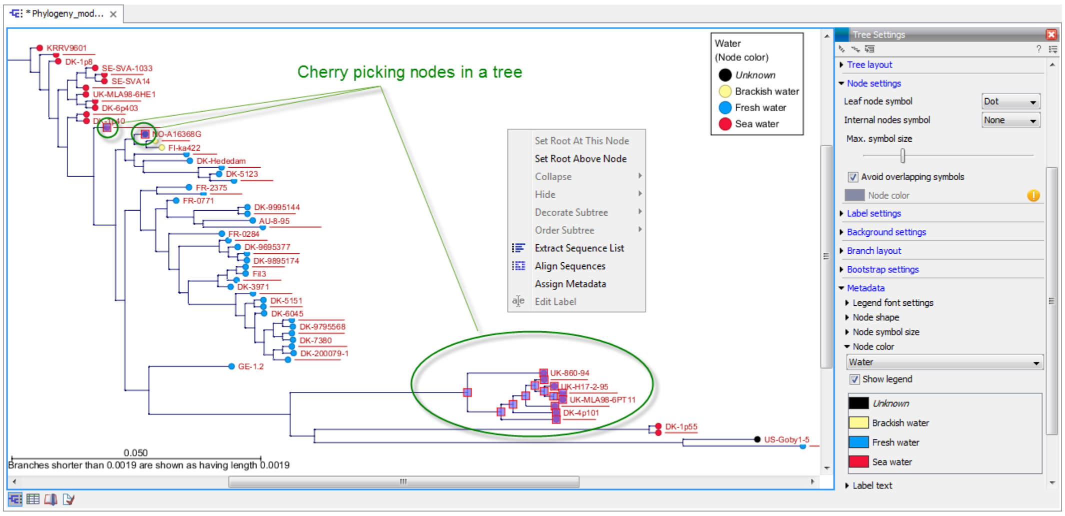
Name	Node type	Branch length	Bootstrap value	Size	Host	Water	Country	ACCNo	Year
	Root	0.00		0					
KRRV9601	Leaf	3.28E-4		1524	Japanese flounder	Sea water	JP	AB672614.1	1996
	Internal node	3.28E-4		0					
	Internal node	3.14E-6	40	0					
	Internal node	1.43E-5	40	0					
	Internal node	6.44E-4	61	0					
	Internal node	6.57E-4	60	0					
	Internal node	4.73E-6	36	0					
	Internal node	6.55E-4	71	0					
SE-SVA14	Leaf	6.45E-4		1524	Rainbow trout	Sea water	SE	AY546622.1	1998
DK-4p37	Leaf	1.11E-5		1524	Blue whiting	Sea water	DK	FJ460590.1	1997
SE-SVA-1033	Leaf	2.21E-6		1524	Rainbow trout	Sea water	SE	FJ460591.1	2000
DK-5e59	Leaf	1.31E-3		1524	Dab	Sea water	DK	AY546583.1	1998
	Internal node	1.32E-3	98	0					
	Internal node	6.59E-4	90	0					
SE-SVA31	Leaf	6.51E-4		1524	Atlantic Herring	Sea water	SE	AY546626.1	2000
DK-6p403	Leaf	6.62E-4		1524	Atlantic Herring	Sea water	DK	AY546584.1	1999

Column width: Manual

Show column
- ☑ Name
- ☐ Description
- ☑ Node type
- ☑ Branch length
- ☑ Bootstrap value
- ☑ Size
- ☐ Accession
- ☐ Start of sequence
- ☐ Latin name
- ☐ Taxonomy
- ☐ Common name
- ☐ Linear
- ☐ Sequence

Figure 6.5: *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.*

and click **Finish**. The multiple alignment will now be generated.



Branches shorter than 0.0019 are shown as having length 0.0019

Leaf node symbol: Dot

Internal nodes symbol: None

Max. symbol size

☑ Avoid overlapping symbols

Node color

Node color: Water

☑ Show legend

- ■ *Unknown*
- 🟨 Brackish water
- 🟦 Fresh water
- 🟥 Sea water

Figure 6.6: *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.*

Bibliography

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[Höhl et al., 2007] Höhl, M., Rigoutsos, I., and Ragan, M. A. (2007). Pattern-based phylogenetic distance estimation and tree reconstruction. *Evolutionary Bioinformatics*, 2:0–0.

[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.

[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.

[Michener and Sokal, 1957] Michener, C. and Sokal, R. (1957). A quantitative approach to a problem in classification. *Evolution*, 11:130–162.

[Posada and Crandall, 1998] Posada and Crandall (1998). Modeltest: testing the model of dna substitution. *Bioinformatics*.

[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.

[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.

[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.