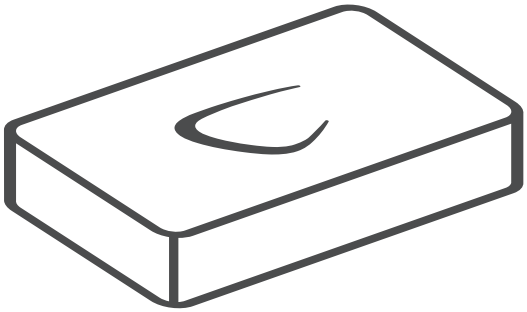

Bioinformatics Cell

User manual

User manual for

CLC Bioinformatics Cell 2.1

Windows, Mac OS X and Linux

June 16, 2010

This software is for research purposes only.

CLC bio
Finlandsgade 10-12
DK-8200 Aarhus N
Denmark

Contents

Chapter 1

Introduction to the CLC Bioinformatics Cell

The CLC Bioinformatics Cell includes three algorithms which have been accelerated using SIMD technology: Smith-Waterman BLAST, ClustalW and HMMER. The algorithms can be accessed in two ways:

- through a command line interface, or
- as an integrated plug-in to one of the CLC Workbenches (see www.clcbio.com/software):
  - CLC DNA Workbench
  - CLC RNA Workbench
  - CLC Protein Workbench
  - CLC Main Workbench
  - CLC Genomics Workbench

If the results of the CLC Bioinformatics Cell algorithms are used for further analysis and you wish to visually inspect the results, we recommend using the CLC Bioinformatics Cell from within the CLC Workbench. In this way you can take advantage of the graphical displays results, which can be used readily for further analyses within the CLC Workbench. The wide array of import and export formats is also an advantage of using the plug-in version.

On the other hand, if you perform e.g. Smith-Waterman BLAST searches with many query sequences, we recommend using the command line version. When using the workbench, you will get a new object for each query sequence, and this might not be the optimal solution with a large number of query sequences.

The same license can be used for both versions of the CLC Bioinformatics Cell, so you can also choose to use both.

In this introduction we will first describe the SIMD technology and how it is used in the CLC Bioinformatics Cell. Then follow three sections describing the algorithms on a general level. In the next chapter the system requirements and installation procedure are described.

Following the introduction are five chapters describing in detail how to use the algorithms, both through the command line interface and as a plug-in to the CLC Workbenches.

1.1 SIMD technology

SIMD (Single Instruction, Multiple Data) is a way to utilize the full power of modern CPU by parallelization. SIMD processors are capable of handling large vectors thus the need for serial processing is limited. This instruction set was introduced with the Intel Pentium MMX processor and was mainly used to accelerate computer games and multimedia. Nowadays most, if not all, personal computers can utilize the SIMD instruction set, thus it is possible to parallelize several bioinformatics algorithms which has not been seen before.

For additional information on SIMD technology see http://en.wikipedia.org/wiki/SIMD. You can also download the Cell White Paper at www.clcbio.com/white-paper which includes an elaborate benchmark of the Cell against a normal software implementation of Smith-Waterman

1.2 MPI support

To add more power to the CLC Bioinformatics Cell, it can be installed on a cluster. It supports MPI (Message Passing Interface). Benchmarks of the scalability of the CLC Bioinformatics Cell can be found at http://www.clcbio.com/white-paper.

A separate chapter of this user manual is devoted to the installation and operation of the MPI-version of CLC Bioinformatics Cell on a cluster (see chapter 3).

1.3 Smith-Waterman BLAST searches

Finding homologue DNA, RNA, or protein sequences in a database can be done in many ways. Smith-Waterman based searches is the only method that identifies all true hits, but the algorithm is very slow when working on large data-sets and most scientists therefore use the much faster BLAST search algorithm.

But the speed of BLAST comes on the expense of the quality. The sensitivity of BLAST is in fact so low, that there is significant risk of missing important sequences of interest. Compared to Smith-Waterman based searches, up to 50% of the search hits are thus not found using BLAST.

With the Cell, you can speed up a Smith-Waterman search previously taking 2 hours to around 1 minute, and the Cell thus removes the argument for using BLAST - at least in situations where you search through data where you not only need some of the answers but all of the answers.

The Cell includes the fastest Smith-Waterman search implementation ever made on standard hardware - nucleotide searches are accelerated up to 110 times and protein searches are accelerated up to 50 times on most computers (read more in the Cell white paper at www.clcbio.com/white-paper).

We call it Smith-Waterman BLAST because it is a combination of Smith-Waterman and BLAST. The Cell uses a BLAST-formatted database as the basis for the Smith-Waterman search, and it uses a lot of the same search parameters. Furthermore, the result of the Smith-Waterman search has the same format as a BLAST result. This makes it very easy to integrate the CLC Bioinformatics Cell into an existing work flow.

Smith and Waterman used a dynamic programming approach to create their algorithm for local sequence alignments. It finds the optimal local alignment, thus it is not a heuristic method as BLAST which does not guarantee an optimal alignment between the query sequence and a found hit. [Smith and Waterman, 1981]

1.4 ClustalW alignments

ClustalW is one of the most widely used methods for doing multiple sequence alignments. When aligning many sequences and when aligning long sequences, the speed of the algorithm is however not impressive.

As ClustalW is so widely used as it is, we have implemented a SIMD-version of it in the Cell, resulting in an acceleration of 4 to 5 times on most computers (read more in the Cell white paper at www.clcbio.com/white-paper).

1.5 HMMER

The Cell includes accelerated versions of hmmpfam and hmmsearch from the HMMER software package. Hmmpfam is used for searching with one or more sequences against an HMM database (like PFAM), and hmmsearch is used to search with a profile HMM against a sequence database.

To see the speed-up of the two HMMER programs, please consult the Cell white paper at www.clcbio.com/white-paper.

1.6 Contact information

The CLC Bioinformatics Cell is developed by:

CLC bio A/S
Science Park Aarhus
Finlandsgade 10-12
8200 Aarhus N
Denmark

http://www.clcbio.com

VAT no.: DK 28 30 50 87

Telephone: +45 70 22 55 09
Fax: +45 70 22 55 19

E-mail: info@clcbio.com

If you have questions or comments regarding the program, you are welcome to contact our support function:

E-mail: support@clcbio.com

Chapter 2

System requirements and installation

2.1 System requirements

2.1.1 Operating system platforms

The system requirements of CLC Bioinformatics Cell are these:

- Windows XP, Windows Vista or Windows 7
- Mac OS X 10.3 or newer
- Linux: Redhat or SuSE
- CPU architectures as described below
- *CLC Sequence Viewer*, *CLC Protein Workbench*, *CLC DNA Workbench*, *CLC RNA Workbench* or *CLC Main Workbench*

2.1.2 Supported Intel CPU architectures

The Cell uses the SSE2 extension of the Intel CPU instruction set. It was introduced in 2001.

Intel uses a number of different CPU microarchitectures with different performance characteristics. The recent ones are:

- The NetBurst microarchitecture:
  - Pentium 4 (670, 661, 660, 651, 650, 641, 640, 631, 630, 551, 541, 531, 524, 521)
  - Pentium D
  - Xeon (7150N, 7140M, 7140N, 7130M 7130N, 7120M, 7120N, 7110M, 7110N, 7041, 7040, 7030, 7020, 5080, 5063, 5060, 5050, 5030)
- The Pentium M microarchitecture:
  - Pentium M (780, 770, 765, 760, 755, 750, 745, 740, 735, 730, 725, 715, 705, 778, 758, 738, 718, 773, 753, 733J, 733, 723, 713)

  - Pentium Core Solo (T1400, T1300, U1500, U1400, U1300)
  - Pentium Core Duo (T2700, T2600, T2500, T2400, T2300, T2300E, L2500, L2400, L2300, U2500, U2400)

- The Core microarchitecture:
  - Pentium Core 2 Duo (E6700, E6600, E6400, E6300, E4300, T7600, T7400, T7200, T5600, T5500, L7400, L7200)
  - Pentium Core 2 Extreme (X6800, QX6700)
  - Xeon (3070, 3060, 3050, 3040, X3220, X3210, X5355, L5320, L5310, E5345, E5335, E5320, E5310, 5160, 5150, 5148 LV, 5140, 5130, 5120, 5110)

As shown, the Pentium Core processors have the Pentium M microarchitecture, while Pentium Core 2 processers have the Pentium Core microarchitecture.

The highest performance per GHz is with the Core microarchitecture while Pentium M has a lower performance and NetBurst is slightly lower.

2.1.3 Supported AMD CPU architectures

AMD introduced the SSE2 extension in 2003, so recent AMD architectures are supported and their performance is generally a little better than Intel Pentium M but not as high as the Intel Core microarchitecture.

2.1.4 How do I determine my CPU type?

If you do not know the type of your CPU, use this guide to find out:

CPU info: Windows XP

- Click **Start**
- Right-click **My computer**
- Click **Properties**

You will now see a dialog similar to the one shown in figure 2.1:

The red circle indicates the CPU information. Check with the list of CPU types above to see if your CPU is supported. If the CPU is not in the list, please send an email to support@clcbio.com with the information from this dialog.

CPU info: Mac OS X

- Click the **Apple** at the upper left corner of the screen
- Right-click **About This Mac**

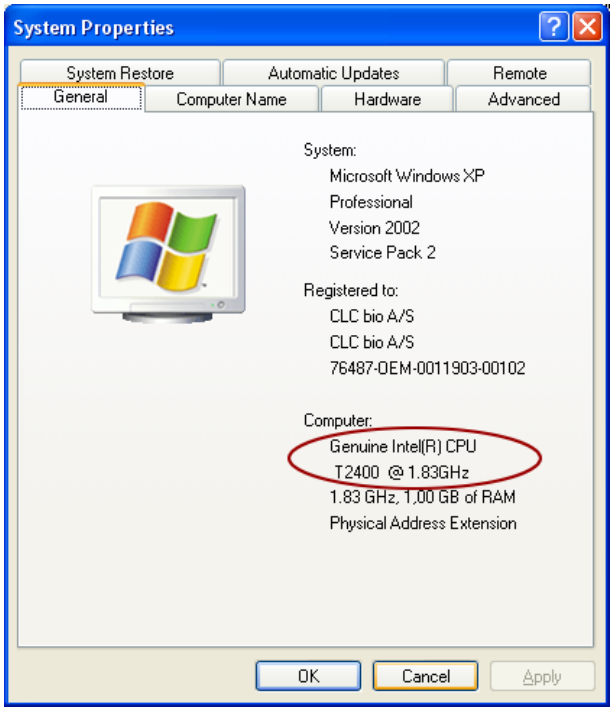

Figure 2.1: *Information about CPU on Windows XP.*

Figure 2.2: *Information about CPU on Mac OS X.*

You will now see a dialog similar to the one shown in figure 2.1:

The red circle indicates the CPU information. Check with the list of CPU types above to see if your CPU is supported. If the CPU is not in the list, please send an email to support@clcbio.com with the information from this dialog.

CPU info: Linux

Enter this:

```
cat /proc/cpuinfo
```

You will now see information about your CPU similar to figure 2.3:

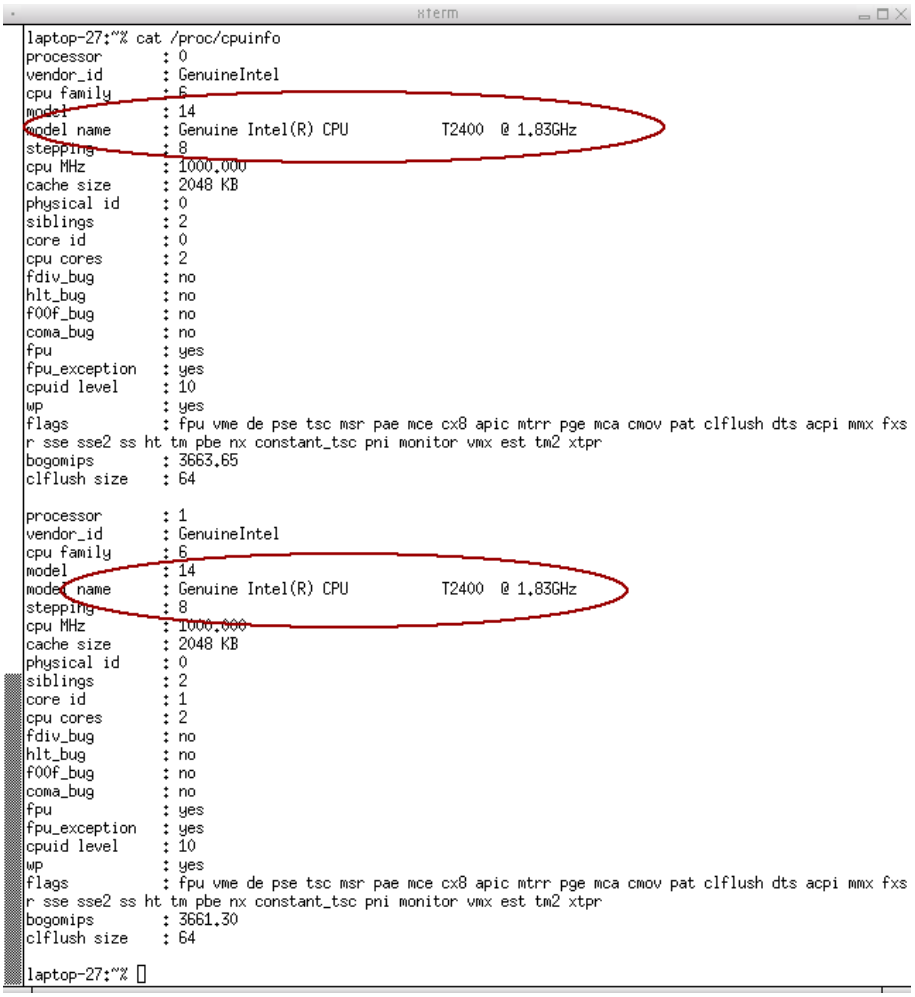

Figure 2.3: *Information about CPU on Linux.*

Check with the list of CPU types above to see if your CPU is supported. If the CPU is not in the list, please send an email to support@clcbio.com with the information from this dialog.

2.2 Installation of plug-in version

To carry out CLC Bioinformatics Cell installation, please go to one of the web pages specified below and follow the step-by-step instructions. The installation procedure is slightly different depending on your computer operating system. Please select the operating system on where the plug-in should run.

- **Mac.** http://www.clcbio.com/index.php?id=993
- **Linux.** http://www.clcbio.com/index.php?id=992
- **Windows.** http://www.clcbio.com/index.php?id=994

2.3 Installation of command line version

The installation of the command line version of CLC Bioinformatics Cell is slightly different depending on your platform:

2.3.1 Windows installation

1. Download the distribution from http://www.clcbio.com/download_bfx_cell.
2. Unzip the files in the zip-file to a folder on your computer.
3. Double-click the file *host_info.bat*.
4. This program will generate an email to license@clcbio.com with information about your computer. This is used to create a license key.
5. Send the email.
6. When we have received your email, we will generate a license key file which is sent back to you by email.
7. Save the license key file (*.lic) in either the
   - *working directory*,
   - `$ALLUSERSPROFILE\CLC bio\Licenses` or
   - `$APPDATA\CLC bio\Licenses`

   or you can save it in another folder and specify this location in the environment variable called *CLCBIO_LICENSE_PATH*.
8. You are ready to use the CLC Bioinformatics Cell.

2.3.2 Mac installation

1. Download the distribution from http://www.clcbio.com/download_bfx_cell.
2. Unzip the files in the zip-file to a folder on your computer.
3. Double-click the file *host_info*. This will generate one or more 16-digit numbers.
4. Copy the first number into an email and send it to license@clcbio.com. This number is used to create a license key.
5. When we have received your email, we will generate a license key file which is sent back to you by email.
6. Save the license key file (*.lic) in either the
   - *working directory*,
   - /Library/Application Support/CLC bio/Licenses or
   - $HOME/Library/Application Support/CLC bio/Licenses

   or you can save it in another folder and specify this location in the environment variable called *CLCBIO_LICENSE_PATH*.
7. You are ready to use the CLC Bioinformatics Cell.

2.3.3 Linux installation

1. Download the distribution from http://www.clcbio.com/download_bfx_cell.
2. Unzip the files in the zip-file to a folder on your computer.
3. Run the file *host_info*. This will generate one or more 16-digit numbers.
4. Copy the first number into an email and send it to license@clcbio.com. This number is used to create a license key.
5. When we have received your email, we will generate a license key file which is sent back to you by email.
6. Save the license key file (*.lic) in either the
   - *working directory*,
   - /etc/clcbio/licenses or
   - $HOME/.clcbio/licenses

   or you can save it in another folder and specify this location in the environment variable called *CLCBIO_LICENSE_PATH*.
   - If you are using tcsh or a similar shell, the command for setting the environment variable would be *setenv CLCBIO_LICENSE_PATH /path/to/license*
   - If you are using bash or a similar shell, the command for setting the environment variable would be *export CLCBIO_LICENSE_PATH=/path/to/license*
7. You are ready to use the CLC Bioinformatics Cell.

2.3.4 Using a license server

If you are using a license server rather than stand-alone licenses for the CLC Bioinformatics Cell, the licensing steps are a little different: The *host_info* program included in the distribution should be run on the computer where the license server is to be installed (if the license server is running a different operating system, you need to download the full distribution, even though you only need the host_info program). The license that you will receive from CLC bio is this valid for that computer.

In order to make the CLC Bioinformatics Cell contact the license server for a license, you need to create a text file in the *working directory* called `license.properties` including the following information:

```
serverip=192.168.1.200
serverport=6200
useserver=true
```

The `serverip` and `serverport` should be edited to match your license server set-up.

You can read more about the license server at the bottom of http://www.clcbio.com/usermanuals.

Chapter 3

Installing and running on a computer cluster

3.1 Platform, network and CPU requirement

From version 2.1 the Cell can run in a Linux cluster environment using *Open MPI*. Open MPI is an open source implementation of MPI-2 (Message Passing Interface version 2) and is developed and maintained by the Open MPI project.

For supported networks see Open MPI project home page at `http://www.open-mpi.org`.

For supported CPUs see sections 2.1.2 and 2.1.3.

3.2 How does the Cell work on a cluster?

A cluster is basically a number of computers, usually called nodes, interconnected using a network. Connected to the cluster are one or more *login servers*, which are the computers from where MPI programs can be started. The set of nodes and the set of login servers are not necessarily disjoint, e.g. a node can also serve as a login server.

The Cell uses a master-slave setup where exactly one *master node* is responsible of scheduling jobs and collecting results, and a number of *slave nodes* are doing the actual calculations.

3.3 Installing Open MPI

In this section you can see how to set up a cluster interconnected by a network. We assume that the nodes are running Linux and have been given valid IP addresses. Furthermore, we assume that a user account, say *clusteruser*, has been added to all nodes and login servers.

Three steps are required before a cluster can run the CLC Bioinformatics Cell:

1. Open MPI must be installed
2. A line must be added to /etc/bashrc
3. The firewall on each node must allow incoming traffic from other nodes

This will be explained in details below:

The Cell version 2.1 uses Open MPI version 1.2.5 stable which has to be downloaded from the Open MPI project home page and installed on all nodes and login servers. Open MPI version 1.2.5 stable can be downloaded from the Open MPI project home page at http://www.open-mpi.org.

To install Open MPI run the following commands as super user:

```
> tar -xjf openmpi-1.2.5.tar.bz2
> cd openmpi-1.2.5
> ./configure
> make all install
```

To make Open MPI programs run correctly, the following line must be added to the file /etc/bashrc on all nodes and login servers:

```
export LD_LIBRARY_PATH=/usr/local/lib
```

Finally, the fire wall on each nodes must allow incoming traffic from other nodes.

To test if Open MPI is installed correctly on say node01, node02 and node03, log in as clusteruser and run the following command:

```
> mpiexec -np 3 -host node01,node02,node03 hostname
```

where the number after *-np* is number of nodes and the comma separated list after *-host* contains the DNS names of the nodes. The command will execute the program "hostname" on the three nodes and output something similar to this:

```
node01.office
node02.office
node03.office
```

3.4 Install licences

The license file must be installed on each slave node. When running an MPI program from the login server using mpiexec, it will try to change the directory on all nodes to working directory on the login server, if possible. If you start the program "pwd" (print working directory) on node01, node02 and node03 from the login server using the following command:

```
> mpiexec -np 3 -host node01,node02,node03 pwd
```

and working directory is */home/clusteruser*. The output will be:

```
/home/clusteruser
/home/clusteruser
/home/clusteruser
```

provided that the directory */home/clusteruser* exists on node01, node02 and node03.

The license file must be located in the working directory on all slaves. That is, if working directory on the login server is */home/clusteruser*, the license file must be located in the directory */home/clusteruser* on all slaves.

As an alternative to using working directory, you can use the environment variable called `CLCBIO_LICENSE_PATH` to specify the location of the license file (see section 2.3).

3.4.1 Obtaining license files

A license file has to be issued specifically for each node in the cluster. This is done in the same way as for single computers by running the *host_info* program on each node and collecting the information for each node and sending it to support@clcbio.com. See section 2.3.3 for information about doing this on a single computer.

Note that the master computer does not need a license - only the slave nodes. This means that if you buy e.g. a license for 8 dual core computers, you will be able to run it on a cluster with one master and 8 dual core nodes.

3.5 Installing the MPI-version of the CLC Bioinformatics Cell

Unzip the CLC Bioinformatics Cell distribution file and install the contents in the same directory on all nodes and all login servers:

- `blastall_cell_mpi`.
- `clustal_cell_mpi`.
- `hmmpfam_cell_mpi`.
- `hmmsearch_cell_mpi`.

3.6 Location of databases

For `blastcell_cell_mpi`, the database files (but not the query sequences) must be located in the same directory, e.g. /databases/blastall/, on all nodes.

For `hmmpfam_cell_mpi`, the hmm files (but not the fasta files) must be located in the same directory, e.g. /databases/hmmpfam/, on all nodes.

For `hmmsearch_cell_mpi`, the fasta files (but not the hmm file) must be located in the same directory, e.g. /databases/hmmsearch/ on all nodes.

To maximize performance it is recommended that the databases, hmm files and fasta files respectively are located on a local disk and not a shared network directory.

3.7 Running the Cell

The Cell MPI programs `blastall_cell_mpi`, `clustal_cell_mpi`, `hmmpfam_cell_mpi` and `hmmsearch_cell_mpi` have the same options as there corresponding non-MPI programs

`blastall_cell`, `clustal_cell`, `hmmpfam_cell` and `hmmsearch_cell` with one exception: if the *-a* option is not set for `blastall_cell`, the number of processors is set to 1. If the -a option is not set for `blastall_cell_mpi`, the number of processors will be set automatically to the number of cores. The reason is that nodes may not have the same number of cores in a cluster.

To start an MPI program on a cluster using Open MPI, use the command *mpiexec*, which among others have the following two options:

-np Number of processes to run

-host List of hosts to invoke processes on

When running the Cell, the first host in the list of hosts will be the master node and the subsequent hosts will be the slaves. To run a Cell MPI program, e.g. `blastall_cell_mpi`, on node01, node02 and node03 execute the command:

```
> mpiexec -np 3 -host node01,node02,node03 blastall_cell_mpi -c cell_sw
-p blastn -i horse_est_100.fasta -d /databases/blastall/horse_est.fasta
```

Chapter 4

Using the Smith-Waterman BLAST plug-in from a CLC Workbench

Smith and Waterman used a dynamic programming approach to create their algorithm for local sequence alignments. It finds the optimal local alignment, thus it is not a heuristic method as BLAST which does not guarantee an optimal alignment between the query sequence and a found hit. [Smith and Waterman, 1981]

If the CLC Bioinformatics Cell is installed into one of the CLC Workbenches as a plug-in, you can access the Smith-Waterman BLAST search via the **Toolbox** like the other analyses. In this way, you can take advantage of the integrated platform of the workbench. If you perform searches with many query sequences, we recommend using the command line based Smith-Waterman BLAST (see section 5). When using the workbench, you will get a new object for each query sequence, and this might not be the optimal solution with a large number of query sequences.

4.1 Defining a local database

The Smith-Waterman BLAST search is performed against a local database. This means that you have to define a database before you can start using the CLC Bioinformatics Cell plug-in. There are two ways of doing this:

- Create the local database using the CLC Workbench. The database is created by clicking **Create Local BLAST Database (**  **)** under **BLAST** () in the **Toolbox**. The database can be based on sequences located in the **Navigation Area** or it can be based on FASTA files on your computer. In the dialog you can click the **Help** button (?) to learn more about how to create BLAST databases.
- Import an existing BLAST database in the CLC Workbench. Click **Import** () and select the .nhr (for nucleotides) or .phr (for proteins) file. Importing a BLAST database in this way creates a reference to the database in the **Navigation Area**.

4.2 Performing Smith-Waterman BLAST search

When the database has been created or imported, you can conduct a Local BLAST search:

Toolbox | **Cell Tools** () | **Smith-Waterman BLAST Search** ()

This opens the dialog shown in figure 4.1:

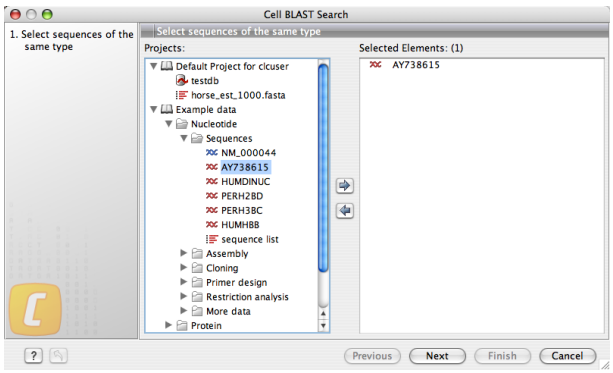

Figure 4.1: *Choose one or more sequences as query sequences.*

If you had already selected one or more sequences, they are now listed in the **Selected Elements**. Use the arrows to add or remove sequences or sequence lists.

Click **Next**.

This opens the dialog seen in figure 4.2:

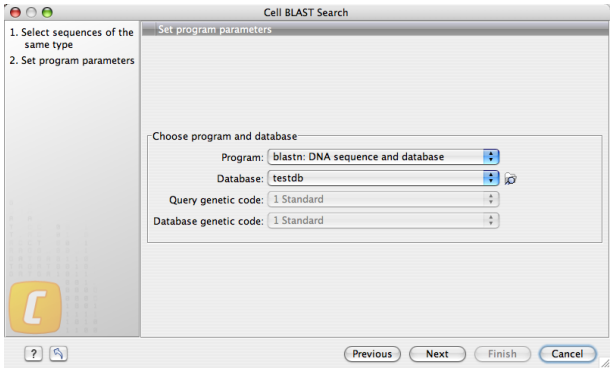

Figure 4.2: *Choose a BLAST program and a local database to conduct BLAST search.*

In **Step 2**, you can choose between different BLAST methods:

- **BLAST search for DNA sequences:**
  - **BLASTn: DNA sequence against DNA database.** This BLAST method is used to identify homologous DNA sequences to your query sequence.
  - **BLASTx: Translated DNA sequence against Protein database.** If you want to search in protein databases, this BLAST method allows for automated translation of the DNA input sequence and searching in various protein databases.
  - **tBLASTx: Translated DNA sequence against Translated DNA database.** Here both the input DNA sequence and the searched DNA database is automatically translated.
- **BLAST search for protein sequences:**
  - **BLASTp: Protein sequence against Protein database.** This the most common BLAST method used when searching for homologous protein sequences having a protein sequence as search input.

- **tBLASTn: Protein sequence against Translated DNA database.** Here the protein sequence is searched against an automatically translated DNA database.

When choosing BLASTx or tBLASTx to conduct a search, you get the option of selecting a translation table for the genetic code. The standard genetic code is set as default. This is particularly useful when working with organisms or organelles which have a genetic code that differs from the standard genetic code.

In this step you also choose which of your local BLAST databases you want to conduct the search in. Clicking **Select Database** ( ) opens the dialog shown in figure 4.3:

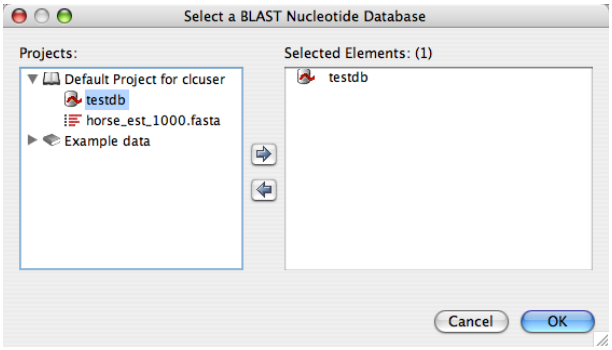

Figure 4.3: *Select your local BLAST database.*

Select a database Click **Next**.

This will show the dialog in figure 4.4:

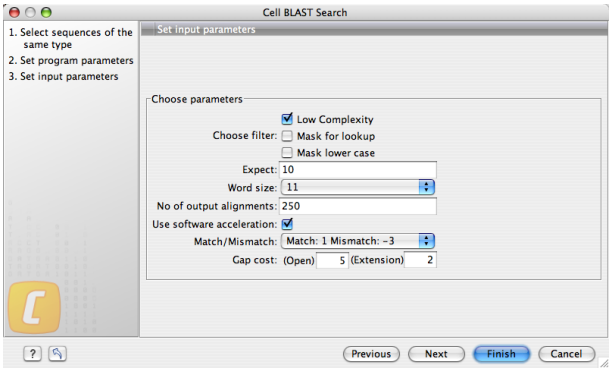

Figure 4.4: *Examples of different limitations which can be set before submitting a BLAST search.*

The following description of BLAST search parameters is based on information from http://www.ncbi.nlm.nih.gov/BLAST/blastcgihelp.shtml.

- **Choose filter**
  - **Low-complexity.** Mask off segments of the query sequence that have low compositional complexity. Filtering can eliminate statistically significant, but biologically uninteresting reports from the BLAST output (e.g. hits against common acidic-, basic- or proline-rich regions), leaving the more biologically interesting regions of the query sequence available for specific matching against database sequences.
  - **Human Repeats.** This option masks Human repeats (LINE's and SINE's) and is especially useful for human sequences that may contain these repeats. Filtering for

repeats can increase the speed of a search especially with very long sequences (>100 kb) and against databases which contain large number of repeats (htgs).

  - **Mask for lookup.** This option masks only for purposes of constructing the lookup table used by BLAST. BLAST searches consist of two phases, finding hits based upon a lookup table and then extending them.
  - **Mask lower case.** With this option selected you can cut and paste a FASTA sequence in upper case characters and denote areas you would like filtered with lower case. This allows you to customize what is filtered from the sequence during the comparison to the BLAST databases

- **Expect.** The statistical significance threshold for reporting matches against database sequences: the default value is 10, meaning that 10 matches are expected to be found merely by chance, according to the stochastic model of Karlin and Altschul (1990). If the statistical significance ascribed to a match is greater than the EXPECT threshold, the match will not be reported. Lower EXPECT thresholds are more stringent, leading to fewer chance matches being reported. Increasing the threshold shows less stringent matches. Fractional values are acceptable.

- **Word size.** BLAST is a heuristic that works by finding word-matches between the query and database sequences. You may think of this process as finding "hot-spots" that BLAST can then use to initiate extensions that might lead to full-blown alignments. For nucleotide-nucleotide searches (i.e. "BLASTn") an exact match of the entire word is required before an extension is initiated, so that you normally regulate the sensitivity and speed of the search by increasing or decreasing the wordsize. For other BLAST searches non-exact word matches are taken into account based upon the similarity between words. The amount of similarity can be varied so that you normally uses just the wordsizes 2 and 3 for these searches.

- **No of output alignments.** Limit the number of sequences to which the query sequence can align. There is a possibility to get more alignments than unique hit sequences as the query sequence may align to different location in the hit sequence.

- **Use software acceleration.** Use accelerated SIMD instructions for the Smith-Waterman BLAST search.

- **Match/Mismatch.** Set reward and penalty costs for nucleotide searches. Only applicable for blastn searches.

- **Gap cost.**

  - **Open gap cost.** Cost to open a gap.
  - **Extension gap cost.** Cost to extend an already inserted gap.

Click **Next** if you wish to adjust how to handle the results. Here you can choose to either **Open** or **Save** the results of the search. Click **Finish** to start the search. The result of the search is a BLAST search element as described below:

4.3 Viewing the search result

The result of the search is shown as a graphical representation of the hits (see figure 4.5). This view can be zoomed using the zoom tools () in the **Toolbar**.

For more help on this view, press the **F1** button on your keyboard or read the **BLAST graphics** section of the Workbench user manual (can be downloaded from http://www.clcbio.com/usermanuals).

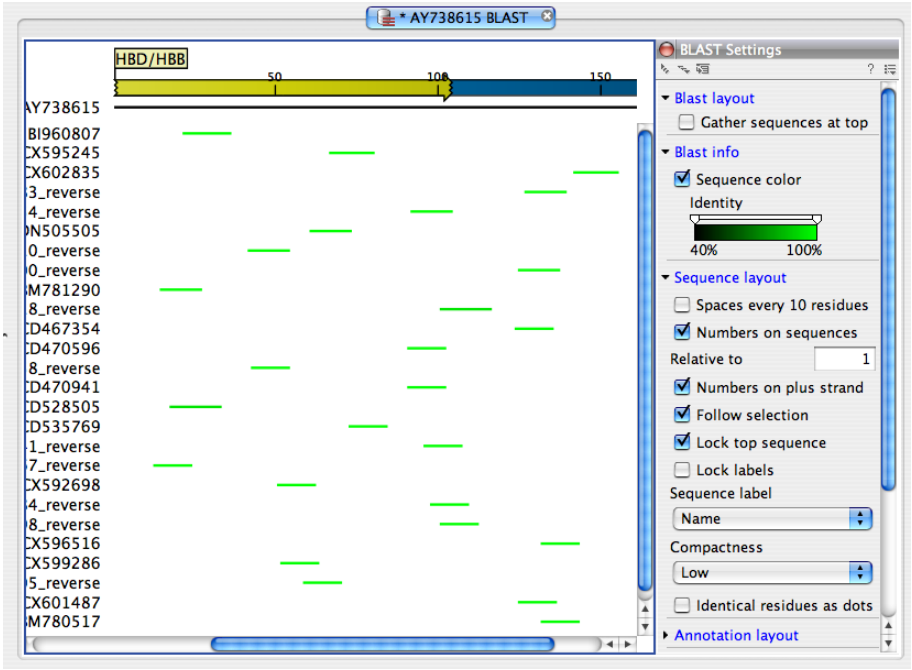

Figure 4.5: *The result of a Smith-Waterman BLAST search.*

The result can also be shown in a table:

Right-click the tab of the result view | Show ( ) | BLAST table ( )

The table shows additional information about the hits as shown in figure 4.6.

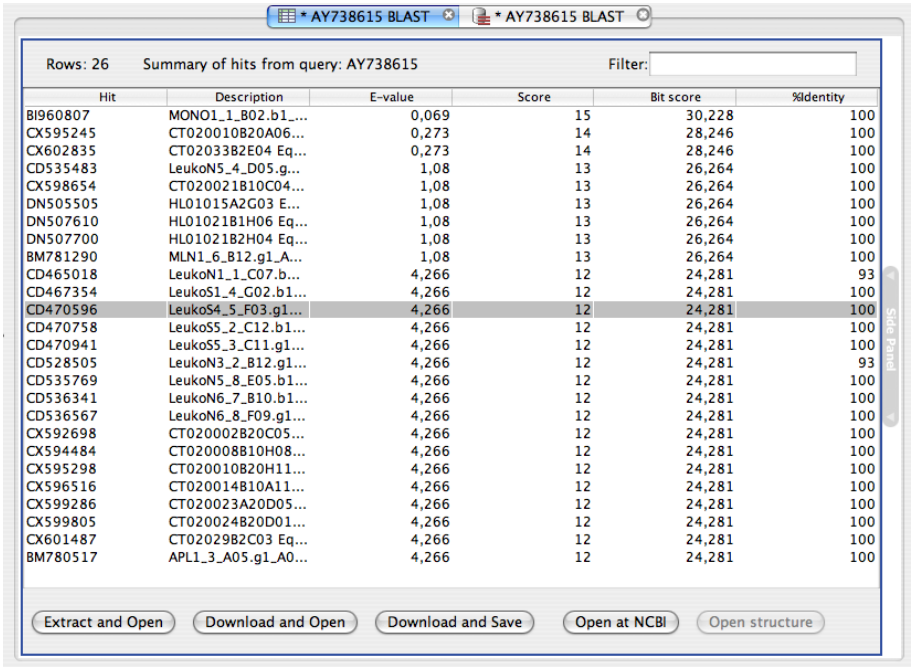

Figure 4.6: *The result of a Smith-Waterman BLAST search shown as a table.*

For more help on this view, press the **F1** button on your keyboard or read the **BLAST table** section of the Workbench user manual (can be downloaded from http://www.clcbio.com/usermanuals)

Chapter 5

Command line usage of Smith-Waterman BLAST

For easy integration into an existing bioinformatics work flow we have used NCBI BLAST 2.2.15 as the basis of the command line interface. Furthermore, the output from the CLC Bioinformatics Cell is fully compatible with the NCBI BLAST output so scripts or programs used to parse BLAST results can also be used to parse the output from CLC Bioinformatics Cell.

5.1 Programs for Smith-Waterman BLAST search

For Smith-Waterman database search, the same options are available as for BLAST in terms of how to treat the query and database sequences:

The program option is the -p option for the command line interface.

Option	Query Type	DB Type	Comparison	Note
blastn	Nucleotide	Nucleotide	Nucleotide-Nucleotide	
blastp	Protein	Protein	Protein-Protein	
tblastn	Protein	Nucleotide	Protein-Protein	The database is translated to protein
blastx	Nucleotide	Protein	Protein-Protein	The queries are translated to protein
tblastx	Nucleotide	Nucleotide	Protein-Protein	The queries and database are translated to protein

5.2 Smith-Waterman BLAST options

Some BLAST options relate to how BLAST makes its approximations when searching the database. These approximations are not used with the Smith-Waterman algorithm, making the options irrelevant for the Smith-Waterman BLAST. PSI BLAST and frame shifts are not supported by the current Smith-Waterman implementation. Here is an overview of available command line options:

Option	Status	Description
-p	Unchanged	Program Name
-d	Unchanged	Database
-i	Unchanged	Query File
-e	Unchanged	Expectation value (E)
-m	Unchanged	alignment view options
-o	Unchanged	BLAST report Output File
-F	Unchanged	Filter query sequence
-G	Unchanged	Cost to open a gap
-E	Unchanged	Cost to extend a gap
-X	Not relevant	X dropoff value for gapped alignment
-I	Unchanged	Show GI's in deflines
-q	Unchanged	Penalty for a nucleotide mismatch
-r	Unchanged	Reward for a nucleotide match
-v	Unchanged	Number of database sequences
	Unchanged	to show one-line descriptions for
-b	Unchanged	Number of database sequence to show alignments for
-f	Not relevant	Threshold for extending hits
-g	Not relevant	Perform gapped alignment
-Q	Unchanged	Query Genetic code to use
-D	Unchanged	DB Genetic code
-a	Unchanged	Number of processors to use
-O	Unchanged	SeqAlign file
-J	Unchanged	Believe the query defline
-M	Unchanged	Matrix
-W	Not relevant	Word size
-z	Unchanged	Effective length of the database
-K	Not relevant	Number of best hits from a region to keep
-P	Not relevant	0 for multiple hit, 1 for single hit
-Y	Unchanged	Effective length of the search space
-S	Unchanged	Query strands to search against database
-T	Unchanged	Produce HTML output
-l	Unchanged	Restrict search of database to list of GI's
-U	Unchanged	Use lower case filtering of FASTA sequence
-y	Not relevant	X dropoff value for ungapped extensions in bits
-Z	Not relevant	X dropoff value for final gapped alignment in bits
-R	Not supported	PSI-TBLASTN checkpoint file
-n	Not relevant	MegaBlast search
-L	Unchanged	Location on query sequence
-A	Not relevant	Multiple Hits window size, default if zero
-w	Not supported	Frame shift penalty (OOF algorithm for blastx)
	Not relevant	Length of the largest intron allowed in
-t	Not relevant	a translated nucleotide sequence when
	Not relevant	linking multiple distinct alignments
-B	Not relevant	Number of concatenated queries, for blastn and tblastn
-V	Not relevant	Force use of the legacy BLAST engine
-C	Unchanged	Use composition-based statistics for tblastn
-s	Not relevant	Compute locally optimal Smith-Waterman alignments
-c	New	use soft_sw for software and
		cell_sw for accelerated SIMD instructions

The option –clcversion will write the version number.

5.3 Examples on command line usage

Before you perform a search, the database has to be preformatted using the *formatdb* program from NCBI (see section 5.4).

If you wish to search for a nucleotide query in a nucleotide database using the accelerated software Smith-Waterman BLAST algorithm in the CLC Bioinformatics Cell, you write:

```
blastall_cell –c cell_sw –p blastn –d database.fasta –i query.fasta
```

The output of the Smith-Waterman search is in exactly the same format as a normal BLAST search, the only difference is that the Smith-Waterman approach usually gives more hits, due to being a more accurate search method than BLAST.

If you have a dual core processor, and you wish the search to be performed on both cores, use this command instead:

```
blastall_cell –c cell_sw –p blastn –a 2 –d database.fasta –i
query.fasta
```

If you wish to search for a nucleotide query in a nucleotide database using the normal (slower) software implementation of Smith-Waterman BLAST, you write:

```
blastall_cell –c soft_sw –p blastn –d database.fasta –i query.fasta
```

5.4 Examples on how to create a local BLAST database

If you want to format a local database for use with BLAST do the following.

- **-t** Title for database file (String) Optional.
- **-i** Input file(s) for formatting (this parameter must be set) (File In)
- **-l** Logfile name: (File Out) Optional default = formatdb.log.
- **-p** -p Type of file
  - T - protein
  - F - nucleotide (T/F)
  - Optional default = T
- **-o** Parse options
  - T - True: Parse SeqId and create indexes.
  - F - False: Do not parse SeqId. Do not create indexes.
  - (T/F) Optional default = F.

- **-a** Input file is database in ASN.1 format (otherwise FASTA is expected)
  - T - True
  - F - False
  - (T/F) Optional default = F.
- **-b** ASN.1 database in binary mode
  - T - binary
  - F - text more
  - (T/F) Optional default = F.
- **-e** Input is a Seq-entry (T/F) Optional default = F
- **-n** Base name for BLAST files (String) Optional

```
formatdb -p T -i database.fasta
```

For detailed usage of the formatdb options please consult the formatdb manual at ftp://ftp.ncbi.nlm.nih.gov/blast/documents/formatdb.html

Chapter 6

Using the ClustalW plug-in from a CLC Workbench

The ClustalW alignment method was in the mid nineties improved over previous progressive alignment methods [Thompson et al., 1994]. Even though this alignment method is more than ten years old it is still a highly popular alignment method and has become the method of choice to many researchers. One of the reasons for the popularity is the availability for most computer platforms and the easy integration on websites.

There is still one problem for ClustalW as for all other stand alone alignment programs. None of them can visualize annotations on the aligned sequences. Most researches align their own sequence to some reference sequences which are often retrieved from one or more of the large public databases, e.g. GenBank. As input these programs require a Fasta file and they typically export in their own alignment format. The ClustalW plug-in for use in a CLC Workbench use the ClustalW program for the alignment calculation but still retains annotations on the sequences when the alignment is displayed (see figure 6.3).

One of the advantages of ClustalW is that it uses very little computer memory on rather large sequences so you can align large sequences without having a state-of-the-art computer.

6.1 Alignment parameters

If the CLC Bioinformatics Cell is installed into one of the CLC Workbenches as a plug-in, you can access the ClustalW alignment algorithm via the **Toolbox** like the other analyses in the workbench.

To create an alignment using the CLC Bioinformatics Cell ClustalW algorithm:

select sequences to align | **Toolbox in the Menu Bar** | **Cell Tools** | **ClustalW** ()

This opens the dialog shown in figure 6.1.

If you had already selected one or more sequences, they are now listed in the **Selected Elements**. Use the arrows to add or remove sequences or sequence lists.

Click **Next** to open the dialog shown in figure 6.2.

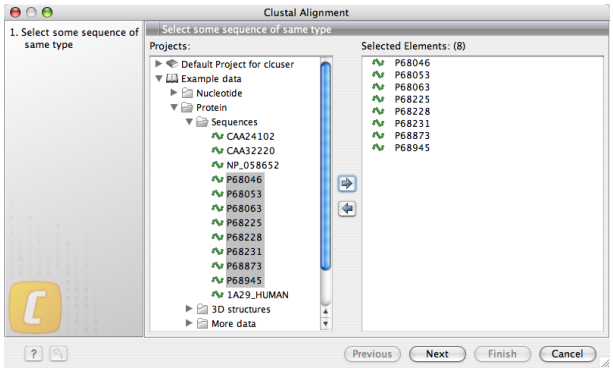

Figure 6.1: *Selecting sequences for the alignment.*

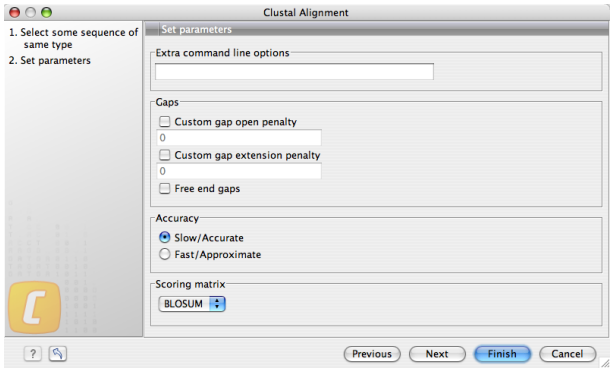

Figure 6.2: *Setting parameters for ClustalW.*

- **Extra command line options.** Here it is possible to specify extra program specific command line options and advanced settings. See section 7 for a full list of available options.
- **Custom gap open penalty.** Set the gap opening costs.
- **Custom gap extension penalty.** Set the gap extension costs.
- **Free end costs.** Enable free end costs.
- **Slow/accurate.** Use a very accurate alignment method.
- **Fast/appriximate.** Use a fast but less accurate alignment method.
- **Scoring matrix.** Select from a list of different scoring matrices.

Click **Next** lets you choose if you wish to see the phylogenetic tree generated by ClustalW.

When you click **Next** again, you can adjust how to handle the resulting alignment. Here you can choose to either **Open** or **Save** the alignment. Click **Finish** to start creating the alignment.

An example of the output of the ClustalW alignment algorithm is shown in figure 6.3.

Notice! Annotations are not displayed per default - use the **Side Panel** to the right to show annotations and customize the view.

For help on the alignment view, press the **F1** button on your keyboard or read the **View alignments** section of the Workbench user manual (can be downloaded from `http://www.clcbio.com/usermanuals`)

Figure 6.3: *Output of ClustalW alignment. Annotations from the original sequence is retained on the output.*

Chapter 7

Command line usage of ClustalW

For easy integration in an existing bioinformatics work flow you can use the command line interface to ClustalW. Furthermore, the output from the CLC Bioinformatics CellClustalW implementation is fully compatible with the output of ClustalW so scripts or programs used to parse original ClustalW results can also be used to parse the output from CLC Bioinformatics CellClustalW implementation.

7.1 ClustalW Options

Below is a thorough listing of available command-line options in ClustalW.

7.1.1 DATA (sequences)

- **-INFILE=file.ext** input sequences.
- **-PROFILE1=file.ext and -PROFILE2=file.ext** profiles (old alignment).

7.1.2 VERBS (do things)

- **-OPTIONS** List the command line parameters.
- **-HELP or -CHECK** outline the command line params.
- **-ALIGN** do full multiple alignment.
- **-TREE** calculate NJ tree.
- **-BOOTSTRAP(=n)** bootstrap a NJ tree (n= number of bootstraps; def. = 1000).
- **-CONVERT** output the input sequences in a different file format.

7.1.3 PARAMETERS (set things)

- **-INTERACTIVE** read command line, then enter normal interactive menus.
- **-QUICKTREE** use FAST algorithm for the alignment guide tree.

- **-TYPE=** PROTEIN or DNA sequences.
- **-NEGATIVE** protein alignment with negative values in matrix.
- **-OUTFILE=** sequence alignment file name.
- **-OUTPUT=** GCG, GDE, PHYLIP, PIR or NEXUS.
- **-OUTORDER=** INPUT or ALIGNE.
- **-CASE** LOWER or UPPER (for GDE output only).
- **-SEQNOS=** OFF or ON (for Clustal output only).
- **-SEQNO_RANGE=** OFF or ON (NEW: for all output formats).
- **-RANGE=m,n** sequence range to write starting m to m+n.

7.1.4 Fast Pairwise Alignments

- **-KTUPLE=n** word size.
- **-TOPDIAGS=n** uTOPDIAGS=n.
- **-WINDOW=n** window around best diags.
- **-PAIRGAP=n** gap penalty.
- **-SCORE** PERCENT or ABSOLUTE.

7.1.5 Multiple Alignments

- **-NEWTREE=** :file for new guide tree
- **-USETREE=** :file for old guide tree
- **-MATRIX=** :Protein weight matrix=BLOSUM, PAM, GONNET, ID or filename
- **-DNAMATRIX=** :DNA weight matrix=IUB, CLUSTALW or filename
- **-GAPOPEN=f** :gap opening penalty
- **-GAPEXT=f** :gap extension penalty
- **-ENDGAPS** :no end gap separation pen.
- **-GAPDIST=n** :gap separation pen. range
- **-NOPGAP** :residue-specific gaps off
- **-NOHGAP** :hydrophilic gaps off
- **-HGAPRESIDUES=** :list hydrophilic res.
- **-MAXDIV=n** :% ident. for delay
- **-TYPE=** :PROTEIN or DNA
- **-TRANSWEIGHT=f** :transitions weighting

7.1.6 Profile Alignments

- **-PROFILE** :Merge two alignments by profile alignment
- **-NEWTREE1=** :file for new guide tree for profile1
- **-NEWTREE2=** :file for new guide tree for profile2
- **-USETREE1=** :file for old guide tree for profile1
- **-USETREE2=** :file for old guide tree for profile2

7.1.7 Sequence to Profile Alignments

- **-SEQUENCES** :Sequentially add profile2 sequences to profile1 alignment
- **-NEWTREE=** :file for new guide tree
- **-USETREE=** :file for old guide tree

7.1.8 Structure Alignments

- **-NOSECSTR1** :do not use secondary structure-gap penalty mask for profile 1
- **-NOSECSTR2** :do not use secondary structure-gap penalty mask for profile 2
- **-SECSTROUT=STRUCTURE or MASK or BOTH or NONE** :output in alignment file
- **-HELIXGAP=n** :gap penalty for helix core residues
- **-STRANDGAP=n** :gap penalty for strand core residues
- **-LOOPGAP=n** :gap penalty for loop regions
- **-TERMINALGAP=n** :gap penalty for structure termini
- **-HELIXENDIN=n** :number of residues inside helix to be treated as terminal
- **-HELIXENDOUT=n** :number of residues outside helix to be treated as terminal
- **-STRANDENDIN=n** :number of residues inside strand to be treated as terminal
- **-STRANDENDOUT=n** :number of residues outside strand to be treated as terminal

7.1.9 Trees

- **-OUTPUTTREE=nj** OR phylip OR dist OR nexus
- **-SEED=n** :seed number for bootstraps.
- **-KIMURA** :use Kimura's correction.
- **-TOSSGAPS** :ignore positions with gaps.
- **-BOOTLABELS=node OR branch** :position of bootstrap values in tree display

The option –clcversion will write the version number.

7.2 Examples on ClustalW command line usage

Note! Using the commandline version under Windows may require slightly different syntax. The program is under Windows invoked using clustalw_cell.exe.

The clustal program offers a text based usage of the program where available options are written on the console while the program is running. This is simply invoked by

```
clustalw_cell
```

If you wish to align all sequences in a fasta using default settings, you write:

```
clustalw_cell sequences.fasta
```

This will run the alignment program and place the resulting alignment in a file called sequences.aln.

Specify gap openening and extension costs.

```
clustalw_cell sequences.fasta –gapopen=5 gapext=2
```

Chapter 8

Using the HMMER plug-in from a CLC Workbench

The plug-in version of the CLC Bioinformatics Cell makes it possible to search with one ore more sequences against an HMM Profile database (hmmpfam). The hmmsearch program cannot be accessed from within the Workbench.

The result can be shown in three different ways:

- As annotations on the sequence showing the domains that were found.
- As a table showing the domains including statistics on E-value and bit score.
- As the traditional text output similar to the command line version of HMMER.

To perform a search:

Toolbox | **Cell Tools** () | **Pfam Domain Search (hmmpfam)** ()

This will bring up the dialog shown in figure 8.1

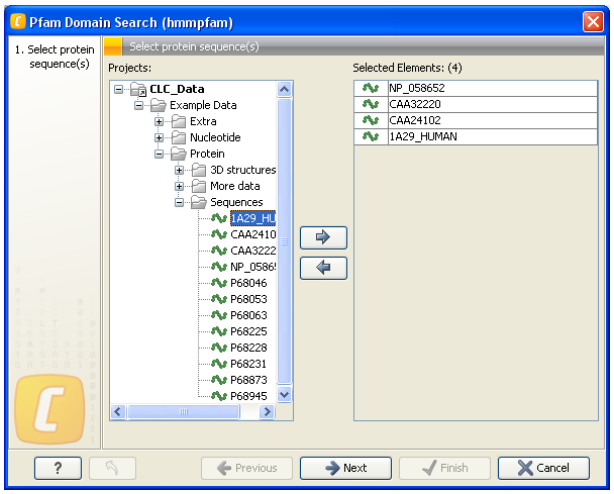

Figure 8.1: *Selecting protein sequences to search against the HMM profile database.*

If a sequence was already selected, this sequence is now listed in the **Selected Elements** window

of the dialog. Use the arrows to add or remove sequences or sequence lists from the selected elements. You can only use protein sequences for the search.

Click **Next** to adjust parameters for the search.

8.1 HMMER parameters

As shown in figure 8.2, there are three parameters to be set:

- **HMM profile.** The HMM profile database to search against. You can download databases from the web - the Pfam database is the most common database and is readily downloaded from http://www.sanger.ac.uk/Software/Pfam/. Alternatively, you can create a database using the hmmbuild tool of the HMMER package which is included in the command line version of the CLC Bioinformatics Cell.
- **Maximum E-value.** Sets an E-value cut-off for the domains to be shown in the result. If you don't specify a value, you might get results that are only distantly related to the sequence. Per default, the E-value cut-off is set to 10.
- **Maximum bit score.** Sets a bit score cut-off for the domains to be shown in the result. If no value is specified, the E-value will determine the cut-off instead.

You can read more about the E-value and bit score in the HMMER User's Guide at http://hmmer.janelia.org/.

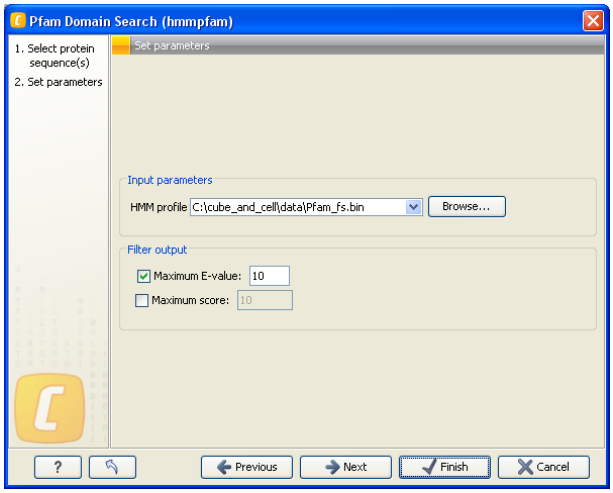

Figure 8.2: .

Click **Next** to see the output options for the Pfam Domain Search.

8.2 Output of HMMER

In figure 8.3, you can see the three ways to report the results of the search.

The three output options are:

- **Add annotations to sequence.** When a domain is found, an annotation is added to the sequence. The annotation includes the following information:

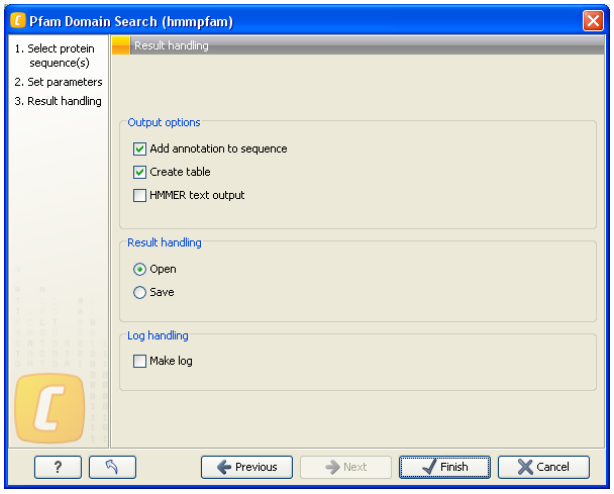

Figure 8.3: *Choosing three different output options.*

  - **Model.** The name of the model.
  - **HMM Start and HMM End.** The part of the model which is annotated as a domain on the sequence.
  - **Score.** The bit score for the hit.
  - **E-value.** The E-value for the hit.
  - A note stating "Found be CLC Cell HMMER" to show the origin of the annotation.

- **Create table.** The table summarizes the information about the domains. If more than one sequence is selected, the table will include information about all the sequences making it easy to compare the results for different sequences. The table displays the following information:

  - **Sequence.** The name of the sequence (only visible if more than one sequence is selected).
  - **Start.** The start position of the domain on the sequence.
  - **End.** The end position of the domain on the sequence.
  - **Model.** The name of the model.
  - **HMM Start and HMM End.** The part of the model which is annotated as a domain on the sequence.
  - **Score.** The bit score for the hit.
  - **E-value.** The E-value for the hit.
  - A note stating "Found by CLC Cell HMMER" to show the origin of the annotation.

  In short, the information in the table is identical with the information on the annotations which are added to the sequence.

- **HMMER text output.** This is the same output as produced by the command line version of HMMER.

It is also possible to create a log summarizing the number of domains found for each sequence. This is particularly convenient when doing batch searches with many sequences.

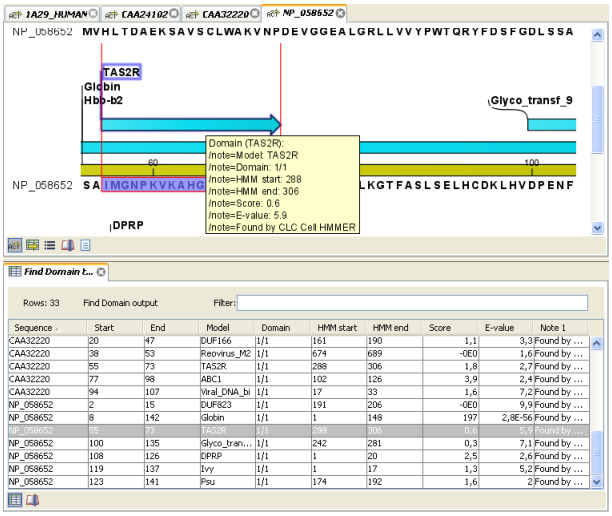

Figure 8.4: *At the top the annotations on the sequence are shown. At the bottom is the table summarizing the result for all four sequences.*

In figure 8.4, you can see the table output and the annotations on the sequence (the two first output options).

Clicking in the table will select the corresponding region on the sequence, so it can be used to easily browse the found domains. The content of the table can be copied and pasted into e.g. Excel for further analysis.

Chapter 9

Command line usage of HMMER

For easy integration in an existing bioinformatics work flow you can use the command line version of HMMER. Furthermore, input and output formats in the Cell version of HMMER are almost identical to the normal HMMER[1], so existing scripts or programs used to parse results can also be used to parse the output from CLC Bioinformatics Cell HMMER implementation.

In general, the implementation is very close to the original HMMER 2 Package, so we encourage you to read the well-written HMMER User's Guide if you are not familiar with the HMMER Package. You can find it at: http://hmmer.janelia.org/.

Included in the distribution of the CLC Bioinformatics Cell are also the original programs from the HMMER Package (version 2.3.2).

9.1 HMMER command line options

Below is a list of the options for both hmmpfam and hmmsearch. Most of them are also implemented in the Cell versions:

[1]The only difference is the histogram, which is not included in the Cell version of hmmsearch.

Option	Status	Description
-h	Unchanged	Help
-n	Not implemented	Nucleic acid (default is protein)
-A <n>	Unchanged	Alignment output limit
-E <x>	Unchanged	E-value cutoff (default is 10)
-T <x>	Unchanged	Bit value cutoff (not limit per default)
-Z <n>	Unchanged	Number of models/sequences for E-value calculation
–compat	Not implemented	See HMMER User's Guide
–cpu <n>	Unchanged	See HMMER User's Guide
–cut_ga	Unchanged	See HMMER User's Guide
–cut_nc	Unchanged	See HMMER User's Guide
–cut_tc	Unchanged	See HMMER User's Guide
–domE <x>	Unchanged	See HMMER User's Guide
–domT <x>	Unchanged	See HMMER User's Guide
–forward	Not implemented	See HMMER User's Guide
–informat <s>	Not implemented	See HMMER User's Guide
–null2	Unchanged	See HMMER User's Guide
–pvm	Not implemented	See HMMER User's Guide
–xnu	Not implemented	See HMMER User's Guide

9.1.1 Examples of hmmpfam

To search the PFAM database with a fasta file containing protein sequences:

```
hmmpfam_cell pfam_database sequences.fasta
```

You can include a number of options in the search, e.g. setting the E-value cutoff for the results to be displayed to 1:

```
hmmpfam_cell –E 1 pfam_database sequence.fasta
```

Notice that you can use both a text-based and a binary version of the Pfam database. Using the binary version reduces the amount of time spent reading from the disk, since the binary database format is more compact than the text format.

An HMM database file can be converted to binary using the hmmconvert program:

```
hmmconvert –b Pfam_fs Pfam_fs.bin
\end{vebatim}

\subsection{Examples of hmmsearch}
To search with one profile against a database of protein sequences:
\begin{verbatim}
hmmsearch_cell hmm_profile sequence.fasta
```

You can include a number of options in the search, e.g. setting the E-value cutoff for the results to be displayed to 1:

```
hmmsearch_cell –E 1 hmm_profile sequence.fasta
```

Bibliography

[Smith and Waterman, 1981] Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *J Mol Biol*, 147(1):195–197.

[Thompson et al., 1994] Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994). Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, 22(22):4673–4680.

Index