CLC **Genomics** Server

Administrator Manual

Administrator Manual for
CLC Genomics Server 3.0
Windows, Mac OS X and Linux

January 26, 2011

This software is for research purposes only.

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

Quick installation guide

The following describes briefly the steps needed to set up a *CLC Genomics Server 3.0* with pointers to more detailed explanation of each step. If you are going to set up job nodes as well, please read chapter 6 first.

1. Download and run the server installer. As part of the installation, choose to start the server (section 2.1).
2. Run the license download script that is part of the installation (section 2.5).
3. The script will automatically download a license file and place it in the server installation directory under `licenses`.
4. Restart the server (section 2.6).
5. Log in to the server using a web browser (note that the default port is 7777) with username **root** and password **default** (section 3).
6. Change the root password (section 4.1).
7. Configure authentication mechanism and optionally set up users and groups (section 4.2).
8. Add data locations (section 3.1).
9. Download and install plug-ins *in the Workbench* needed for the Workbench to contact the server (section 2.7).
10. Check your server set-up using the **Check set-up** link in the upper right corner as described in section 8.1.
11. Your server is now ready for use.

Contents

Chapter 1

Introduction

The *CLC Server* is the central part of CLC bio's enterprise solutions. You can see an overview of the server solution in figure 1.1).

For documentation on the customization and integration, please see the developer kit for the server at `http://www.clcdeveloper.com`.

The *CLC Genomics Server* is shipped with the following tools and analyses that can be started from the *CLC Genomics Workbench*:

- Import tools for high-throughput sequencing data
- Map reads to reference
- De novo assembly
- SNP detection
- DIP detection
- Create detailed mapping report
- ChIP-Seq
- RNA-Seq
- Extract and count small RNA
- Annotate small RNA
- Process tagged sequences (de-multiplexing)
- External applications (integrating with 3rd party programs on the server)

1.1 System requirements

The system requirements of *CLC Genomics Server* are these:

- Windows 2000, Windows XP, Windows Vista or Windows 7

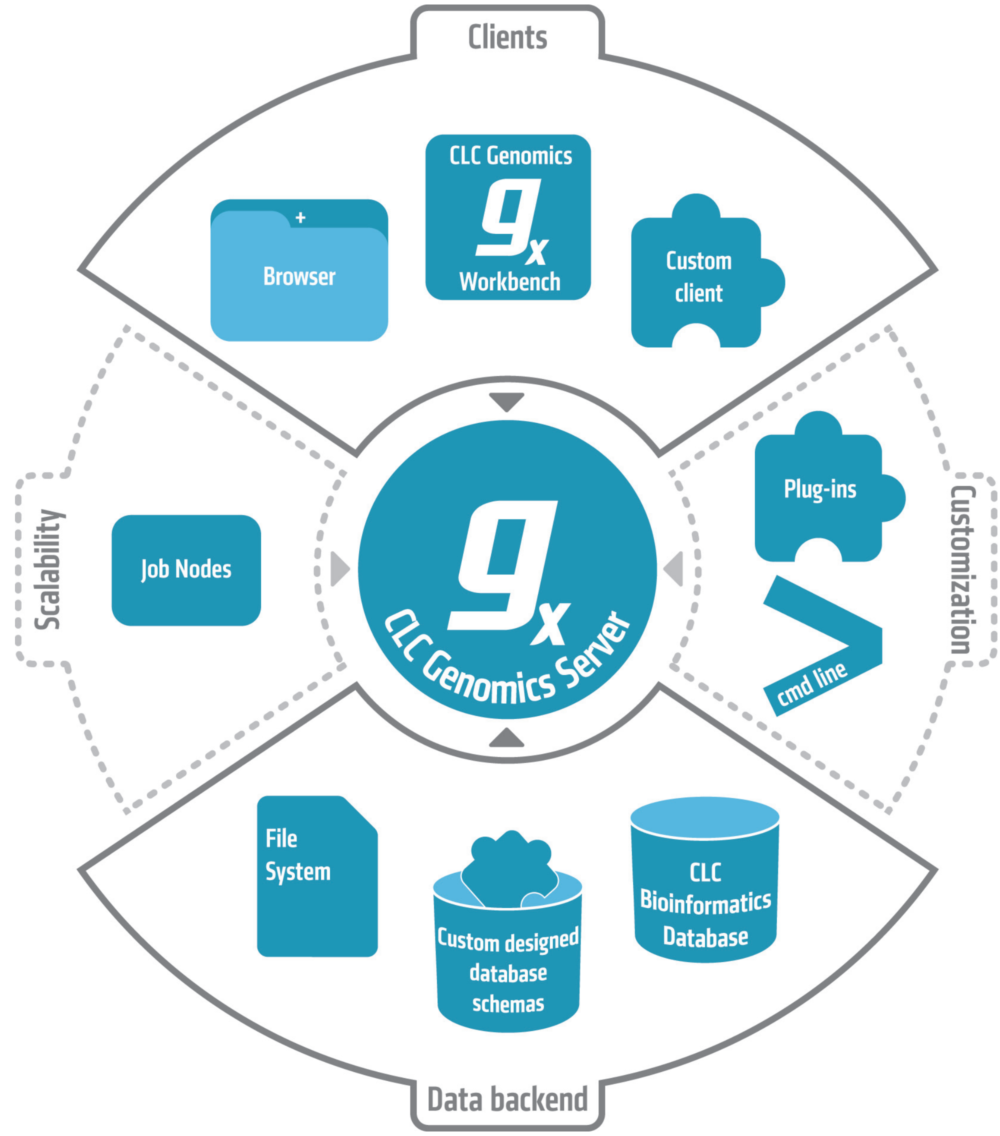


Figure 1.1: *An overview of the server solution from CLC bio. Note that not all features are included with all license models.*

- Mac OS X 10.4 or newer
- Linux: Redhat or SuSE
- Intel or AMD CPU required
- **Small data sets**. Assembly and analysis of genomes up to 50 mega-bases and up to 10 mil. reads
  - 2 GB RAM required,
  - 4 GB RAM recommended
- **Medium data sets**. Assembly and analysis of larger genomes and up to 100 mil. reads
  - 8 GB RAM required,
  - 16 GB RAM recommended

- **Large data sets**. Assembly and analysis of larger genomes and more than 100 mil. reads
    - 16 GB RAM required,
    - 32 GB RAM recommended
- **Special requirements for de novo assembly**. De novo assembly may need more memory than stated above - this depends both on the number of reads and the complexity and size of the genome. See `http://www.clcbio.com/white-paper` for examples of the memory usage of various data sets.
- 64 bit computer and operating system required to use more than 2GB RAM

1.2 Licensing

There are two kinds of licenses needed for running the *CLC Genomics Server*:

- A license for the server. This is needed for running the server. The license will allow a certain number of sessions (i.e. number of log-ins from e.g. the web interface or the workbench). The number of sessions is part of the agreement with CLC bio when you purchase a license.
- A license for the workbench. The Workbench is needed to start analyses on the server and view the results. Find the user manuals and deployment manual for the Workbenches at `http://www.clcbio.com/usermanuals`

The following chapter on installation will give you more information about how to obtain and deploy the license for the server.

Chapter 2

Installation

2.1 Installing the server

Double click on the installer to begin installation. Depending on your operating system you may be prompted for your password or asked to allow the installation to be performed. The installation can only be performed from a user with administrative privileges.

Please enter your password:

You must be a member of the Administrator group

Figure 2.1: *Enter your password.*

Note that not all operating systems will ask for a password.

Next you will be asked where to install the server. If you do not have a particular reason to change this, simply leave it at the default setting. The chosen directory will be referred to as the *server installation directory* throughout the rest of this manual.

The installer allows you to specify the maximum amount of memory the CLC Server will be able to utilize. The range of choice depends on the amount of memory installed on your system and on the type of machine used. On 32 bit machines you will not be able to utilize more than 2 GB of memory – on 64 bit machines there is no such limit.

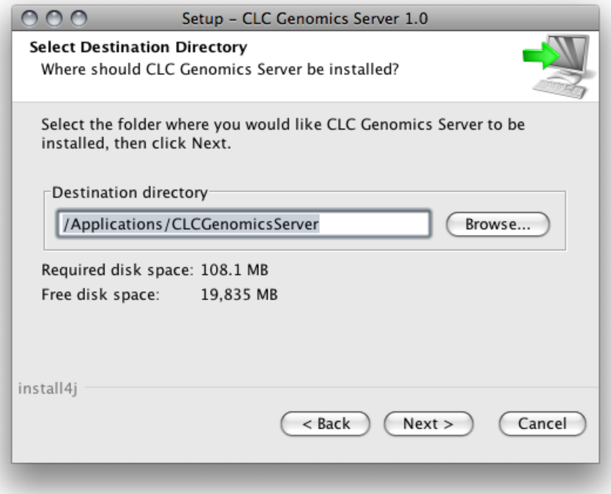
If you do not have a reason to change this value you should simply leave it at the default setting.

If you are installing the server on a Windows system you will be able to choose if the service is started manually or automatically by the system.

The installer will now extract the necessary files and start the service.

2.2 Silent installation

The installer also has a silent installation mode which is activated by the `-q` parameter when running the installer from a command line, e.g.



Setup - CLC Genomics Server 1.0

Select Destination Directory

Where should CLC Genomics Server be installed?

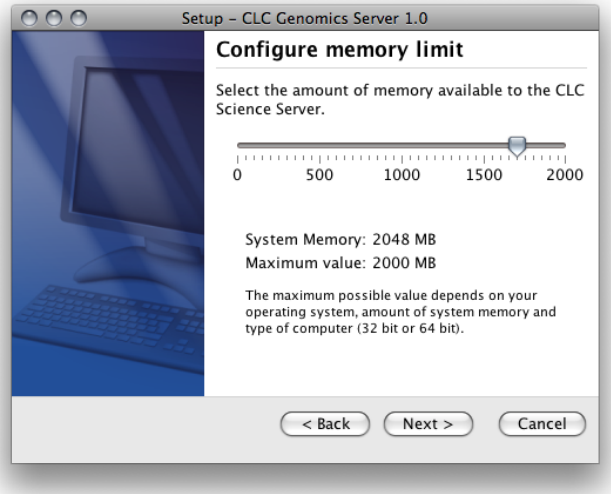
Select the folder where you would like CLC Genomics Server to be installed, then click Next.

Destination directory: /Applications/CLCGenomicsServer

Required disk space: 108.1 MB

Free disk space: 19,835 MB

Figure 2.2: *Choose where to install the server.*



Setup - CLC Genomics Server 1.0

Configure memory limit

Select the amount of memory available to the CLC Science Server.

System Memory: 2048 MB

Maximum value: 2000 MB

The maximum possible value depends on your operating system, amount of system memory and type of computer (32 bit or 64 bit).

Figure 2.3: *Choose the maximum amount of memory used by the server.*

```
CLCGenomicsServer_3_0.exe -q
```

On Windows, if you wish to have console output, `-console` can be appended *as the second parameter* (this is only needed when running on Windows where there is no output per default):

```
CLCGenomicsServer_3_0.exe -q -console
```

You can also in silent mode define a different installation directory: `-dir`.

```
CLCGenomicsServer_3_0.exe -q -console -dir "c:\bioinformatics\clc"
```

Note! Both the `-console` and the `-dir` options only work when the installer is run in silent mode.

The `-q` and the `-console` options work for the Uninstall program as well.

2.3 Upgrading an existing installation

Upgrading an existing installation is very simple. First, make sure that nobody are using the server (see section 4.4). A standard procedure would be to give users a notice in advance for system maintenance.

Next, simply install the server in the same installation directory as the one already installed. All settings will be maintained.

Note! When upgrading from (*CLC Genomics Server*) *2.X* to *3.0*, the settings for external applications (section 7) will be erased because these settings have been fundamentally changed.

For major versions (e.g. from (*CLC Genomics Server*) *2.0* to *3.0*), a new license needs to be downloaded (see section 2.5).

2.4 Allowing access through your firewall

The server listens for TCP-connections on port 7777 (See section 3.2 for info about changing this).

If you are running a firewall on your server system you will have to allow incoming TCP-connections on this port before your clients can contact the server from a Workbench or web browser. Consult the documentation of your firewall for information on how to do this.

Besides the public port described above the server also uses an internal port on 7776. There is no need to allow incoming connections from client machines to this port.

2.5 Downloading a license

The license server will look for licenses in the `licenses` folder. This means that all license files should be located in this folder. Check the platform-specific instructions below to see how to download a license file.

2.5.1 Windows license download

License files are downloaded using the `licensedownload` script. To run the script, right-click on the file and choose **Run as administrator**. This will present a window as shown in figure 2.4.

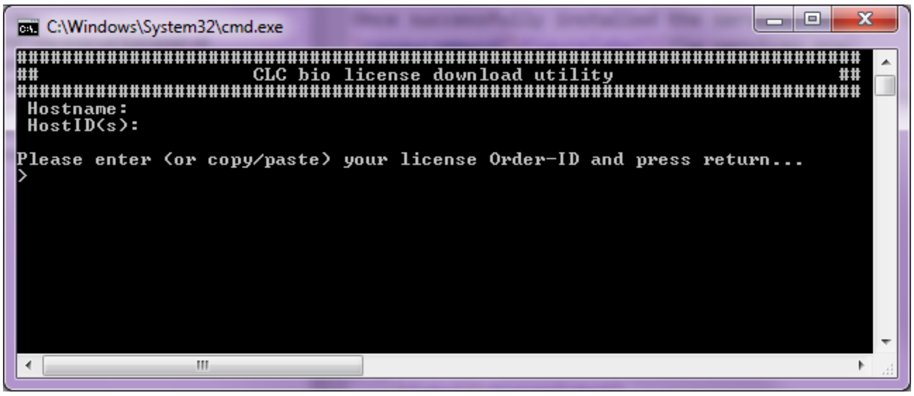
Paste the Order ID supplied by CLC bio (right-click to **Paste**) and press Enter. Please contact support@clcbio.com if you have not received an Order ID.

Note that if you are *upgrading* an existing license file, this needs to be deleted from the `licenses` folder. When you run the `licensedownload` script, it will create a new license file.

Restart the server for the new license to take effect (see how to restart the server in section 2.6.1).

2.5.2 Mac OS license download

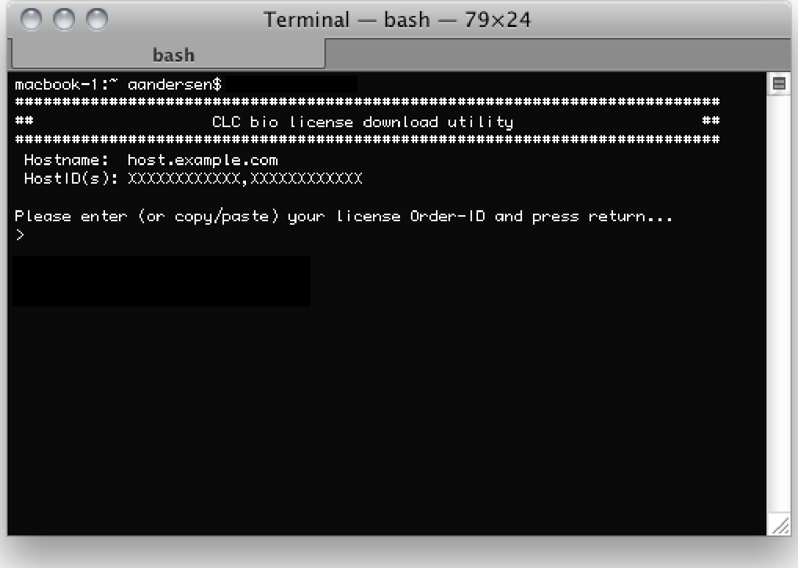
License files are downloaded using the `downloadlicense.command` script. To run the script, double-click on the file. This will present a window as shown in figure 2.5.



```
###########################################################################
##                  CLC bio license download utility                     ##
###########################################################################
 Hostname:
 HostID(s):

Please enter (or copy/paste) your license Order-ID and press return...
>
```

Figure 2.4: *Download a license based on the Order ID.*



```
macbook-1:~ aandersen$ [redacted]
###########################################################################
##                  CLC bio license download utility                     ##
###########################################################################
 Hostname:  host.example.com
 HostID(s): XXXXXXXXXXXX,XXXXXXXXXXXX

Please enter (or copy/paste) your license Order-ID and press return...
>

[redacted]
```

Figure 2.5: *Download a license based on the Order ID.*

Paste the Order ID supplied by CLC bio and press Enter. Please contact support@clcbio.com if you have not received an Order ID.

Note that if you are *upgrading* an existing license file, this needs to be deleted from the `licenses` folder. When you run the `downloadlicense.command` script, it will create a new license file.

Restart the server for the new license to take effect (see how to restart the server in section 2.6.2).

2.5.3 Linux license download

License files are downloaded using the `downloadlicense` script. Run the script and paste the Order ID supplied by CLC bio. Please contact support@clcbio.com if you have not received an Order ID.

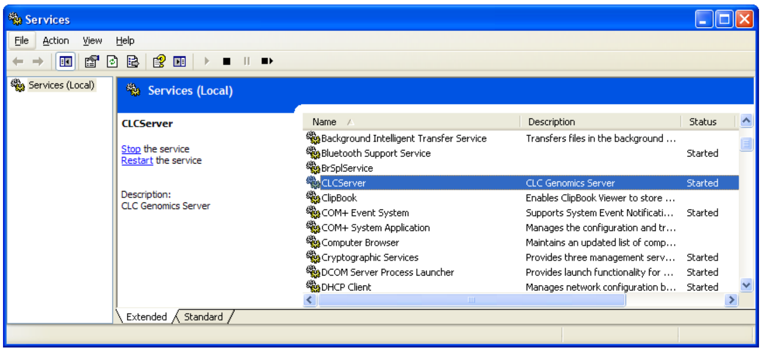
Note that if you are *upgrading* an existing license file, this needs to be deleted from the `licenses` folder. When you run the `downloadlicense` script, it will create a new license file.

Restart the server for the new license to take effect (see how to restart the server in section 2.6.3).

2.6 Starting and stopping the server

2.6.1 Microsoft Windows

On Windows based systems the *CLC Genomics Server* can be controlled through the *Services* control panel. Choose the service called `CLCGenomicsServer` and click the start, stop or restart link as shown in figure 2.6.



CLCServer

Stop the service
Restart the service

Description:
CLC Genomics Server

Name	Description	Status
Background Intelligent Transfer Service	Transfers files in the background ...	
Bluetooth Support Service		Started
BrSplService		
CLCServer	CLC Genomics Server	Started
ClipBook	Enables ClipBook Viewer to store ...	
COM+ Event System	Supports System Event Notificati...	Started
COM+ System Application	Manages the configuration and tr...	
Computer Browser	Maintains an updated list of comp...	
Cryptographic Services	Provides three management serv...	Started
DCOM Server Process Launcher	Provides launch functionality for ...	Started
DHCP Client	Manages network configuration b...	Started

Figure 2.6: *Stopping and restarting the server on Windows by clicking the blue links.*

2.6.2 Mac OS X

On Mac OS X the server can be started and stopped from the command line.

Open a terminal and navigate to the CLC Server installation directory. Once there the server can be controlled with the following commands.

To start the server run the command:

```
sudo ./CLCGenomicsServer start
```

To stop the server run the command:

```
sudo ./CLCGenomicsServer stop
```

To view the current status of the server run the command:

```
sudo ./CLCGenomicsServer status
```

2.6.3 Linux

On Linux the server can be started and stopped from the command line.

Open a terminal and navigate to the CLC Server installation directory. Once there the server can be controlled with the following commands.

To start the server run the command:

```
./CLCGenomicsServer start
```

To stop the server run the command:

```
./CLCGenomicsServer stop
```

To view the current status of the server run the command:

```
./CLCGenomicsServer status
```

2.7 Installing relevant plug-ins in the Workbench

In order to use the *CLC Genomics Server* from a CLC Workbench, you need to install a plug-in to the Workbench. If you connect to the server from a *CLC Genomics Workbench*, you should install the CLC Genomics Server Plug-in. If you are using e.g. *CLC Main Workbench*, you should install the generic CLC Server Plug-in. Both plug-ins allow you to log in to the server and access data from the server data locations, but the CLC Genomics Server Plug-in also includes the tools to access the analyses on the server (e.g. assembly, RNA-Seq).

The plug-ins can be found on the same web page as the server installer or at http://www.clcbio.com/plugins.

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

Help in the Menu Bar | Install Plug-ins ()

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

Install the plug-in 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 is provided by CLC bio on the page where you downloaded the server installation.

You need to restart the Workbench before the plug-in is ready for use.

Note that if you want users to be able to use **External applications** (see chapter 7) on the server, there is a separate plug-in (CLC External Applications Plug-in) that needs to be installed in the Workbench the same way as described above.

---

[1] In order to install plug-ins on Windows Vista, the Workbench must be run in administrator mode: Right-click the program shortcut and choose "Run as Administrator". Then follow the procedure described below. When you start the Workbench after installing the plug-in, it should also be run in administrator mode.

Chapter 3

Configuring and administering the server

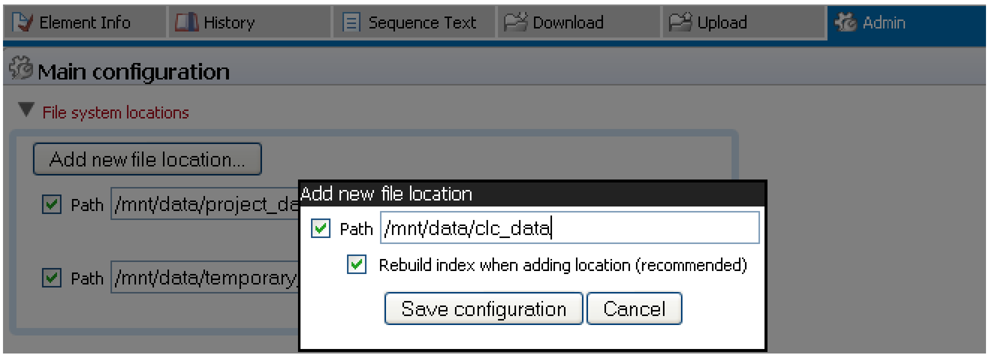
Once the server is running, you can log in via a web browser. Simply type the host name of the server followed by the port (default is 7777). An example would be `http://clccomputer:7777/`. First time you log in, username is **root** and password is **default**.

3.1 Adding locations for saving data

Before you can use the server for doing analyses you will need to add one or more locations for storing your data.

3.1.1 Adding a file system location

Under the **File system locations** heading, click the **Add New File Location** button to add a new file system location (see figure 3.1).



☑ Path /mnt/data/project_da

☑ Path /mnt/data/temporary

☑ Path /mnt/data/clc_data

☑ Rebuild index when adding location (recommended)

Figure 3.1: *File system location settings.*

In this dialog, enter the path to the folder you want to use for storing the data. All the data is in clc format and this folder should be dedicated for use by the *CLC Genomics Server*. This folder should be directly accessed only by the server - not the Workbenches. The Workbenches have to access the location by logging in to the server.

The path should point to an existing folder, and the user running the server process (not the user logging in to the server) needs to have read and write access to the folder.

Once you have pressed **Save Configuration** (learn more about rebulding the index in section

3.1.2), this location will be added and it should now appear in the **Navigation Area** in the left hand side of the window, and it will appear in the Workbench on next login. You can use the checkbox next to the location to determine whether it should be shown for the users.

Note that pressing **Remove Location** will only remove the location from this list - it will not affect any data already stored in this folder. The data will be accessible again simply by adding the folder as a new location again.

File locations for job node set-ups

When you have a job node set-up, all the job node computers need to have access to the same data location folder. This is because the job nodes will write files directly to the folder rather than passing through the master node (which would be a bottleneck for big jobs). Furthermore, the user running the server (not the user logged in to the server) must be the same for all the job nodes and it needs to act as the same user when accessing the folder no matter whether it is a job node or a master node.

The data location should be added **after** the job nodes have been configured and attached to the master node. In this way, all the job nodes will inherit the configurations made on the master node.

One relatively common problem faced in this regard is *root squashing* which often needs to be disabled, because it prevents the servers from writing and accessing the files as the same user - read more about this at `http://nfs.sourceforge.net/#faq_b11`.

In order to test that access works for both job nodes and the master node, you can perform the following check:

1. Create a new folder in the Workbench in the new data location (on disk, this will be done by the master node). Import some data into this folder.
2. In the Workbench, run one of the analyses in the server toolbox and choose to save the results in the new folder (this will be done by the job node)
3. Once finished, try to rename one of the result files in the Workbench (on disk, this will be done by the master node). If this works, it means that both master and job nodes are able to write to the same files.

3.1.2 Rebuilding the index

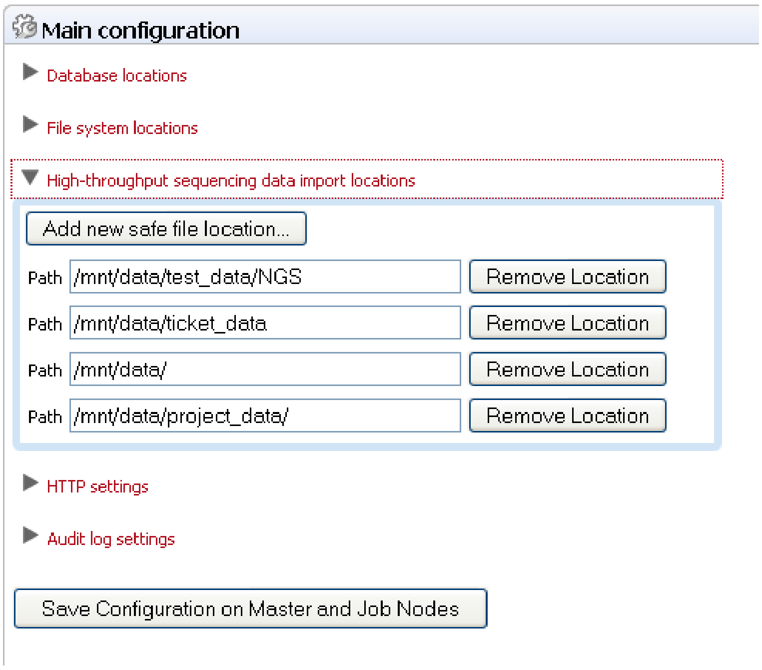
The server maintains an index of all the elements in the data locations. The index is used when searching for data. For all locations you can choose to **Rebuild Index**. This should be done only when a new location is added or if you experience problems while searching (e.g. something is missing from the search results). This operation can take a long time depending on how much data is stored in this location.

If you move the server from one computer to another, you need to move the index as well. Alternatively, you can re-build the index on the new server (this is the default option when you add a location). If the rebuild index operation takes too long time and you prefer to move the old index, simply copy the folder called `searchindex` from the old server installation folder to the new server.

The status of the index server can be seen in the **User Statistics** pane showing information on where the index server resides and the number of locations currently being serviced.

manual-server/sections/safe-files-import

The folders that should be available for browsing files from the Workbench are specified in the web interface in the **Main configuration** tab under **High-throughput sequencing data import locations** (see figure 3.2).



Path /mnt/data/test_data/NGS

Path /mnt/data/ticket_data

Path /mnt/data/

Path /mnt/data/project_data/

Figure 3.2: *Defining source folders that should be available for browsing from the Workbench.*

. Press the **Add new import location** button to specify a path to a folder on the server. This folder and all subfolders will then be available for browsing in the Workbench.

The idea is that you can specify a folder dedicated for storing the sequencing data that will eliminate the need for transferring files that are already accessible from the server to the Workbench computer.

Note that the server is accessing the file system as the user running the server - not as the user who is logged in. This means that you should be careful when opening the server filesystem in this way. For security reasons only folders that do not contain sensitive information should be added.

3.2 Changing the listening port

The default listening port for the CLC Server is 7777. This has been chosen to minimize the risk of collisions with existing web-servers using the more familiar ports 80 and 8080. If you would like to have the server listening on port 80 in order to simplify the URL, this can be done in the following way.

- Navigate to the CLC Server installation directory.
- Locate the file called *server.xml* in the conf directory.

- Open the file in a text editor and locate the following section

```
<Connector port="7777" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
```

- Change the port value to desired listening port (80 in the example below)

```
<Connector port="80" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
```

- Restart the service for the change to take effect (see how to restart the server in section 2.6).

3.3 Changing the tmp directory

Since the *CLC Genomics Server* often uses a lot of disk space for temporary files, it can be convenient to specify a special tmp directory with enough disk space (the tmp files are of course deleted when they are not needed anymore).

In the *server installation directory* you will find a file called `CLCGenomicsServer.vmoptions`. Open this file in a text editor and add a new line like this: `-Djava.io.tmpdir=/path/to/tmp` with the path the new tmp directory. Restart the server for the change to take effect (see how to restart the server in section 2.6).

3.4 Other configurations

3.4.1 HTTP settings

Under the **Admin** (⚙) tab, click **Configuration**, and you will be able to specify HTTP settings. Here you can set the time out for the user HTTP session and the maximum upload size (when uploading files through the web interface).

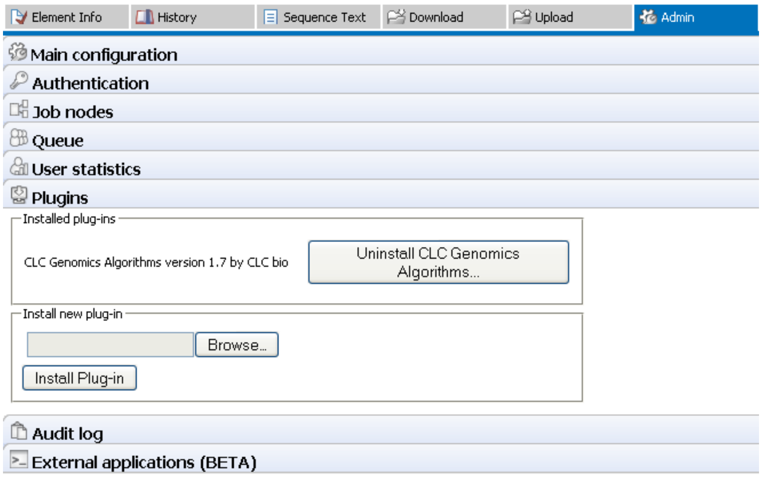
3.4.2 Audit log settings

The audit log records all the actions performed in the web interface and through the Workbenches. Note that data management operations (copying, deleting and adding files) done through the Workbench are not recorded.

Under the **Admin** (⚙) tab, click **Configuration**, and you will be able to specify settings for the **Audit log**. Here you can specify the **Log type** and the **Database type** to use for the audit log.

3.4.3 Deployment of server information to CLC Workbenches

See the *Deployment manual* at `http://www.clcbio.com/usermanuals` for information on pre-configuring the server log-in information when Workbench users log in for the first time.



CLC Genomics Algorithms version 1.7 by CLC bio

Figure 3.3: *Installing and uninstalling server plugins.*

3.5 Server plug-ins

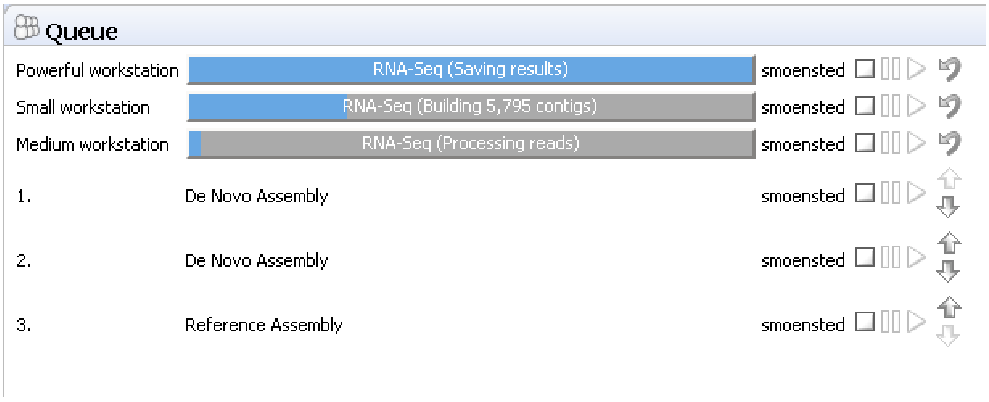
You can install plug-ins on the server under the **Admin** (⚙) tab (see figure 3.3).

Click the **Browse** button and locate the plug-in .cpa file to install a plug-in. To uninstall a plug-in, simply click the button next to the plug-in. The server does not need to be restarted after installation/uninstallation of plug-ins.

Read more about developing server plug-ins at `http://www.clcdeveloper.com/`.

3.6 Queue

Clicking the **Queue** panel will show a list of all the processes that are currently in the queue (including the one in progress). An example is shown in figure 3.4.



Powerful workstation	RNA-Seq (Saving results)	smoensted
Small workstation	RNA-Seq (Building 5,795 contigs)	smoensted
Medium workstation	RNA-Seq (Processing reads)	smoensted
1.	De Novo Assembly	smoensted
2.	De Novo Assembly	smoensted
3.	Reference Assembly	smoensted

Figure 3.4: *The process queue.*

For each process, you are able to **Cancel** (☐) and re-prioritize the order of the processes by clicking the up and down arrows. Some processes also allow you to pause and resume. At the top, you can see the progress of the process that is currently running.

You can also **Stop and requeue** (↶) a job that is currently being processed (this is only possible if you have a job node set-up)..

Chapter 4

Managing users and groups

4.1 Logging in the first time - root password

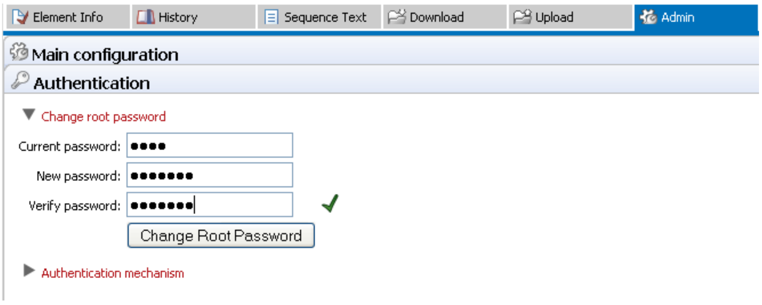
When the server is installed, you will be able to log in using the root password. In order to configure the server for use, please log in through the web interface using the root password:

- **User name**: `root`
- **Password**: `default`

Once logged in, you should as a minimum set up user authentication (see section 4.2) and data locations (see section 3.1) before you can start using the server.

For security reasons, you should change the root password (see figure 4.1):

Admin (⚙) | Authentication (🔑) Change root password



Current password: ●●●●

New password: ●●●●●●●

Verify password: ●●●●●●● ✓

Figure 4.1: *We recommend changing the root password away from default. The verification of the root password is shown with the green checkmark.*

Note that if you are going to use job nodes, it makes sense to set this up first before changing the authentication mechanism and root password (see section 6).

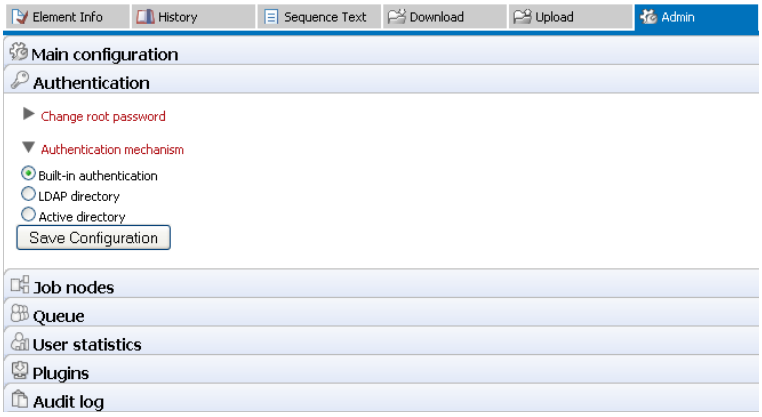
4.2 User authentication using the web interface

When the server is installed, you can log in using the default root password (username=root, password=default). Note that if you are going to use job nodes, it makes sense to set this up first before changing the authentication mechanism and root password (see section 6).

Once logged in, you can specify how the general user authentication should be done:

Admin (⚙) | Authentication (🔑) Authentication mechanism

This will reveal the three different modes of authentication as shown in figure 4.2.



● Built-in authentication
○ LDAP directory
○ Active directory

Figure 4.2: *Three modes of user authentication.*

The options are:

- **Built-in authentication**. This option will enable you to set up user authentication using the server's built-in user management system. This means that you create users, set passwords, assign users to groups and manage groups using the web interface (see section 4.2.1) or using the Workbench (see section 4.3.1). All the user information is stored on the server and is not accessible from other systems.
- **LDAP directory**. This option will allow you to use an existing LDAP directory. This means that all information needed during authentication and group memberships is retrieved from the LDAP directory.
- **Active directory**. This option will allow you to use an existing Active directory which is Microsoft's LDAP counterpart. This means that all information needed during authentication and group memberships is retrieved from the Active directory.

For the two last options, a settings panel will be revealed when the option is chosen, allowing you to specify the details of the integration.

Note that membership of the admin group is used for allowing users access to the admin part of the web interface.

4.2.1 Managing users using the web interface

To create or remove users or change their password:

Admin (⚙) | Users and groups (👥) Manage user accounts

This will display the panel shown in figure 4.3.

4.2.2 Managing groups using the web interface

To create or remove groups or change group membership for users:

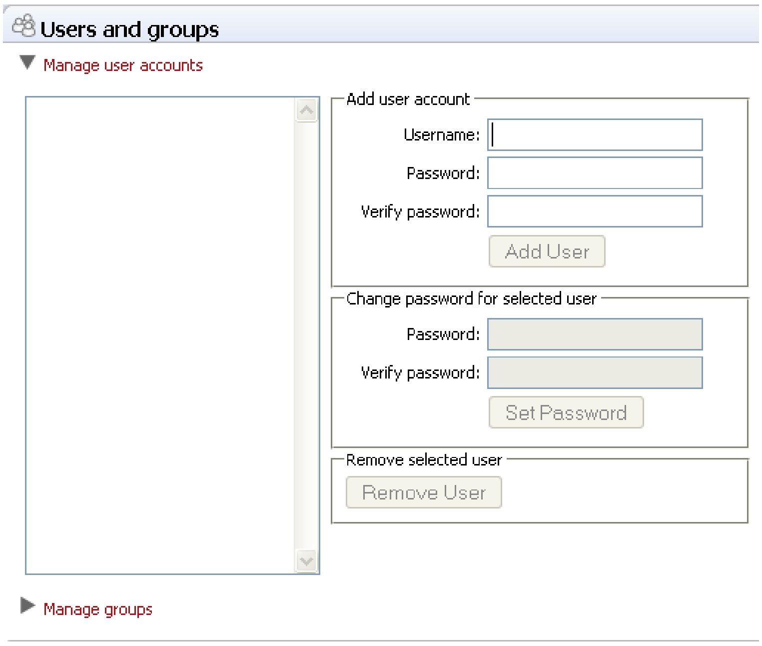


Figure 4.3: *Managing users.*

Admin () | Users and groups () Manage groups

This will display the panel shown in figure 4.4.

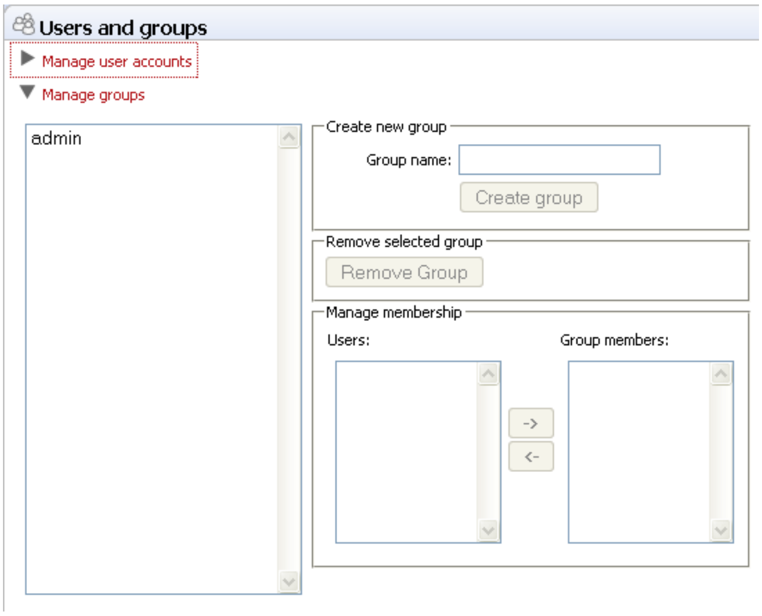


Figure 4.4: *Managing users.*

The same user can be a member of several groups.

Note that membership of the admin group is used for allowing users access to the admin part of the web interface. Users who should have access to the administrative part of the server should be part of the "admin" group which is the only special group (this group is already created for you).

Note that you will always be able to log as root with administrative access.

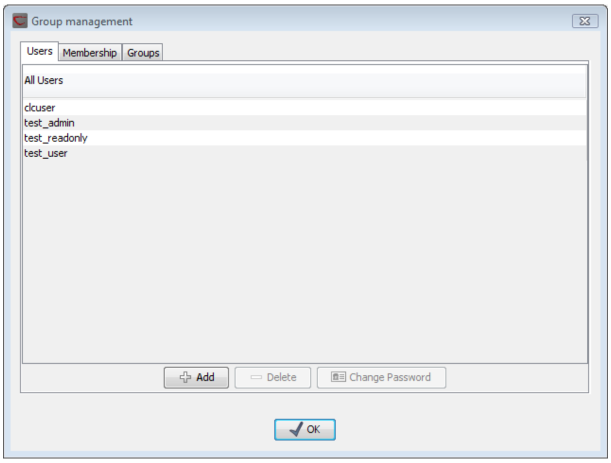
The functionality of this plug-in depends on the user authentication and management system: if the built-in system is used, all the functionality described below is relevant; if an external system is used for managing users and groups, the menus below will be disabled.

4.3 User authentication using the Workbench

Users and groups can also be managed through the Workbench (note that you need to set up the authentication mechanism as described in section 4.2):

File | Manage Users and Groups

This will display the dialog shown in figure 4.5.



All Users

clcuser
test_admin
test_readonly
test_user

Figure 4.5: *Managing users.*

4.3.1 Managing users through the Workbench

Click the **Add** (✚) button to create a new user. Enter the name of the user and enter a password. You will be asked to re-type the password. If you wish to change the password at a later time, select the user in the list and click **Change password** (▤).

To delete a user, select the user in the list and click **Delete** (▭).

4.3.2 Managing groups through the Workbench

Access rights are granted to groups, not users, so a user has to be member of one or more groups to get access to the data location. Here you can see how to add and remove groups, and next you will see how to add users to a group.

Adding and removing groups is done in the **Groups** tab (see figure 4.6).

To create a new group, click the **Add** (✚) button and enter the name of the group. To delete a group, select the group in the list and click the **Delete** (▭) button.

4.3.3 Adding users to a group

When a new group is created, it is empty. To assign users to a group, click the **Membership** tab. In the **Selected group** box, you can choose among all the groups that have been created. When

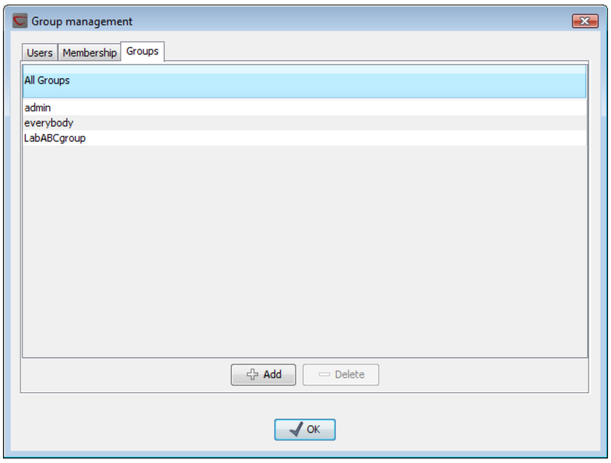


Figure 4.6: *Managing groups.*

you select a group, you will see its members in the list below (see figure 4.7). The the left you see a list of all users.

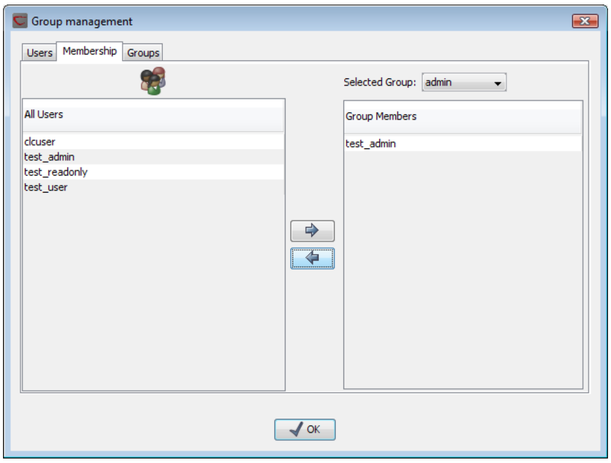

Selected Group: admin

Figure 4.7: *Listing members of a group.*

To add or remove users from a group, click the **Add** (➡) or **Remove** (⬅) buttons. To create new users, see section 4.3.1.

The same user can be a member of several groups.

4.4 User statistics

Clicking the **User statistics** panel will show a summary of the current usage of the server. An example is shown in figure 4.8.

You can see the number of users currently logged in, and you can see the number of sessions for each user. The two green dots indicate that this user is logged in twice (e.g. through the Workbench and through the web interface). The other two users have been logged in previously.

You can also log users off by expanding the user sessions on the + sign and the click **Invalidate Session...**. This will open confirmation dialog where you can also write a message to the user that will displayed either in the Workbench or the browser.

Number of users logged in: 1
Number of logins: 2

Figure 4.8: *The user statistics (user names have been blurred).*

Chapter 5

Customized attributes on server data locations

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

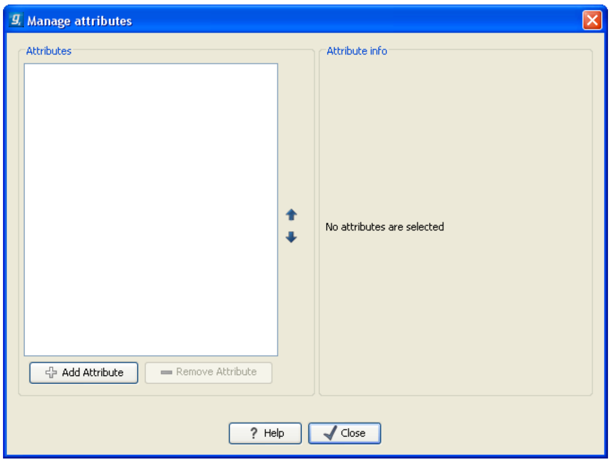
5.1 Configuring which fields should be available

To configure which fields that should be available, you need to log in as administrator via the Workbench.

Log in as administrator and:

right-click the data location (📁) | Location | Attribute Manager

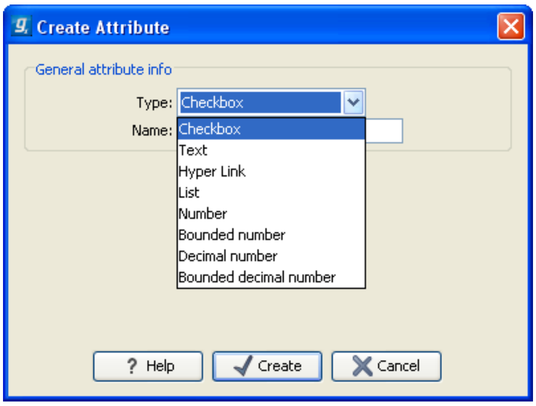
This will display the dialog shown in figure 5.1.



No attributes are selected

Figure 5.1: *Adding attributes.*

Click the **Add Attribute** (✚) button to create a new attribute. This will display the dialog shown in figure 5.2.



Type: Checkbox

Name:

Figure 5.2: *The list of attribute types.*

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

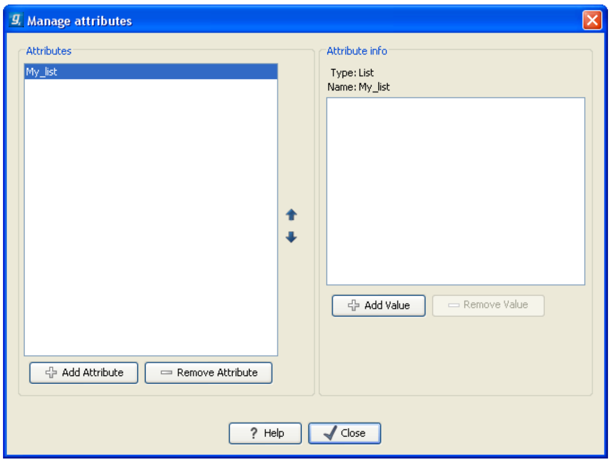
- **Checkbox**. This is used for attributes that are binary (e.g. true/false, checked/unchecked and yes/no).
- **Text**. For simple text with no constraints on what can be entered.
- **List**. Lets you define a list of items that can be selected (explained in further detail below).
- **Number**. Any positive or negative integer.
- **Bounded number**. Same as number, but you can define the minimum and maximum values that should be accepted. If you designate some kind of ID to your sequences, you can use the bounded number to define that it should be at least 1 and max 99999 if that is the range of your IDs.
- **Decimal number**. Same as number, but it will also accept decimal numbers.
- **Bounded decimal number**. Same as bounded number, but it will also accept decimal numbers.
- **Hyper Link**. This can be used if the attribute is a reference to a web page. A value of this type will appear to the end user as a hyper link that can be clicked. Note that this attribute can only contain one hyper link. If you need more, you will have to create additional attributes.

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

5.1.1 Editing lists

Lists are a little special, since you have to define the items in the list. When you click a list in the left side of the dialog, you can define the items of the list in the panel to the right by clicking **Add Item** (➕) (see figure 5.3).

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



Type: List
Name: My_list

Figure 5.3: *Defining items in a list.*

5.1.2 Removing attributes

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

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

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

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

5.1.3 Changing the order of the attributes

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

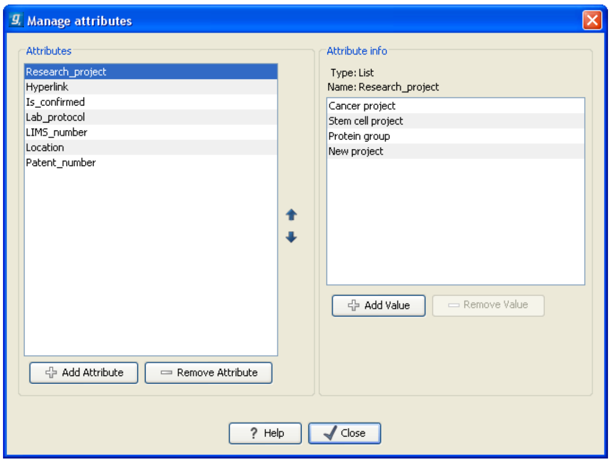
5.2 Filling in values

When a set of attributes has been created (as shown in 5.4), the end users can start filling in information.

This is done in the element info view:

> **select a sequence or another element in a server data location | Show () in the Toolbar | Element info ()**

This will open a view similar to the one shown in figure 5.5.



Attributes:

- Research_project
- Hyperlink
- Is_confirmed
- Lab_protocol
- LIMS_number
- Location
- Patent_number

Type: List
Name: Research_project

- Cancer project
- Stem cell project
- Protein group
- New project

Figure 5.4: *A set of attributes defined in the attribute manager.*

Latin name Edit
Homo sapiens

Common name Edit
human

Taxonomy name Edit
Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Mammalia; Eutheria; Euarchontoglires; Primates; Haplorrhini; Catarrhini; Hominidae; Homo.

Research project Clear
Cancer project

Patent number Clear
11.782

Is confirmed Clear
☑

LIMS number Clear
204

Location Clear
Lab 23

Lab_instructions Edit Clear

Hyperlink Edit Clear
http://pfam.sanger.ac.uk/family?acc=PF03143

Figure 5.5: *Adding values to the attributes.*

You can now enter the appropriate information and **Save**. When you have saved the information, you will be able to search for it (see below). Note that the sequence needs to be saved in the data location before you can edit the attribute values.

When nobody has entered information, the attribute will have a "Not set" written in red next to the attribute (see figure 5.6).

Research_project Clear
Cancer project Not set

Figure 5.6: *An attribute which has not been set.*

This is particularly useful for attribute types like checkboxes and lists where you cannot tell from the displayed value if it has been set or not. Note that when an attribute has not been set, you can't search for it, even if it looks like it has a value. In figure 5.6, you will *not* be able to find this sequence if you search for research projects with the value "Cancer project", because it has not been set. To set it, simply click in the list and you will see the red "Not set" disappear.

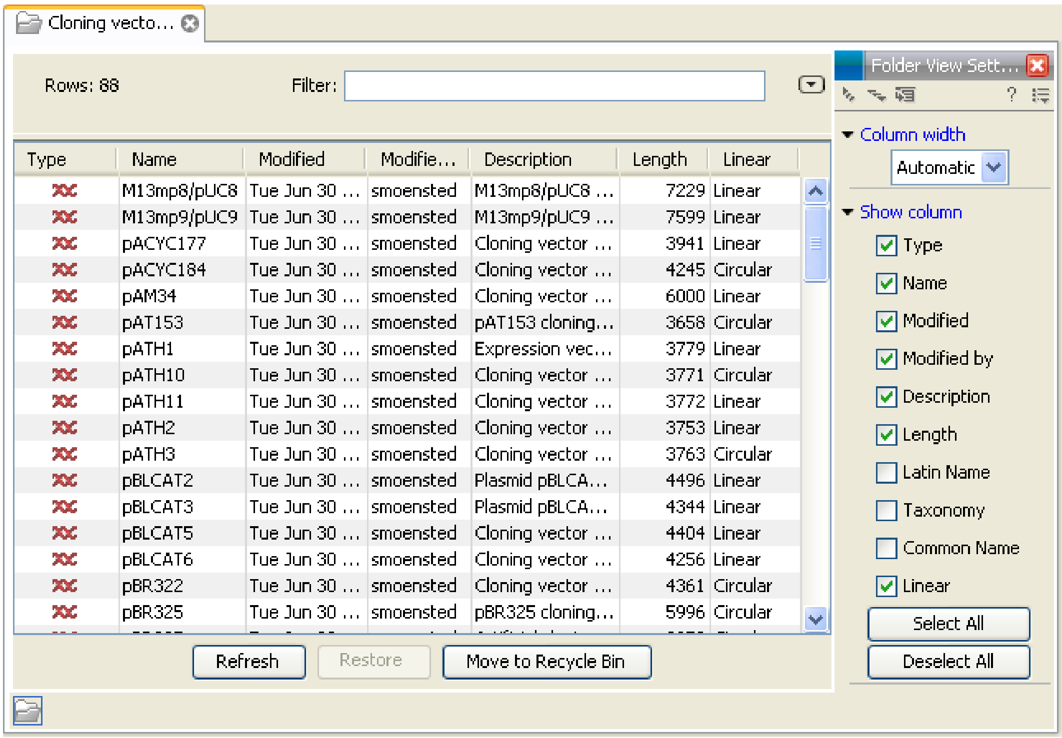
If you wish to reset the information that has been entered for an attribute, press "Clear" (written in blue next to the attribute). That will return it to the "Not set" state.

5.2.1 Show folder elements in a table

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

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

An example is shown in figure 5.7.



Rows: 88 Filter:

Type	Name	Modified	Modifie...	Description	Length	Linear
	M13mp8/pUC8	Tue Jun 30 ...	smoensted	M13mp8/pUC8 ...	7229	Linear
	M13mp9/pUC9	Tue Jun 30 ...	smoensted	M13mp9/pUC9 ...	7599	Linear
	pACYC177	Tue Jun 30 ...	smoensted	Cloning vector ...	3941	Linear
	pACYC184	Tue Jun 30 ...	smoensted	Cloning vector ...	4245	Circular
	pAM34	Tue Jun 30 ...	smoensted	Cloning vector ...	6000	Linear
	pAT153	Tue Jun 30 ...	smoensted	pAT153 cloning...	3658	Circular
	pATH1	Tue Jun 30 ...	smoensted	Expression vec...	3779	Linear
	pATH10	Tue Jun 30 ...	smoensted	Cloning vector ...	3771	Circular
	pATH11	Tue Jun 30 ...	smoensted	Cloning vector ...	3772	Linear
	pATH2	Tue Jun 30 ...	smoensted	Cloning vector ...	3753	Linear
	pATH3	Tue Jun 30 ...	smoensted	Cloning vector ...	3763	Circular
	pBLCAT2	Tue Jun 30 ...	smoensted	Plasmid pBLCA...	4496	Linear
	pBLCAT3	Tue Jun 30 ...	smoensted	Plasmid pBLCA...	4344	Linear
	pBLCAT5	Tue Jun 30 ...	smoensted	Cloning vector ...	4404	Linear
	pBLCAT6	Tue Jun 30 ...	smoensted	Cloning vector ...	4256	Linear
	pBR322	Tue Jun 30 ...	smoensted	Cloning vector ...	4361	Circular
	pBR325	Tue Jun 30 ...	smoensted	pBR325 cloning...	5996	Circular

Column width
Automatic

Show column
☑ Type
☑ Name
☑ Modified
☑ Modified by
☑ Description
☑ Length
☐ Latin Name
☐ Taxonomy
☐ Common Name
☑ Linear

Figure 5.7: *Viewing the elements in a folder.*

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

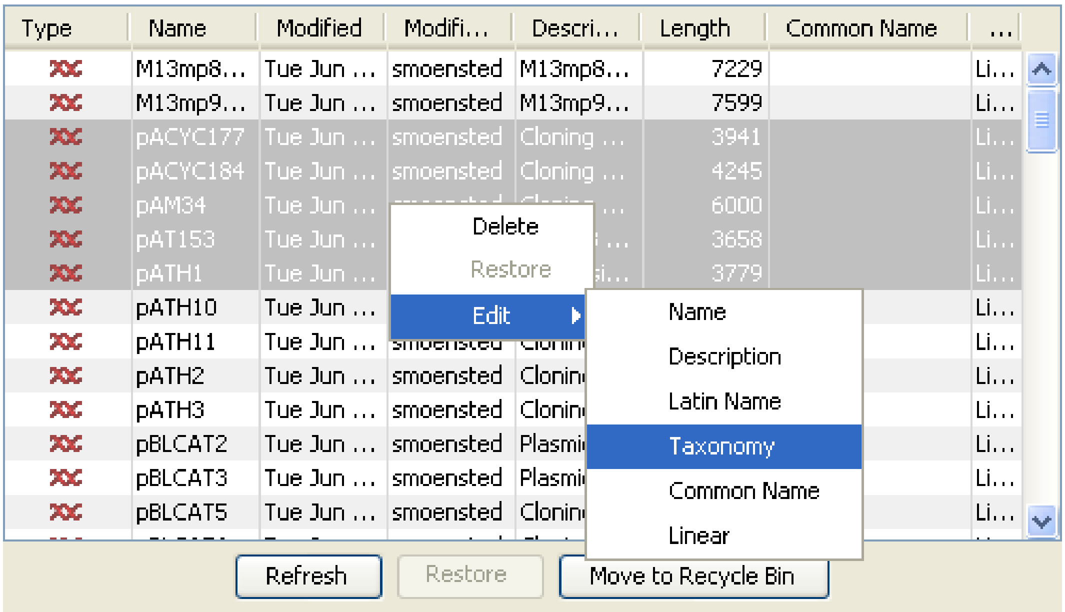
Sorting the elements in a view does not affect the ordering of the elements in the **Navigation Area**.

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

Batch edit folder elements

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

In figure 5.8 you can see an example where the common name of five sequence are renamed in one go. In this example, a dialog with a text field will be shown, letting you enter a new common name for these five sequences. **Note!** This information is directly saved and you cannot undo.



Type	Name	Modified	Modifi...	Descri...	Length	Common Name	...
	M13mp8...	Tue Jun ...	smoensted	M13mp8...	7229		Li...
	M13mp9...	Tue Jun ...	smoensted	M13mp9...	7599		Li...
	pACYC177	Tue Jun ...	smoensted	Cloning ...	3941		Li...
	pACYC184	Tue Jun ...	smoensted	Cloning ...	4245		Li...
	pAM34	Tue Jun ...		...	6000		Li...
	pAT153	Tue Jun ...		...	3658		Li...
	pATH1	Tue Jun ...		i...	3779		Li...
	pATH10	Tue Jun ...					Li...
	pATH11	Tue Jun ...					Li...
	pATH2	Tue Jun ...	smoensted	Clonin			Li...
	pATH3	Tue Jun ...	smoensted	Clonin			Li...
	pBLCAT2	Tue Jun ...	smoensted	Plasmi			Li...
	pBLCAT3	Tue Jun ...	smoensted	Plasmi			Li...
	pBLCAT5	Tue Jun ...	smoensted	Clonin			Li...

Figure 5.8: *Changing the common name of five sequences.*

5.2.2 What happens when the sequence gets outside the server data location?

Since the information entered in the **Element info** is very tightly connected to the attributes set in the data location, they will appear in a different way when it is stored in another location. When you save the sequence in another server data location with a different attribute set, or on the Workbench's own data location, the information will become "static" which means that it can't be changed and searched for. It can only be deleted. Note that attributes that were "Not set" will disappear when you go outside the data location.

If the sequence is moved back into the data location, the information will be available for editing and searching again.

5.3 Searching

When an attribute has been created, it will automatically be available for searching. This means that in the **Local Search** (), you can select the attribute in the list of search criteria (see figure 5.9).

It will also be available in the **Quick Search** below the **Navigation Area** (press Shift+F1 and it will be listed - see figure 5.10).

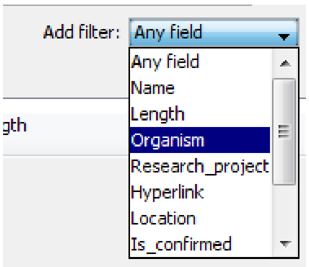


Figure 5.9: *The attributes from figure* 5.4 *are now listed in the search filter.*

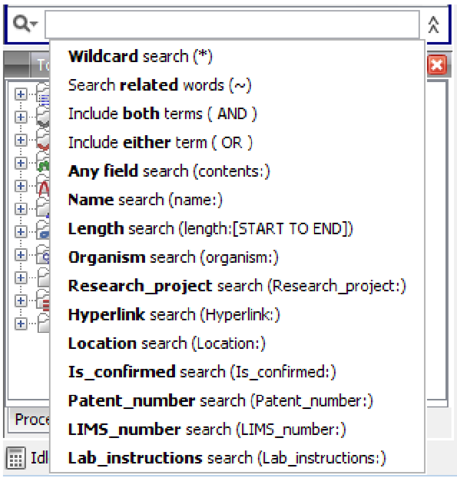


Figure 5.10: *The attributes from figure* 5.4 *are now available in the Quick Search as well.*

Chapter 6

Job nodes

The *CLC Server* has the concept of *Job nodes*. This means that you have a master server and one or more job nodes which are separate computers that are also running the *CLC Server*. When the user logs in to the master server via the Workbench and starts a job on the server, the job is distributed to an available job node. The responsibility of the master server is to manage the queue and act as a middle layer serving data to the Workbenches. It will not run any jobs itself. The job nodes, on the other hand, are waiting for the master node to assign a job, and once the job is executed, they are available for the next job.

The benefit of this set-up is that you can spread out the work load by having several computers doing calculations. Each single job is not executed faster in this set-up, since the job is not divided and parallelized on several computers. Rather, the idea is that when you have many jobs, they do not have to wait in the queue so long.

Another advantage of having job nodes is that you can specialize the nodes to only accept certain kinds of jobs. This means that you can dedicate a node to e.g. import. This means that import jobs will not wait in the queue for other kinds of jobs to be executed.

6.1 Setting up job nodes

When setting up job nodes, you need first of all to install the *CLC Server* on all the computers involved (see section 2). Note that you do not need to get license files for the job nodes - this is controlled by the master node.

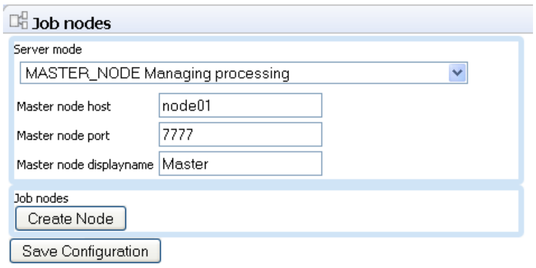
When the server has been installed and is running on all job nodes, you need to make sure that the authentication is identical for all nodes, including the master nodes (you can log in to the job nodes using the web interface and change this). From this point, all configuration takes place on the master server. To attach job nodes to the master:

> **Admin ()** | **Job nodes ()**

First, set the server mode to `MASTER_NODE` and fill in information about the master node, port and a human-readable name as shown in figure 6.1).

Next, click **Attach Node** to specify a job node. Fill in the appropriate information about the node (see figure 6.2).

Besides information about the node hostname, port and displayname, you can also configure



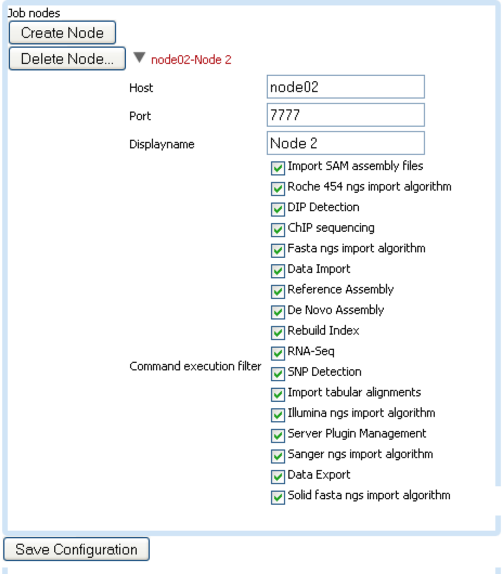
Server mode: MASTER_NODE Managing processing

Master node host: node01

Master node port: 7777

Master node displayname: Master

Figure 6.1: *Setting up a master server.*



Host: node02

Port: 7777

Displayname: Node 2

Command execution filter:

- ☑ Import SAM assembly files
- ☑ Roche 454 ngs import algorithm
- ☑ DIP Detection
- ☑ ChIP sequencing
- ☑ Fasta ngs import algorithm
- ☑ Data Import
- ☑ Reference Assembly
- ☑ De Novo Assembly
- ☑ Rebuild Index
- ☑ RNA-Seq
- ☑ SNP Detection
- ☑ Import tabular alignments
- ☑ Illumina ngs import algorithm
- ☑ Server Plugin Management
- ☑ Sanger ngs import algorithm
- ☑ Data Export
- ☑ Solid fasta ngs import algorithm

Figure 6.2: *Setting up a master server.*

what kind of jobs that this node should be able to execute.

Repeat this process for each job node you wish to attach and click **Save Configuration** when you are done.

Note that you will get a warning dialog if there are jobs that are not enabled on any of the job nodes.

The configuration of the job nodes affect the way the queue works. When a node has finished a job, it will take the first job in the queue that it is configured to process. This also means that it is not necessarily the job that is number one in the queue that will be processed first.

Once set up, the job nodes will automatically inherit all configurations made on the master node. At any time, you can log in to the job nodes themselves to check their configuration.

Chapter 7

External applications

Third-party command line applications (external applications) can easily be integrated with the *CLC Server*. The integration is configured in the web interface, and a special plug-in is installed in the client Workbench to run the command line application using data from the CLC environment. The idea is that server administrators can make command-line based tools available in the Workbench for end users that will use the familiar Workbench user interface and their own data. The actual tool will be executed on the server, so the administrator will have full control over the execution environment.

Please contact support@clcbio.com to get access to the Workbench plug-in needed to use the external applications.

Figure 7.1 shows an overview of the technical details of the external applications integration.

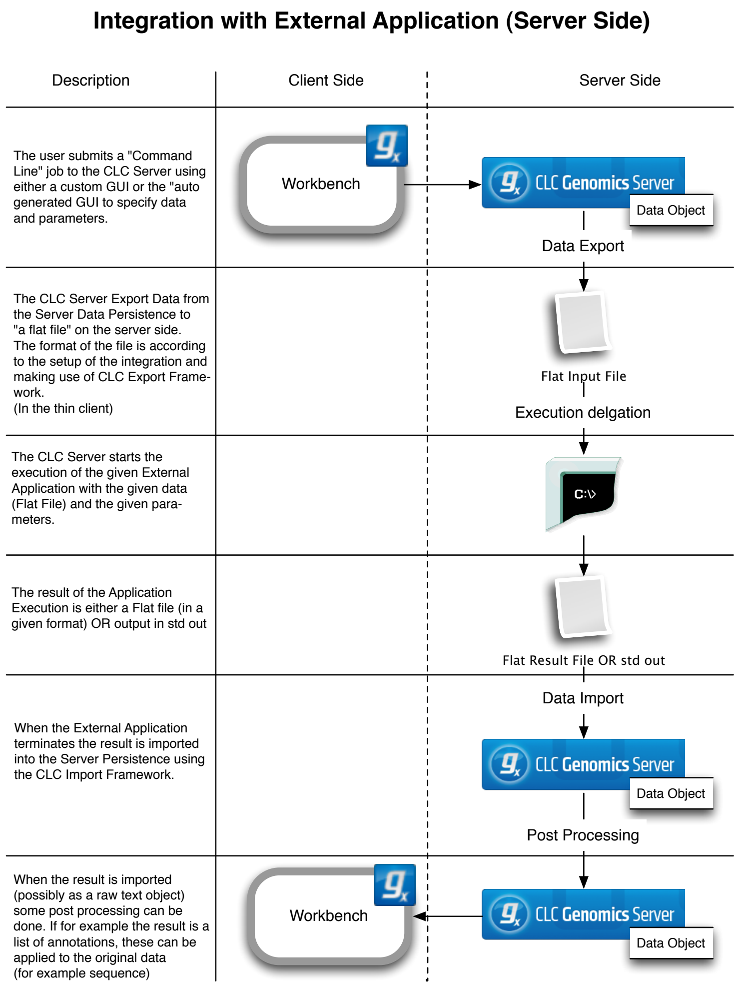
When the integration is configured, the basic work flow is this:

1. The user selects input data and parameters and starts the job from the Workbench
2. The server exports the input data to a temporary file
3. The server starts the command line application, using the parameters specified from the user and the temporary file as input
4. When the command line application is done, the server imports output data back into the CLC environment, saving it in the data location on the server
5. The user is notified that the job is done, and the results are available for viewing and further analysis in the Workbench

The best way to describe the integration is through a couple of examples that will illustrate the concepts of the integration.

7.1 External applications integration: Velvet

Velvet [Zerbino and Birney, 2008] is a popular de novo assembler for next-generation sequencing data. We have provided example scripts and configurations to set this up as an external application on *CLC Server*.



The user submits a "Command Line" job to the CLC Server using either a custom GUI or the "auto generated GUI to specify data and parameters.

The CLC Server Export Data from the Server Data Persistence to "a flat file" on the server side. The format of the file is according to the setup of the integration and making use of CLC Export Framework. (In the thin client)

The CLC Server starts the execution of the given External Application with the given data (Flat File) and the given parameters.

The result of the Application Execution is either a Flat file (in a given format) OR output in std out

When the External Application terminates the result is imported into the Server Persistence using the CLC Import Framework.

When the result is imported (possibly as a raw text object) some post processing can be done. If for example the result is a list of annotations, these can be applied to the original data (for example sequence)

Figure 7.1: *An overview of the external applications integration.*

The velvet package includes two programs that need to be run consecutively. Because the external application on the *CLC Server* is designed to call one program, a script is needed to encapsulate this.

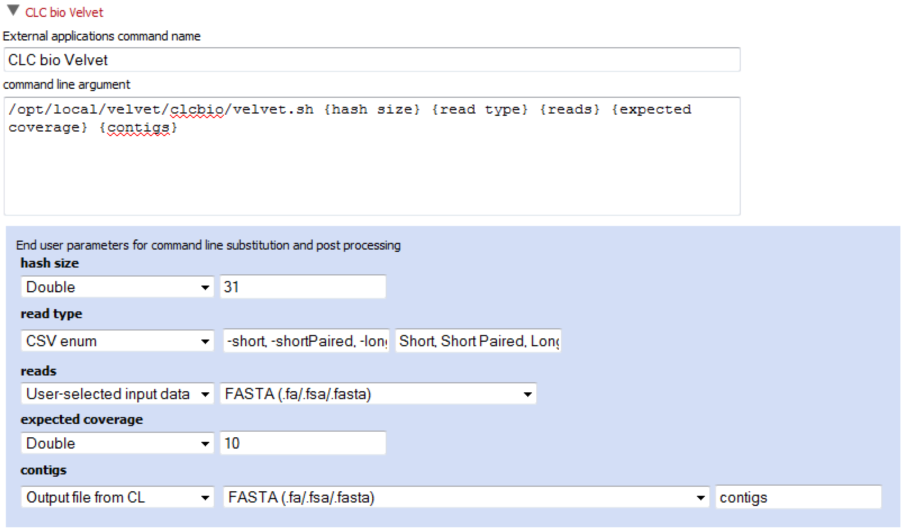
7.1.1 Installing Velvet

To get started, you need to do the following:

- Install Velvet on the server computer (download from `http://www.ebi.ac.uk/~zerbino/velvet/`). Note that if you have job nodes, it needs to be installed on all nodes that will be configured to run Velvet. We assume that Velvet is installed in `/usr/local/velvet` but you can just update the paths if it is placed elsewhere.
- Download the scripts and configuration files made by CLC bio from `http://www.clcbio.com/external-applications/velvet.zip`
- Unzip the file and place the `clcbio` folder and contents in the Velvet installation directory (this is the script tying the two Velvet program together). You need to edit the script if you did not place the Velvet binary files in `/usr/local/velvet`.

- Make sure execute permissions are set on the script and the executable files in the Velvet installation directory. Note that the user executing the files will be the user who started the Server process (if you are using the default start-up script, this will be *root*).
- Use the `velvet.xml` file as a new configuration on the server: Log in to the server via the web interface and go to the **External applications** () tab under **Admin** () and click **Import Configuration**.

When the configuration has been imported, click the **CLC bio Velvet** header and you should see a configuration as shown in figure 7.2.



CLC bio Velvet

External applications command name

CLC bio Velvet

command line argument

```
/opt/local/velvet/clcbio/velvet.sh {hash size} {read type} {reads} {expected
coverage} {contigs}
```

End user parameters for command line substitution and post processing

hash size

Double ▾ 31

read type

CSV enum ▾ -short, -shortPaired, -lon Short, Short Paired, Lon

reads

User-selected input data ▾ FASTA (.fa/.fsa/.fasta) ▾

expected coverage

Double ▾ 10

contigs

Output file from CL ▾ FASTA (.fa/.fsa/.fasta) ▾ contigs

Figure 7.2: *The Velvet configuration has been imported.*

Update the path to the Velvet installation at the very top if necessary.

7.1.2 Running Velvet from the Workbench

Next step is to test if it can actually be executed. Open the Workbench with the **External Applications Client Plug-in** installed. Go to:

Toolbox | CLC Server () | External Applications ()

You will now see a list of all the applications that have been set up (figure 7.3).

In this case there is only one. When you click **Next**, you can select () some sequences, set a few parameters and click **Next** and **Finish**.

The process that follows has three steps:

1. The sequencing reads are exported by the server to a fasta file. The fasta file is a temporary file that will be deleted when the process is done.
2. The velvet script is executed using this fasta file and the user-specified parameters as input.
3. The resulting output file is imported into the save location specified in the save step of the Workbench dialog, and the user is notified that the process is done.

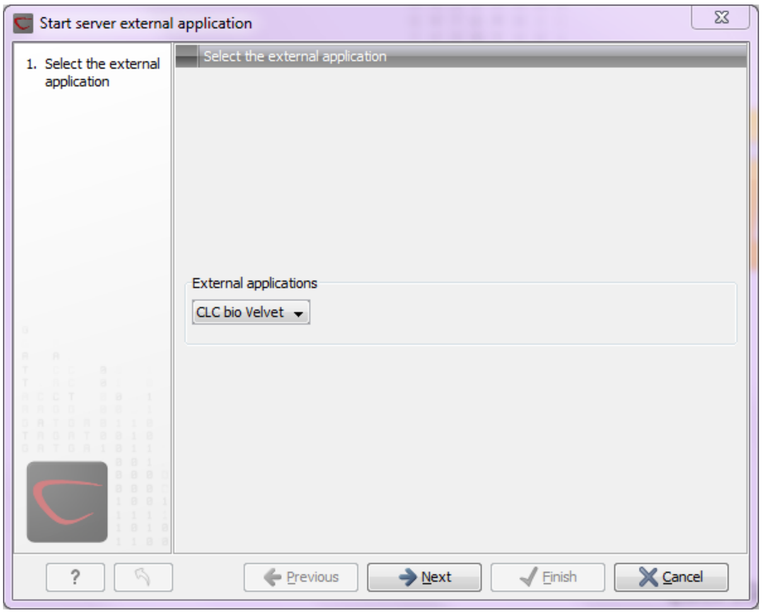


Figure 7.3: *Running Velvet from the Workbench.*

4. All temporary files are deleted

7.1.3 Understanding the Velvet configuration

We will now explain how the configuration that we made actually works. And hopefully this will make it possible for you to design your own integrations.

Going back to figure 7.2, there is a text field at the top. This is where the command expression is created, in this case:

```
/opt/local/velvet/clcbio/velvet.sh {hash size} {read type}
        {reads} {expected coverage} {contigs}
```

The first is the path to the script, and the following are parameters that are interpreted by the server when calling the script because they are surrounded by curly brackets { }. Note that each parameter entered in curly brackets gets an entry in the panel below the command line expression.

The first one, `hash size`, can be entered as a **Double** (which is a number in computer parlance) and it is thus up to the user to provide a value. A default value is entered here in the configuration (31).

The second one is the `read type` which has been configured as a **CSV enum** which is basically a list. The first part consists of the parameters to be used when calling the script (`-short, -shortPaired, -long, -longPaired`), and the second part is the more human-readable representation that is shown in the Workbench (`Short, Short Paired, Long, Long Paired`).

The third parameter is `reads` which is the input data. When the **User-selected input data** option is chosen, a list of all the available export formats is presented. In this case, Velvet expects a fasta file. When a user starts Velvet from the Workbench, the server starts exporting the selected input data to a temporary fasta file before running the script.

The `expected coverage` is similar to hash size.

The last parameter is `contigs` which represents the output file. This time, a list of import data formats is available used to import the data back into the folder that the user selected as save destination.

The rest of the configurations listed below are not used in this example, see the Bowtie example below 7.2.

7.2 External applications integration: Bowtie

Bowtie [Langmead et al., 2009] is a short-reads mapper that can map sequencing reads to a reference sequence. In this example, we show how to make an integration of the mapper where the users selects sequencing reads and sets a few parameters, and then Bowtie is executed using a pre-built index of the reference genome. Because Bowtie needs an indexed reference genome, the selection of a reference sequence as you know from CLC bio's own read mapper is not part of the integration described as an example here.

This means you can either make a list of pre-built index files, or you can use the integration of the indexing tool so that users will be able to work on new organisms that they provide a reference for themselves. These circumstances complicate the explanation a little bit, so for the first part we assume index files are in place.

7.2.1 Installing Bowtie

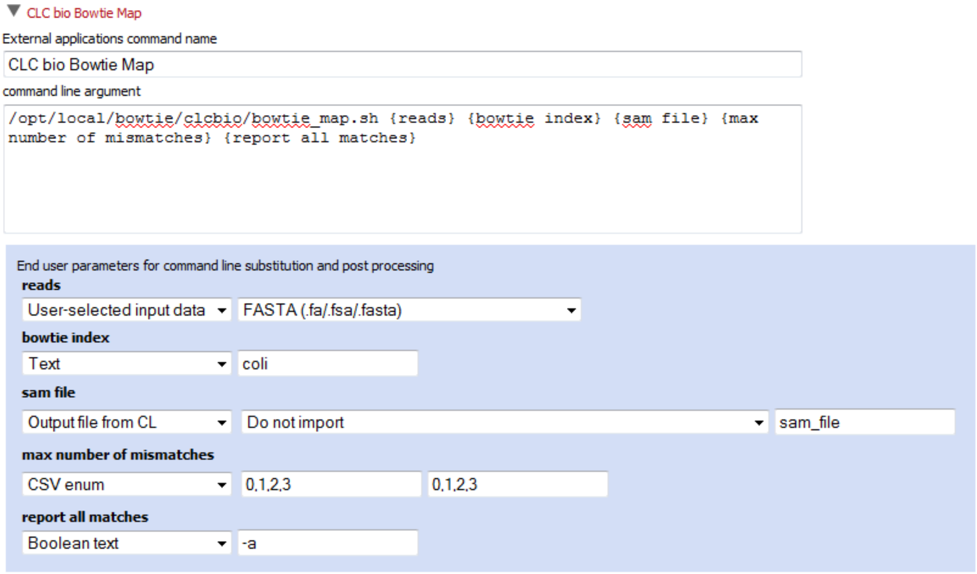
To get started:

- Install Bowtie from `http://bowtie-bio.sourceforge.net/index.shtml`. We assume that Bowtie is installed in `/usr/local/bowtie` but you can just update the paths if it is placed elsewhere.
- Download the scripts and configuration files made by CLC bio from `http://www.clcbio.com/external-applications/bowtie.zip`
- Place the `clcbio` folder and contents in the Velvet installation directory. This is the script used to wrap the Bowtie functionality.
- Make sure execute permissions are set on the scripts and the executable files in the Bowtie installation directory. Note that the user executing the files will be the user who started the Server process (if you are using the default start-up script, this will be *root*).
- Use the `bowtie.xml` file as a new configuration on the server: Log in to the server via the web interface and go to the **External applications** (>_) tab under **Admin** (⚙) and click **Import Configuration**.

From `ftp://ftp.cbcb.umd.edu/pub/data/bowtie_indexes/` you can download pre-built index files of many model organisms. Download the index files relevant for you and extract them into the `indexes` folder in the Bowtie installation directory.

7.2.2 Understanding the Bowtie configuration

Once the `bowtie.xml` has been imported, you can click the **CLC bio Bowtie Map** header to see the configuration as shown figure 7.4.



External applications command name: CLC bio Bowtie Map

command line argument:

```
/opt/local/bowtie/clcbio/bowtie_map.sh {reads} {bowtie index} {sam file} {max
number of mismatches} {report all matches}
```

End user parameters for command line substitution and post processing

reads: User-selected input data ▾ | FASTA (.fa/.fsa/.fasta) ▾

bowtie index: Text ▾ | coli

sam file: Output file from CL ▾ | Do not import ▾ | sam_file

max number of mismatches: CSV enum ▾ | 0,1,2,3 | 0,1,2,3

report all matches: Boolean text ▾ | -a

Figure 7.4: *The Bowtie configuration has been imported.*

The basic configuration is very much similar to the Velvet set-up (section 7.2). The `index` parameter is used to point to the relevant index file and could be substituted by a **CSV enum** to provide a fixed set of index files to be presented to the users.

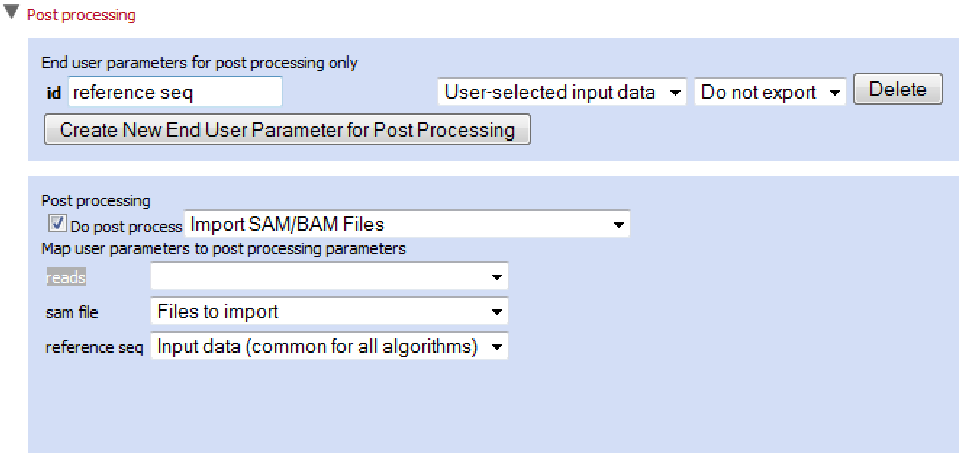
There is one thing that complicates the Bowtie integration when compared to the relatively simple Velvet set-up. The result of Bowtie is a SAM or BAM file which cannot be imported using the standard import framework of *CLC Server*. This is because the file needs to be paired with a reference sequence. But SAM/BAM import is available on the server as an algorithm, similar to Trim, Read mapping and other tools.

In order to make use of this algorithm, the standard export-run-import flow cannot be used. Instead we use the concept of *post-processing* as an alternative to import.

The sam file parameter is set as the **Output file from C**L and the option selected is **Do not import**. If you expand the **Post-processing** panel, you can see the logic needed to handle the SAM file from Bowtie together with the reference sequence provided by the user (see figure 7.5).

At the top, there is a panel for specifying **End user parameters for post processing only** which in this case is the reference sequence. It is not needed by Bowtie which uses the index file as reference, so it is only needed for the SAM/BAM import post-processing.

Below the post-processing algorithm is specified, in this case **Import SAM/BAM Files**, and the input parameters are specified below. To the left, the various parameters (either from the command line expression at the top or from the **End user parameters for post processing only** panel) are listed and you can then map them to the corresponding input parameters for SAM/BAM import. In this case the `sam file` is mapped to **Files to import**, and the `reference seq` is mapped to **Input data**.


Post processing

End user parameters for post processing only

id reference seq — User-selected input data — Do not export — Delete

Create New End User Parameter for Post Processing

Post processing

☑ Do post process: Import SAM/BAM Files

Map user parameters to post processing parameters

reads:

sam file: Files to import

reference seq: Input data (common for all algorithms)

Figure 7.5: *The Bowtie post-processing set-up.*

You may have noticed that the parameters that are available here are not all the ones from the top. This is because the server interprets the parameter types and only lists the ones that are suitable as input to parameters of the SAM/BAM import. In this case it is the output file from the command line program and sequence files selected by the user (this is the reason why `reads` is listed here, because it also represent sequences).

7.2.3 Parameter overview

Since the set-up and flow of parameters can be quite complex, there is a **Parameter flow** panel at the bottom of the configuration with a small graph for each parameter (see figure 7.6).

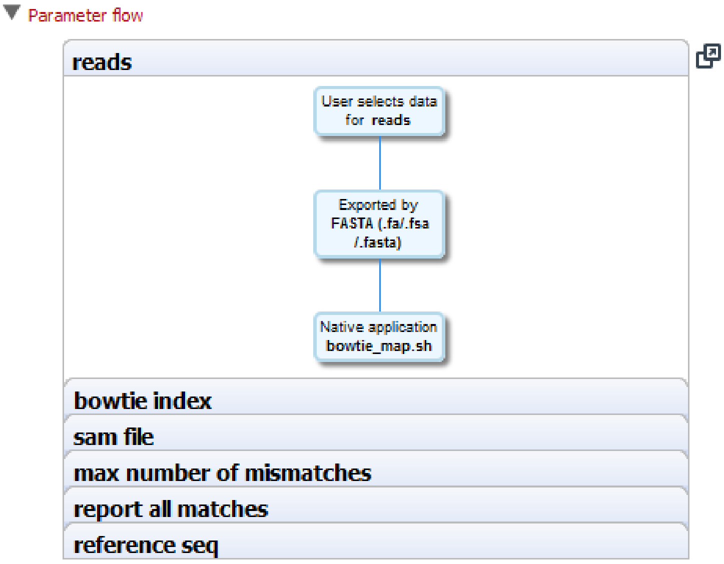


Figure 7.6: *The reads parameter.*

The `reads` parameter starts with user selecting data, and the sequences are exported in fasta format and used as input for the Bowtie script.

Figure 7.7 shows the `max number of mismatches` parameter which also starts with the user selecting a value that is passed to the Bowtie script.

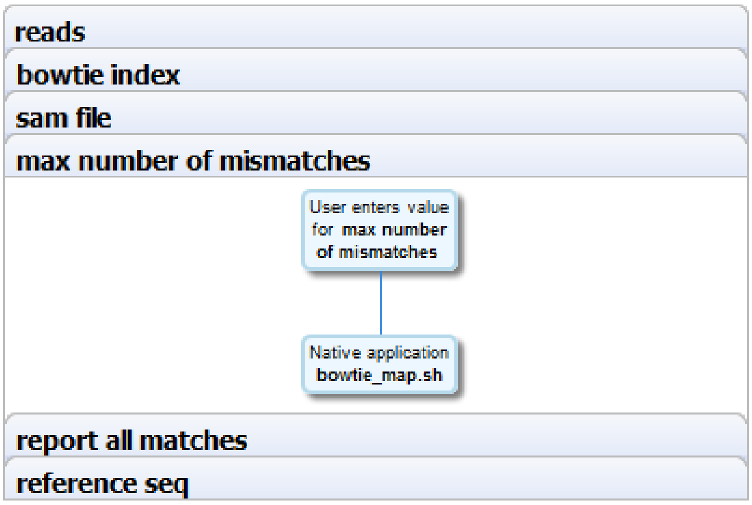


Figure 7.7: *The max number of mismatches parameter flow.*

Figure 7.8 shows the reference seq parameter which also starts with the user selecting data which is passed on to the SAM/BAM import.

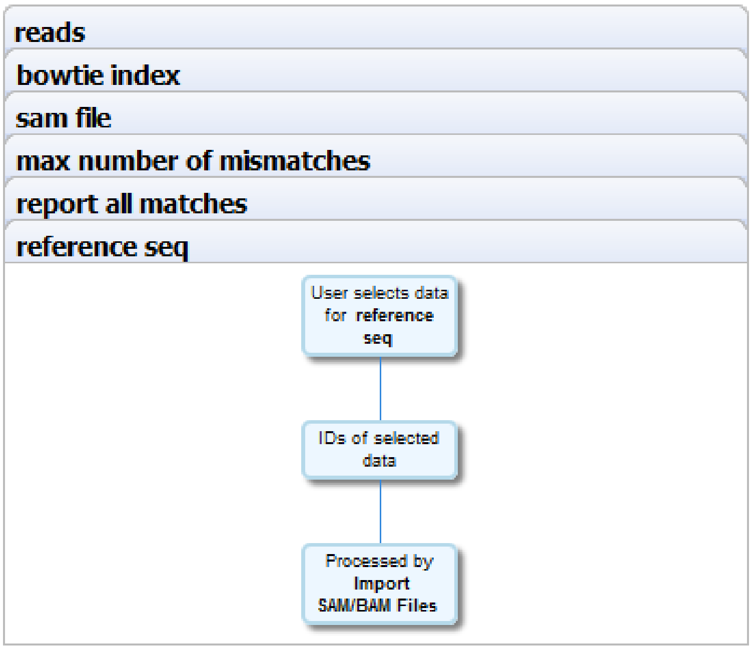
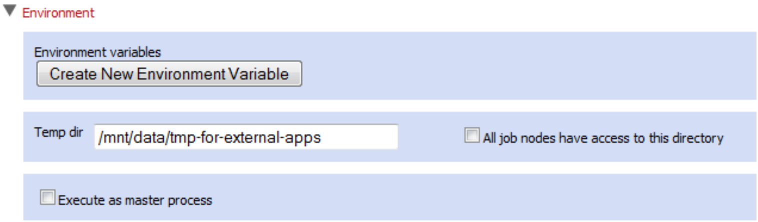


Figure 7.8: *The reference seq parameter flow.*

7.2.4 Setting path for temporary data

The **Environment** handling shown in figure 7.9 allows you to specify a folder for temporary data and add additional environment variables to be set when running the external application.

For Bowtie there is a special case about the temporary data. This is because the post-processing step needs to access the SAM file in a place that is also part of the locations defined for direct import of high-throughput sequencing data (see section **??**). So the path for temporary data should point to a folder that is contained in these locations.



Temp dir /mnt/data/tmp-for-external-apps

☐ All job nodes have access to this directory

☐ Execute as master process

Figure 7.9: *Temporary data should be defined for Bowtie.*

The checkbox **All job nodes have access to this directory** can be checked if the temporary folder is shared between all job nodes. This will eliminate the overhead of transferring the temporary files between job nodes (because the export process could run on a different node than the actual execution of the Bowtie script and the post-processing).

The checkbox to **Execute as master process** can be used if the process does not involve any data processing on the server side (e.g. if a script is called that executes a program on another computer). In this case, the job should not block the queue but act as a master process allowing other jobs to be run at the same time.

7.2.5 Tools for building index files

We have also included scripts and configurations for building index files using the external applications on *CLC Server*. This also includes the possibility of listing the index files available. To get these to work, please make sure the path to the Bowtie installation directory is correct.

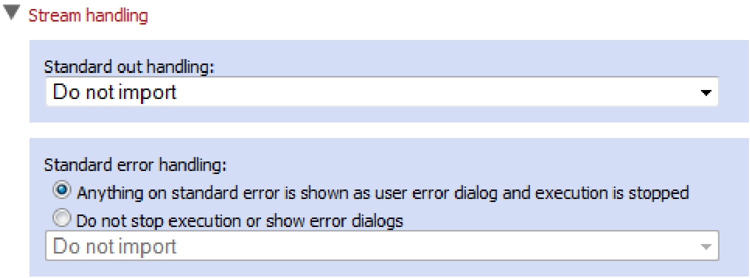
You should also note that the Bowtie distribution includes scripts to download index files of various organisms.

7.3 Stream handling

The examples shown above illustrate the main concepts of the set-up.

However, there is also a general configuration of stream handling available that has not been explained in these examples.

The stream handling shown in figure 7.10 allows you to specify where standard out and standard error for the external application should be handled.



Standard out handling: Do not import

Standard error handling:
● Anything on standard error is shown as user error dialog and execution is stopped
○ Do not stop execution or show error dialogs
Do not import

Figure 7.10: *Stream handling.*

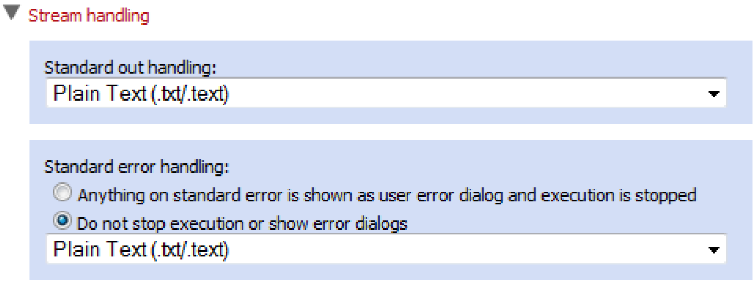
Basically, you can choose to ignore it, or you can import it using one of the importers available on the server. For some applications, standard out produces the main result, so here it makes sense to choose an appropriate importer. But also for debugging purposes it can be beneficial to import standard out and standard error as text so that you can see it in the Workbench after a run.

7.4 Checking the configuration

Since there is no check of consistency of the configuration when it has been set up, errors will only be seen on runtime when the application is executed. In order to help trouble-shooting in case of problems, there are a few things that can be done:

First, in the error dialog that will be presented in the workbench, you can see the actual command line call in the **Advanced** tab at the bottom. This can be a great help identifying syntax errors in the call.

Second, if you choose to import standard out and standard error as text, this will make it possible to check error messages posted by the external application (see figure 7.11).



Standard out handling:
Plain Text (.txt/.text)

Standard error handling:
○ Anything on standard error is shown as user error dialog and execution is stopped
● Do not stop execution or show error dialogs
Plain Text (.txt/.text)

Figure 7.11: *Importing the result of standard error and standard out.*

Once the set-up is running and stable, you can deselect these options.

Chapter 8

Troubleshooting

If there are problems regarding the installation and configuration of the server, please contact support@clcbio.com.

8.1 Check set-up

In order to check that your server has been set up correctly, you can run the **Check set-up** tool. Log in on the web interface of the server as an administrator and click the **Check Set-up** link at the upper right corner. This will show a dialog where you click **Generate Diagnostics Report**.

This will show a list of test that are performed on the system as shown in figure 8.1.

If any of the tests fail, it will be shown in the list. You can expand each of the tests to display more information about what the test is checking and information about the error if it fails.

8.2 Bug reporting

When contacting support@clcbio.com regarding problems on the server, you will often be asked for additional information about the server set-up etc. In this case, you can easily send the necessary information by submitting a bug report:

> **Log in to the web interface of the server as administrator | report a bug (at the top right corner) | Enter relevant information with as much detail as possible | Submit Bug Report to CLC bio**

You can see the bug report dialog in 8.2.

The bug report includes the following information:

- Log files
- A subset of the audit log showing the last events that happened on the server
- Configuration files of the server configuration

In a job node set-up you can include all this information from the job nodes as well by checking the **Include comprehensive job node info** checkbox in the **Advanced** part of the dialog.

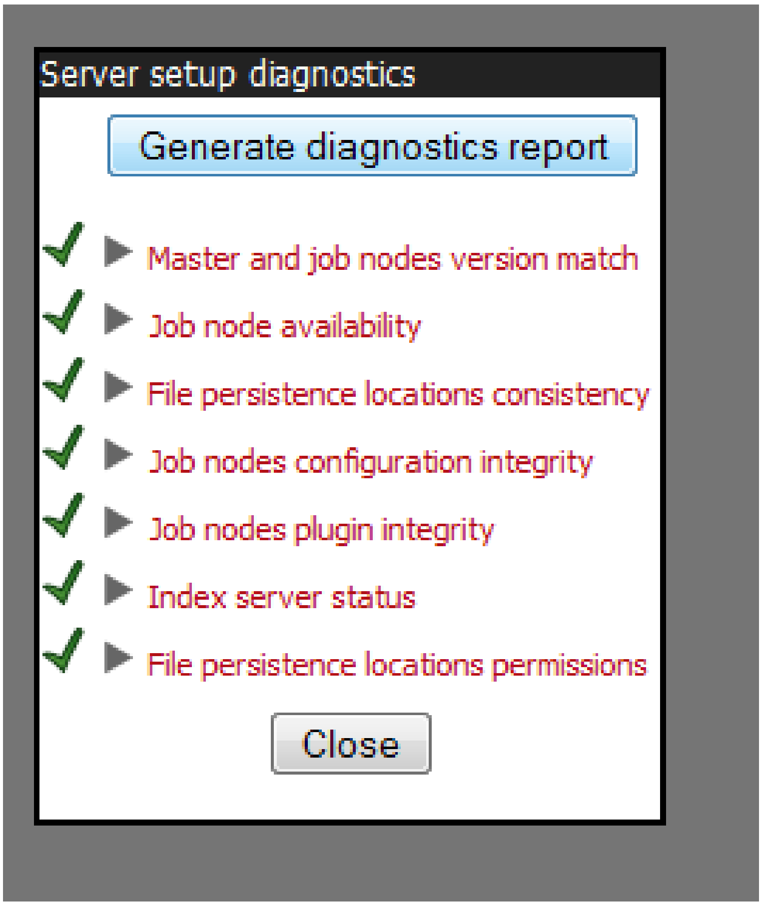
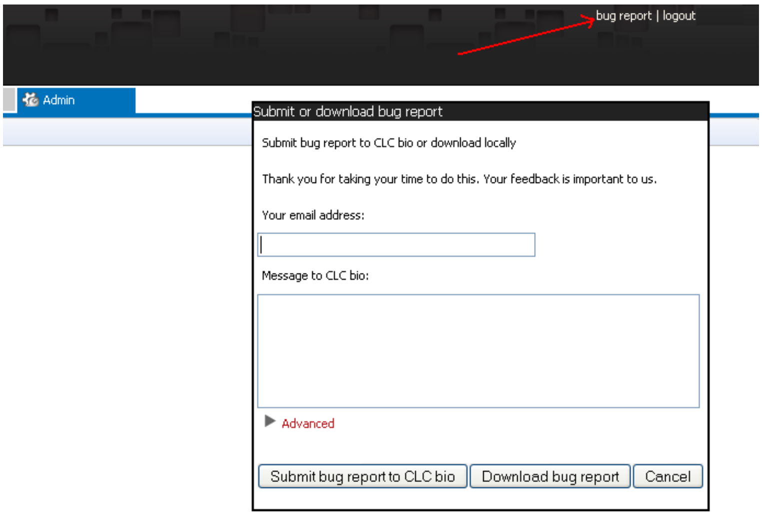


Figure 8.1: *Check system.*



Submit bug report to CLC bio or download locally

Thank you for taking your time to do this. Your feedback is important to us.

Your email address:

Message to CLC bio:

Figure 8.2: *Submitting a bug report to CLC bio.*

If the server does not have access to the internet, you can **Download bug report**. This will create a zip file containing all the information and you can pass that on to CLC bio support. If the server has access to the internet, you can **Submit Bug Report to CLC bio**.

Note that the process of gathering the information for the bug report can take a while, especially for job node set-ups. If a Workbench user experiences a server-related error, it is also possible to submit a bug report from the Workbench error dialog. This report will include the same archive as when submitting a bug report from the web interface. All data sent to support@clcbio.com is treated confidentially.

No password information is included in the bug report.

Bibliography

[Langmead et al., 2009] Langmead, B., Trapnell, C., Pop, M., and Salzberg, S. L. (2009). Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome Biol*, 10(3):R25.

[Zerbino and Birney, 2008] Zerbino, D. R. and Birney, E. (2008). Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res*, 18(5):821–829.

Index