Deployment Manual

CLC WORKBENCHES

Manual for

CLC Workbenches: deployment and technical information, version 1.6

Windows, Mac OS X and Linux

June 19, 2015

This software is for research purposes only.

QIAGEN Aarhus A/S
Silkeborgvej 2
Prismet
DK-8000 Aarhus C
Denmark

Contents

1 Introduction 5

1.1 Deployment strategies 5

1.2 System requirements 6

2 Installation of Workbenches 9

2.1 Available installers 9

2.1.1 Java 9

2.2 What does a Workbench installer do when it is run? 9

2.2.1 Extracting and copying files to the installation directory 9

2.2.2 Setting the amount of memory available 10

2.2.3 Shortcuts and file associations 10

2.3 Silent installation 10

3 License 12

3.1 License server set-up on clients 12

4 Plugins and resources 14

5 Workflows 15

6 Connecting to a CLC Server 16

7 Security policies 17

8 Storing and backing up data 19

8.1 Storing data 19

8.1.1 Data structure 19

8.1.2 Changing the default location 20

8.2 Back-up of data 21
8.3 Special configurations for large amounts of data 21
8.3.1 Temporary data 21
8.3.2 Disk space requirements 22

9 System resources 23
9.1 Setting the amount of memory available for the JVM 23
9.2 Setting the number of cores to use 24

10 Overview - where do we put things? 26
10.1 Computer-level information 26
10.2 Property files overview 26
10.3 User-level information 27

Index 28

Chapter 1

Introduction

The information in this manual is aimed at administrators who are responsible for Workbench installation, configuration and maintenance at a site. We include information here about installers, licenses and where the files associated with a Workbench installation are put.

For installing the *CLC Bioinformatics Database* and *CLC Server*, please refer to the installation chapter in the respective user manuals (see `http://www.clcbio.com/usermanuals`).

At the moment, the CLC Workbenches are:

- *CLC Genomics Workbench*
- *Biomedical Genomics Workbench*
- *CLC Drug Discovery Workbench*
- *CLC Main Workbench*

In addition, there is the *CLC Sequence Viewer*, which is freely available.

In the first part of this manual, we take a closer look at deployment. Then the license system is discussed, followed by an explanation of the concept of plugins and workflows. Finally there is an overview chapter which tells you where to find all the different files in a client installation. This is useful when deploying the Workbench and also for back-up purposes.

Please contact support@clcbio.com if your questions about deployment are not answered in this manual.

1.1 Deployment strategies

Deployment strategies should be developed so that they fit your existing IT set-up.

The Workbench is very self-contained and simple to configure. We provide here a brief summary of steps that can be taken to cut down on the work required to deploy Workbenches to a number of machines at a site. These steps can be tailored to suit the requirements on your own site. The steps below still require you to log into each machine to install the Workbench, but cuts down on time spent configuring each machine separately withi things like plugins, memory settings, default data locations and temporary file locations. If you are working with a network license, this also saves time configuring these details on each machine. The rest of this manual is devoted to providing more details about Workbench installation, licensing and configuration.

1. On a single computer with the same OS as the target computers, complete a full installation of the Workbench including:
   - Running the installer (read more about what installers in section 2)
   - Set up the license server connection if you are using a network license setup. If you are using static licenses, you will need to download a license for this machine to complete the work described here and then activate licenses on each computer individually after the Workbench is installed on them.
   - Install the desired plugins. If you have any commercial plugins, set up the license server connection for these or license this particular machine copy if you are using static licenses. See the individual plugin manuals for instructions on how to do this.
   - Configure security policies, path where temporary data should be written to and the default data locations to be used to save data.
   - Specify proxy server information if needed for establishing connections to the external network.
2. Run the installer in silent mode (`-q`) on all the target computers.
3. Copy the following files from the installation in 1) to the target computers.
   - All files from the `settings` folder in the installation directory
   - `plugins` and `resource` folder from the installation directory

Creating a GHOST image or similar to copy to all the target computers is also a possibility, but there are a few routines performed by the installer that need to be taken into account, especially allocating memory (see section 9.1) and the creation of shortcuts and file associations.

1.2 System requirements

The system requirements of the CLC workbenches except the *CLC Genomics Workbench* are these:

- Windows Vista, Windows 7, Windows 8, or Windows Server 2008.
- Mac OS X 10.7 or later.
- Linux: RHEL 5.0 or later. SUSE 10.2 or later. Fedora 6 or later.
- 32 or 64 bit.
- 1 GB RAM required.
- 2 GB RAM recommended.
- 1024 x 768 display required.
- 1600 x 1200 display recommended. .

The requirements for the *CLC Genomics Workbench* are:

- Windows Vista, Windows 7, Windows 8 or Windows Server 2008
- Mac OS X 10.7 or later.
- Linux: RHEL 5.0 or later. SUSE 10.2 or later. Fedora 6 or later.
- 2 GB RAM required
- 4 GB RAM recommended
- 1024 x 768 display required
- 1600 x 1200 display recommended
- Intel or AMD CPU required
- **Special requirements for the 3D Molecule Viewer**
  - **System requirements**
  - **3D Graphics Requirements**
    - A graphics card capable of supporting OpenGL 2.0. Note that the CLC workbenches except the *CLC Genomics Workbench* only uses the GPU for the OpenGL 3D rendering. The GPU is not used to speed up molecular simulations.
    - Updated graphics drivers. Please make sure the latest driver for the graphics card is installed .
  - **System Recommendations**
  - **3D Graphics Recommendations**
    - A discrete graphics card from either Nvidia or AMD/ATI. Modern integrated graphics cards (such as the Intel HD Graphics series) may also be used, but these are usually slower than the discrete cards.
    - A 64-bit workbench version is recommended for working with large complexes.
- **Special requirements for read mapping**. The numbers below give minimum and recommended memory for systems running mapping and analysis tasks. The requirements suggested are based on the genome size. Systems with less memory than specified below will benefit from installing the legacy read mapper plugin (see http://www.clcbio.com/plugins). This is slower than the standard mapper but adjusts to the amount of memory available.
  - **E. coli K12 (4.6 megabases)**
    - Minimum: 2 GB RAM
    - Recommended: 4 GB RAM
  - **C. elegans (100 megabases)** and **Arabidopsis thaliana (120 megabases)**
    - Minimum: 2 GB RAM
    - Recommended: 4 GB RAM
  - **Zebrafish (1.5 gigabases)**
    - Minimum: 2 GB RAM

    - Recommended: 4 GB RAM
  - **Human (3.2 gigabases)** and **Mouse (2.7 gigabases)**
    - Minimum: 6 GB RAM
    - Recommended: 8 GB RAM

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

Chapter 2

Installation of Workbenches

This chapter describes the Workbench installation process when using the standard installers.

2.1 Available installers

There are installers available for each platform (Windows, Mac OS X and Linux). Separate 32-bit and a 64-bit installers are available for Linux and windows systems. For Linux, there is both a `.sh` installer and an `.rpm` package.

2.1.1 Java

The Workbenches are Java software, which means there must be a Java Runtime Environment (JRE) to run the Workbench. The Workbench installers have a built-in JRE that is installed under the installation directory of the Workbench. The advantages of this include:

1. For computers that do not already have a JRE installed, no JRE needs to be downloaded and installed from another source.
2. For computers that already have a JRE installed, compatibility problems are avoided because the Workbench uses its own JRE.

The JRE used for running the CLC Workbench do not interfere with existing JREs on the computer.

2.2 What does a Workbench installer do when it is run?

The installer performs the following tasks:

2.2.1 Extracting and copying files to the installation directory

The Workbench is installed into the following directory by default (we use *CLC Main Workbench 6* as example):

Windows C:\Program files\Main Workbench 7

Mac OS X Applications/Main Workbench 7

Note that each major version of a Workbench has its own installation directory. This means that when upgrading between major versions, e.g. *CLC Main Workbench 6* to *CLC Main Workbench 7*, the old installation directory of version 6 will be left untouched.

If you wish to remove the old installation, please run the Uninstall tool included with the Workbench.

Minor updates use the existing installation directory of the Workbench. For example, when upgrading from *CLC Main Workbench 7* to *CLC Main Workbench 7.01* or *7.1*, the usual route would be to upgrade in place. This would thus use the existing installation area for that major version of the software.

If you do not wish to use the default installation area, or you are not sure where this will end up being, you can choose to define the installation directory during installation by selecting the option to install to a different installation area that is presented.

2.2.2 Setting the amount of memory available

The installer investigates the amount of RAM on the machine during installation and sets the amount of memory that the Workbench can use. Read more in section 9.1.

2.2.3 Shortcuts and file associations

The installer also creates shortcuts for starting the Workbench in the case of Windows and Mac systems, and the option to create file associations so that `.clc` files will be opened by the Workbench is presented near the end of the installation procedure.

2.3 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.

```
CLCMainWorkbench_7_5.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):

```
CLCMainWorkbench_7_5.exe -q -console
```

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

```
CLCMainWorkbench_7_5.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.

Linux and Mac systems are also supported.

On Mac, this looks something like:

```
/Volumes/CLCGenomicsWorkb/CLC Genomics Workbench Installer.app
/Contents/MacOS/JavaApplicationStub -q
```

On Linux the following options are supported:

```
CLCMainWorkbench_7_5.exe -c
```

This forces the installer to run in console mode.

To do a fully unattended installation use the following options:

```
CLCMainWorkbench_7_5.exe -c -q -dir /opt/clcgenomicsserver
```

This installs the product in /opt/clcgenomicsserver.

Chapter 3

License

There are fundamentally two kinds of licenses for the Workbenches:

Fixed license A license order ID has to be activated against our server for each computer. The license will then be fixed to this computer. This requires manual intervention for each activation.

Floating license A license server is installed in your organization. It hosts a number of licenses which can be shared among all computers. Note that the license server is available for both Linux, Windows and Mac OS X.

For large installations, the floating license is by far the best option, since all the license administration takes place on the server (find the manual for the license server together with the server distribution). The fixed license requires manual work during installation and also if the licenses need to be updated.

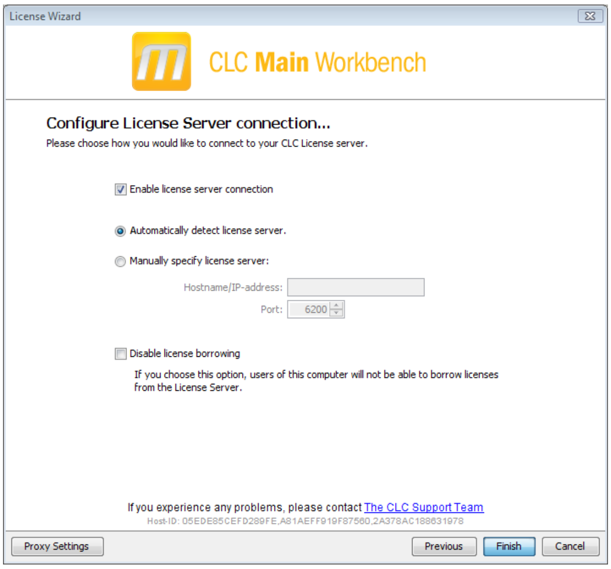
For information on how to use the floating license, please refer to the user manual for the relevant Workbench (see http://www.clcbio.com/manuals).

Plugins use the same licensing system as the Workbenches, so all the concepts described here also apply to the plugin licenses.

3.1 License server set-up on clients

The connection to the license server can be set up as described in the Workbench user manual (see also figure 3.1).

The license server information is stored in a file called license.properties in the settings folder in the Workbench installation directory. This means, that you need write access to the installation directory (with the default installation directory, you need to be an administrator to have this write access) in order to set up a connection to the license server. The file contains the following:



Configure License Server connection...

Please choose how you would like to connect to your CLC License server.

☑ Enable license server connection

● Automatically detect license server.

○ Manually specify license server:

Hostname/IP-address:

Port: 6200

□ Disable license borrowing

If you choose this option, users of this computer will not be able to borrow licenses from the License Server.

If you experience any problems, please contact The CLC Support Team

Host-ID: 05EDE85CEFD289FE,A81AEFF919F87560,2A378AC188631978

Figure 3.1: *Connecting to a license server.*

```
serverip=
serverport=6200
disableborrow=false
autodiscover=true
useserver=true
```

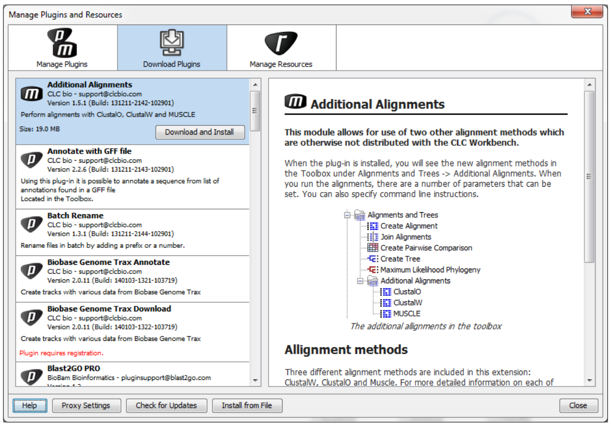
Since all this information is stored in a file in the installation directory, it can easily be copied to all clients, and the license configuration is completed. When the Workbench is started, it will look in this file, and if `useserver=true` then it will try to connect to the license server, and no license dialogs will be shown to the user.

You can download a sample `license.properties` file at http://clcbio.com/files/deployment/license.properties.

Chapter 4

Plugins and resources

There is a graphical user interface to install plugins called the **Plugin Manager** ( ) which is invoked in the **Help** menu (see figure 4.1).



Additional Alignments
CLC bio - support@clcbio.com
Version 1.5.1 (Build: 131211-2142-102901)
Perform alignments with ClustalO, ClustalW and MUSCLE
Size: 19.0 MB

Annotate with GFF file
CLC bio - support@clcbio.com
Version 2.2.6 (Build: 131211-2143-102901)
Using this plug-in it is possible to annotate a sequence from list of annotations found in a GFF file
Located in the Toolbox.

Batch Rename
CLC bio - support@clcbio.com
Version 1.3.1 (Build: 131211-2144-102901)
Rename files in batch by adding a prefix or a number.

Biobase Genome Trax Annotate
CLC bio - support@clcbio.com
Version 2.0.11 (Build: 140103-1321-103719)
Create tracks with various data from Biobase Genome Trax

Biobase Genome Trax Download
CLC bio - support@clcbio.com
Version 2.0.11 (Build: 140103-1322-103719)
Create tracks with various data from Biobase Genome Trax
Plugin requires registration.

Blast2GO PRO
BioBam Bioinformatics - pluginsupport@blast2go.com

Additional Alignments

This module allows for use of two other alignment methods which are otherwise not distributed with the CLC Workbench.

When the plug-in is installed, you will see the new alignment methods in the Toolbox under Alignments and Trees -> Additional Alignments. When you run the alignments, there are a number of parameters that can be set. You can also specify command line instructions.

The additional allignments in the toolbox

Allignment methods

Three different alignment methods are included in this extension: ClustalW, ClustalO and Muscle. For more detailed information on each of

Figure 4.1: *The Plugin Manager.*

Plugins are either general modules or extensions provided by CLC bio (see http://www.clcbio.com/plugins) or can be custom-made plugins specific to your organization.

Plugins can either be downloaded and installed directly in the **Plugin Manager**, or they can be installed from a file Install from File button at the bottom of the **Plugin Manager**.

Resources are installed in the same way as plugins. Resources can be e.g. PFAM databases used by the Workbench's PFAM Domain Search ( ).

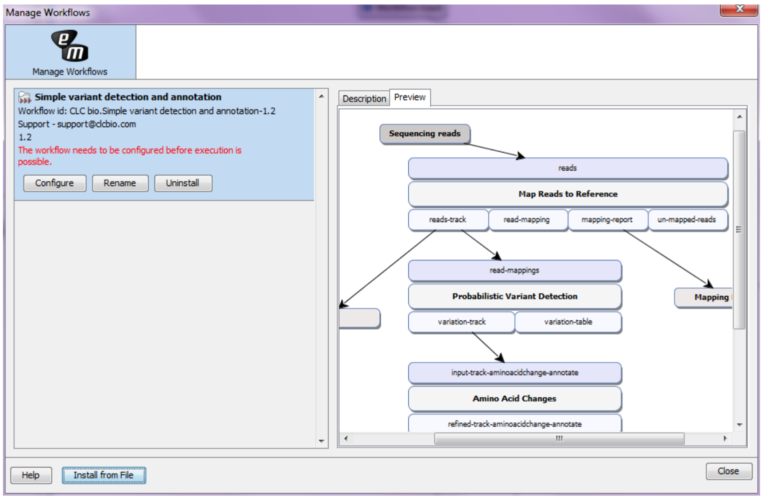
Installing a plugin is basically just a matter of putting files in the right folder. All plugin files are put in `plugins` and all resources in `resource` in the installation directory. This means that the contents of these folders can be copied to other computers, and they will have the plugins installed.

Licenses for the plugins are handled the same way as the Workbench licenses, see section 3.

Chapter 5

Workflows

There is a graphical user interface to install workflows called **Workflows** () which is invoked in the **Help** menu (see figure 5.1).



Simple variant detection and annotation
Workflow id: CLC bio.Simple variant detection and annotation-1.2
Support - support@clcbio.com
1.2
The workflow needs to be configured before execution is possible.

Figure 5.1: *The workflow manager.*

Workflows can be created in any workbench and distributed as an installer file that can be installed in any workbench or server.

A workflow is always installed per user. The workflow definition is stored in the user home (see section 10.3).

The ability to install workflows can be disabled by the policy (see section 7).

When the workbench is part of a *CLC Genomics Server* set-up, it will be a great advantage to manage workflows to be used by all users on the server which means no local deployment when workflows are updated (see more in the user manual at http://www.clcbio.com/usermanuals).

Chapter 6

Connecting to a CLC Server

Information about server name and port can be stored in a file called serverinfo.properties in the settings folder in the Workbench installation directory. When the user opens the log-in dialog, the Workbench will read in the information from this file.

If the file does not exist, the information that the user enters will be saved in the user settings. The user name and password is stored with the user settings.

The serverinfo.properties file contains the following:

```
port=7777
host=hostname
```

Since all this information is stored in a file in the installation directory, it can easily be copied to all clients.

You can download a sample serverinfo.properties file at http://clcbio.com/files/deployment/serverinfo.properties.

Chapter 7

Security policies

The Workbench has a security policy configuration that enables administrators to restrict users' access to:

- **Tools accessing services on the internet.** This includes NCBI BLAST, NCBI and Uniprot Searches.
- **Notifications about updates.** Update notifications on new Workbench and plugin versions
- **Plugin management.** Installation of plugins.

The configuration is specified in a simple properties file called `policy.properties` that resides in the `settings` folder of the installation directory (e.g. `C:\Program Files\CLC Main Workbench 6\settings` on Windows). Note that users without administrator access will not be able to change the contents of this file. Each of the following keys can be followed be either *allow* or *deny*:

workbench_version_check Controls whether notifications for Workbench updates should be shown.

plugin_version_check Controls whether notifications for plugin updates should be shown. Note that if `plugin_download` is not allowed, plugin update notifications will not be shown, regardless of this setting.

online_search Controls whether the **Download Genome ()**, **Search for Sequences at NCBI ()**, **Search for PDB Structures at NCBI ()** and **Search for Sequences in UniProt ()** in the **Download** menu should be available (these can be used to search for and download sequences, reference data resources and 3D structures from NCBI and Swissprot/Uniprot).

online_ncbi_blast Controls whether all the tools performing BLAST at NCBI's servers should be available. This is **NCBI BLAST ()** both from the **Toolbox** and from sequence selections.

online_ncbi_download Controls whether the tool performing download of BLAST databases from NCBI's servers should be available. This is **Download BLAST databases ()** tool from the **Toolbox**.

plugin_manage Controls whether the **Plugins and Resources** manager should be available for the user. Note that users can still install plugin updates if `plugin_download` and `plugin_version_check` are allowed.

plugin_file_install Controls whether this user should be allowed to install plugins and resources from a local file.

plugin_download Controls whether this user should be allowed to install CLC plugins and resources downloaded directly within the plugin manager dialog. This also includes manually checking for updated plugins in the **Plugins and Resources** manager dialog and also the automatic check for plugin updates at start-up.

workflow_manage Controls whether the user should be allowed to manage workflows.

workflow_file_install Controls whether the user should be allowed to install workflows from a file.

workflow_download Controls whether the user should be allowed to download and install workflows form the CLC workflows repository.

sequence_to_structure It is possible to remove the 'Link Variants To Structure' action by adding a line called: *sequence_to_structure = deny* to the policy.properties. This will disable the action, and prevent users from clicking 'Link to Structure' links in variant tables. It is also possible to remove the 'Download Find Structure Database' (DDWB) / 'Download 3D Protein Structure Database' (GWB + CRWB) action by adding the line called: *sequence_to_structure = deny* to the policy.properties.

Per default, there is no policy.properties file, so everything is allowed. A commented sample file that you can download and edit is located at `http://clcbio.com/files/deployment/2/policy.properties`. Download the file and place it in the `settings` folder, update the relevant values, and the new policy will take effect next time the Workbench is started.

Chapter 8

Storing and backing up data

This chapter explains how data is stored, gives general guidance on size of data, and outlines configurations needed for running analyses on large amounts of data.

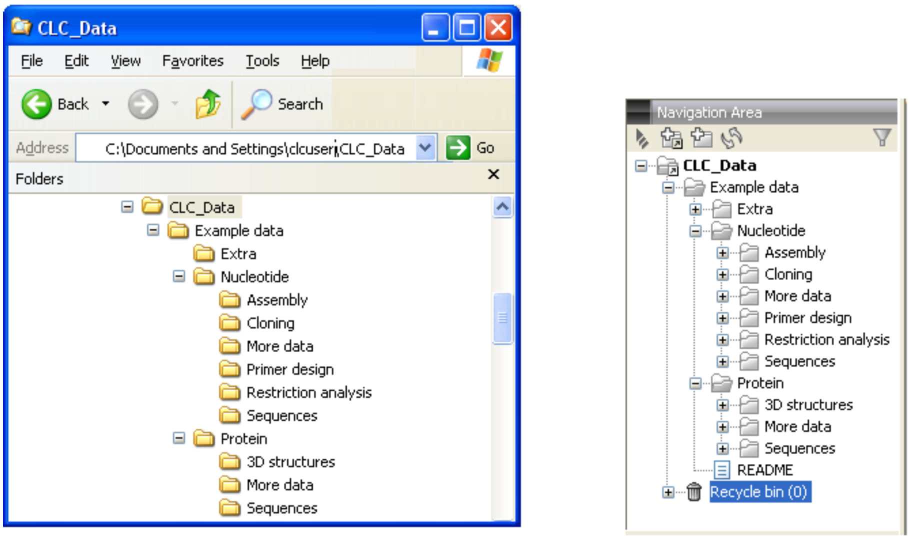
8.1 Storing data

8.1.1 Data structure

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

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

This is explained visually in figure 8.1. The full path to the system folder can be located by mousing over the data location as shown in figure 8.2.



Address C:\Documents and Settings\clcuser\CLC_Data

Figure 8.1: *In this example the location called 'CLC_Data' points to the folder at C:\Documents and settings\clcuser\CLC_Data.*

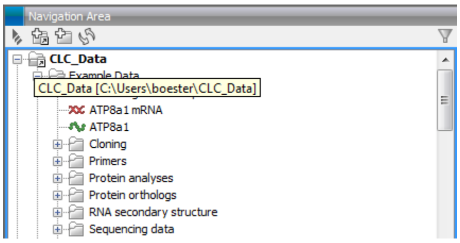


Figure 8.2: *Mousing over the location called 'CLC_Data' shows the full path to the system folder, which in this case is C:\Users\boester\CLC_Data.*

If the Workbench is connected to a CLC Server, the server's locations will automatically show up when the user is logged in. This chapter does not deal with server locations - please refer to the server user manual at `http://www.clcbio.com/usermanuals`.

The list of locations is stored in a file called `model_settings_300.xml` in the `settings` folder in the user home (see section 10.3). We do not recommend manual editing of this file, although it is standard xml.

8.1.2 Changing the default location

In some set-ups, storing data in the default location `CLC_Data` in the user home is not desired. This could be for roaming user profiles or in situations where there is a quota on this disk.

The default location that is used the first time the Workbench starts can be configured in the `path.properties` file that resides in the `settings` folder of the installation directory (e.g. `C:\Program Files\CLC Main Workbench 5\settings` on Windows). Add a line like this to the file to change the default location: The file should include one line like this:

```
datadir = c:\clcdata
```

The following variables can be used to construct the path to the desired location:

$user the user name of the current user

$home the home directory of the current user

$product the short name of the workbench (example: clcgenomicswb or clcmainwb)

A few examples:

```
datadir = $home/CLC_Data        (default)
datadir = X:\clcstorage\$user   (seperate disk / network mount in Windows)
```

Note that the folder does not need to exist - it will be created if needed. You can use both slash (normally used on Mac and Linux) and backslash (normally used on Windows) in the configuration file - the Workbench will use the appropriate one depending on the platform.

Note that the default location is only considered the very first time the Workbench starts. When the Workbench closes the first time, the `model_settings_300.xml` file is created and this is where it will look for the locations further on. Deleting this file will make the Workbench look for the datadir property in the path.properties file. The model_settings_300.xml file is located in the settings folder in the user home (see section 10.3).

You can download a sample path.properties file at http://clcbio.com/files/deployment/path.properties.

8.2 Back-up of data

Since all data used in the Workbench is stored as files in the locations specified, a back-up procedure has to include all the locations. If the data needs to be restored from a back-up, simply copy the files back into the folder locations and start the Workbench.

Database locations needs a different back-up procedure.

Besides the data itself, user-level settings should also be included in the back-up (see section 10.3).

8.3 Special configurations for large amounts of data

Especially the *CLC Genomics Workbench* is often used with large amounts of data. This means that special configurations often need to be made. This concerns locations for temporary data and disk space in general.

8.3.1 Temporary data

The Workbench has a built-in cache system that intends to make sure that the Workbench does not run out of memory even for large data sets. During various processes such as assembly and RNA-Seq analysis, the Workbench often writes temporary files to the disk. Depending on the data set, these temporary files can take up a lot of disk space.

If there is not enough space in the default tmp directory, the tmp directory can be re-directed:

Create a text file called path.properties and save it in the settings folder in the Workbench installation directory. Please ensure this file does not have a ".txt" extension. The file should include one line like this:

```
tmpdir = /path-to-temp
```

Instead of "/path-to-temp" you write the absolute path to the new tmp directory. When the Workbench is restarted, it will then use the new directory for storing temporary data.

You can download a sample path.properties file at http://clcbio.com/files/deployment/path.properties.

Note! It is imperative for acceptable performance that data transfer to the temp directory is not over a network connection. Since the Workbench will spend a lot of time writing and reading these files, disk speed has a great impact on overall performance when working with large data sets.

8.3.2 Disk space requirements

It is hard to give general guidance on disk space requirements, but we have made an example of a typical work flow for *CLC Genomics Workbench* to illustrate.

For calculating disk space for next-generation sequencing data you need to consider the following:

- Reads are imported and take up space as raw reads (see details below). Once imported, you can delete the original sequence file if you do not need it for other purposes.
- When the data has been assembled, either de-novo or against a reference, they take up space once again (this time more space since there is also information about where they map etc).
- Reference sequences also take up space.
- The computer doing the analysis needs space for tmp files. Once the assembly is done, the temporary files are deleted. The temporary files usually do not take up more space than the final result

The formulae giving the disk space usage:

Bytes per read: 28 + (length of read name) + 0.25 x (length of read)

Note that you can discard read names during import.

If quality scores are present, add: 6 + (length of read)

If color space encoding is present, add: 7

As an example, a data set of 5.2 million 35 bp reads imported by *CLC Genomics Workbench* using the **Discard sequence names** option including quality scores takes up:

5,244,764 x ((28 + 0 + 0.25 x 35) + (6 + 35)) = 389 MB

When mapped to a 4.7 Mbp annotated reference sequence, the mapping results takes up 473 MB.

Chapter 9

System resources

You can specify the number of CPU and the amount of memory that the java process of the Workbench is allowed to use.

9.1 Setting the amount of memory available for the JVM

When running the Workbench, the Java Virtual Machine (JVM) needs to know how much memory it can use. This depends on the amount of physical memory (RAM) and can thus be different from computer to computer. Therefore, the installer investigates the amount of RAM during installation and sets the amount of memory that the Workbench JVM can use to 50% of the computer's RAM or 50GB, whichever is the smaller. We recommend using these defaults.

For 32 bit systems, an upper limit exists, which is between 1000MB and 2000MB, depending on the system. On Windows, this will likely be 1200MB.

The reason behind these defaults is explained below. First, we explain how to change these values, should you wish to.

Changing the memory setting for the Workbench java process You can raise or lower the amount of heap space that the Workbench java process is allowed to use by editing a setting in a file, as described below.

On **Windows** and **Linux**, this value is stored in a property file called `workbenchname.vmoptions` (e.g. `clcmainwb.vmoptions`). It is the Xmx setting. The content of this file looks like this:

```
-Xmx8000m
```

The value provided (8000m) is the amount of memory the Workbench java process is allowed to use. Here, 8000MB. The vmoptions file is placed directly under the installation folder (e.g. on Windows, this would be at a location like

`C:\Program Files\CLC Main Workbench 7\clcmainwb.vmoptions`).

On **Mac OS X**, the -Xmx value used to be stored in `Info.plist` in the application bundle (Control-click the application and choose "Show Package Contents"). In more recent releases it is stored at the same location in the package path but in a file called `vmoptions.txt`.

The `Info.plist` file can still be used for customized -Xmx settings but the following needs to happen first:

1. add the following line into the `Info.plist` file, under the 'VMOptionArray' section: `-Xmx4100m` (you can adjust "4100" to the memory setting you want)
2. the -Xmx setting inside the `vmoptions.txt` file needs to be removed, otherwise it will overwrite the setting in the Info.plist.'

If you do not wish to use the installer on each computer and plan to use an image instead, either all computers should have the same amount of RAM, or set the value used in the settings file to 50% of the computer with the smallest amount of RAM.

es need to be created if there is not one memory setting that will fit all Workbench computers.

Default memory setting recommendations explained The suggested upper limit of 50GB is because there are limited gains in most circumstances if the value is increased beyond this java process beyond this and it ensures that the JVM does not spend too much time running garbage collection (GC) processes. The GC is a memory handling subsystem of the JVM, which scans through the currently used heap, and frees up memory by removing contents no longer in use. The larger the maximum heap, which is what this memory setting is affecting, the longer each check through the heap will take. The JVM has to suspend all other activities while running the GC. These suspensions are normally on the scale of milliseconds to a few seconds, and in this range, they are generally not noticeable. In combination with the GC subystem of the JVM, the CLC Workbenches and Servers use a sophisticated caching system, moving unneeded data out of memory and into temporary disk storage. The result of this caching system means that there will not be any noticeable performance improvement with a Java heap space larger than 50GB, while GC pauses may become more noticeable.

The 50% of physcial RAM suggestion is made because most CLC Workbenches and CLC Servers have some optimized external binaries responsible for compute-heavy tasks such as read mapping and de novo assembly. These external binaries are not affected by the maximum heap space limit. They request memory from the OS, just like any other executable. Thus, this suggested 50% limit for the heap space setting for the Java Virtual Machine (JVM) is to try to ensure that the machine will have sufficient free memory to run the external binaries.

9.2 Setting the number of cores to use

A number of the algorithms, particularly in Workbenches aimed at analyzing NGS data can use all the cores available on your system for large jobs to make the analysis as fast as possible.

You can restrict the number of cores that can be used to a predefined number via a properties file. To do this, create a text file called `cpu.properties` and save it in the `settings` folder in the Workbench installation directory. The file should include one line like this:

```
maxcores = 3
```

Here the maximum number of cores that the Workbench or binary algorithms (like read mapping or de novo analysis) would use would be 3.

After this file is in place and the Workbench is restarted, it will comply with this setting. Please note that this is not a guarantee that the Workbench will never use more than the number of cores specified, but that use of any more than this would be for very brief and infrequent peaks and should not affect performance of other applications running on your system. The only exception is when a user starts several jobs to run concurrently on the Workbench.

You can download a sample `cpu.properties` file at `http://clcbio.com/files/deployment/cpu.properties`.

Chapter 10

Overview - where do we put things?

This part gives you an overview of all the parts of a CLC Workbench installation. Some parts are at the level of the computer, i.e. shared by all users, whereas other parts are at the user level.

10.1 Computer-level information

In the *Workbench installation directory*, you will find the following:

Licenses The license information depends on what kind of license you use:

- **Fixed licenses** Stored in the `licenses` folder (they are unique for each computer)
- **Floating licenses** Information about the license server connection is stored in the `license.propertie` file in the `settings` folder.

Plugins are stored in the `plugins` folder and can be copied to other computers running the same version of the Workbench.

Resources are stored in the `resource` folder and can be copied to other computers running the same version of the Workbench.

Memory allocation for the VM is stored in the `workbenchname(version\_number).vmoptions` (e.g. clcgenomicswb7.vmoptions) file under the installation area of the Workbench on Windows and Linux. On Mac OS X, it is stored in `vmoptions.txt` in the application bundle (for example /Applications/CLC Genomics Workbench 8/CLC Genomics Workbench 8.app/Contents/).

Various property files are stored in the `settings` folder. See a full overview in section 10.2

In addition, file associations for .clc files are stored in the registry database on Windows.

10.2 Property files overview

The following property files are all stored in the `settings` folder of the Workbench installation, and can be added to a profile when using *Java Web Start* (see section **??**).

policy.properties for specifying if certain options should be removed from the Workbench (see section 7).

path.properties for specifying default data location (see section 8.1.2) and where to store temporary data (see section 8.3.1)

serverinfo.properties specify host and port for connecting to a CLC server (see section 6).

proxy.properties holds information about proxy server (when the Workbench needs access to online services). Read more in the Workbench user manual at `http://clcsupport.com/clcgenomicsworkbench/current/index.php?manual=Network_configuration.html`

license.properties holds information about the host and port of a license server that should be used (see section 3.1).

cpu.properties specifies the maximum number of cores to be used by the Workbench (see section 9.2).

10.3 User-level information

The user-level information is found in the application data folder:

Windows 2000 `C:\Documents and settings\username\Application data\CLC bio`

Windows Vista, 7 and 8 `C:\Users\username\Appdata\Roaming\CLC bio`

Mac OS X `User home/Library/Application Support/CLC bio`

Linux `$HOME/.clcbio`

In this folder, the following information may be useful to you:

User settings The user settings file stores information such as view settings, parameters, workspaces, user name and password to server log-ins and other settings customized by the user. The user settings file is found in the application data folder under `settings/workbench name/version name`. It is recommended to back up the user settings file. When upgrading to a new version of the Workbench, the user's old user settings file is copied by the Workbench the first time it is run.

Locations Information about which locations the user has added in the **Navigation Area** is stored in the `model_settings_300.xml` file in the `settings` folder.

Log Files If problems arise, these can be used by our support team to troubleshoot the issue. How to contact the Support team with the relevant log information is covered in our FAQ area at `https://secure.clcbio.com/helpspot/index.php?pg=kb.page&id=73`. If you wish to investigate the logs yourself, the key log files are called `clc-workbench.log`, `output.log` and `error.log`. Their location depends on the system you are working on. Details of the locations for each operating system can be found in our FAQ area at `https://secure.clcbio.com/helpspot/index.php?pg=kb.page&id=67`.

Workflows Workflows are saved in a `workflows/workbench name/version name` folder. When upgrading to a new version of the Workbench, the user's old workflows are copied by the Workbench the first time it is run, unless there are major changes to the workflow framework that break compatibility. If you are using a *CLC Genomics Server* or *CLC Science Server*, it can decrease overhead related to upgrading, and maintaining consistent versions of Workflows for all users, if Workflows are deployed and maintained on the Server rather than on the Workbench.

Index

CLC Bioinformatics Database, 5
CLC Server, 5
32-bit installer, 9
64 bit installer, 9

Back up
    user settings, 27
Back-up, 21
Block access to internet, 17

Command-line installation, 10
Cores, restrict usage, 24
CPU, restrict usage of, 24

Data storage, 19
Data structure, 19
Database
    local, 19

Floating license, 12

GHOST image, 6

Image, copy, 6
Install
    plugins, 14
    resources, 14
    workflows, 15
Introduction, 5

Java, 9
JRE, Java Runtime Environment, 9
JVM, Java Virtual Machine, 9

License order ID, 12
License server, 12
    set-up on clients, 12
Licenses, 12
    for plugins, 12
Linux installer vs. package, 9
Locations, 27
Log files, 27

Memory allocation, 23

Online tools, block access to, 17

Plugin licenses, 12
Plugin Manager, 14
Plugins, 14
Policy, 17

Quiet installation, 10

RAM, 23
Resources, 14
.rpm, Linux package, 9

Security policies, 17
Server connection, 16
.sh, Linux installer, 9
Silent installation, 10
System requirements, 6
System resources, 23

Temporary data, 21

User settings, 27

VM, Virtual Machine, 9
.vmoptions, memory allocation, 23

Workflows, 15

Xmx argument, 23