Mr.SymBioMath
High Performance, Cloud and Symbolic Computing in Big-Data Problems applied to Mathematical Modeling of Comparative Genomics
EU FP7 IAPP Project Nr. 324554

Summer School

Bioinformatics, Biomedicine & Cloud Computing

COMPARATIVE GENOMICS GUIDED EXERCISE

Version v1: August 2013
Developed by:Mr.SBM team
Report incidents to:
ortrelles@uma.es

1. Contents

This document containsstep by step information to perform a comparative genomic exercise: from original FASTA data files containing the raw sequences to the final generation of its corresponding phylogenetic tree.

Note about the summer school targets

This summer school is aimed to provide an overview on the three main fields that converge in the Mr.SymBioMath project: bioinformatics, biomedicine and Cloud computing. Sessions have been organized to provide a practical rather than theoretical approach to these fields. We have special interest in user friendly interfaces in particular in those to connect mobile-devices such as smartphones, tablets and similar, to the cloud computing environment. Therefore one of the main objectives is to describe the type of data and its representation to the groups in charge of visualization aspects.

The exercise described in this tutorial is aimed to understand the genome-comparison strategy to be implemented in the project, and to receive feedback for its better development.

Please, report bugs, recommendations and suggestions to ortrelles@uma.es

2. Getting started

2.1Software requirements

What do we need to have installed in our computer?The exercise will be performed through a remote connection; therefore we need to connect to the server and in some steps to download some data files. Please ensure you already have this software otherwise you can install SSH and FTP clients (suggestions are provided in *Annex1: Installation*)

2.2 Data and Software organization

Server	mango.ac.uma.es
User	student01
Password	to be provided in the meeting room
Directories:	see text-box (right)

`/home/mrsbm/bin`: binaries and Scripts

`/home/mrsbm/data/`: Original FASTA data files Including the dictionaries

`/home/mrsbm/src`: source code Restricted access

`/home/teacher/fungi`: `files for teacher conducted exercise`

`/home/student01/` `home directory for students accounts`

Directories organization in „mango“

```
+ home-+ mrsbm ----+ bin
       |           |
       |           + data --+ dictionaries
       |           |
       |           + src
       |
       |
       + teacher---+ fungi
       |
       |
       |
       + student01-+ nourita
       |           |
       |           + oscar
       |           |
       |           + ots
       |           |
       |           + pgalante
       |           |
|           + personalNameStud1
       |           |
       |           + personalNameStud2
       |           |
       |           + personalNameStud...
       |           |
```

Notes:

1.Use `‘ls –l dirPathway’` command for surfing the directories content

2. Refer to “*Annex2: Output files*” to have a clearer vision of what are the outcomes you should get for each execution, and the content of directories

3. In general in the UNIX/Linux environment to invoke a program you need to include the directory path where the command is stored(i.e.`‘./script’` when the script is in the working directory; or /home/mrsbm/bin/script). However, to simplify invocation the $PATHenvironment variable is modified during the login

process to include the '/home/mrsbm/bin' directory. Therefore you can invoke the script directly without the PATH name

4. Writing the full pathfile for each file in each command invocation is boring and prone to error. Thus, we provide a script (ejeStep0.csh) to create software links (in the user directory) for each file in this exercise, in order to simplify the command lines

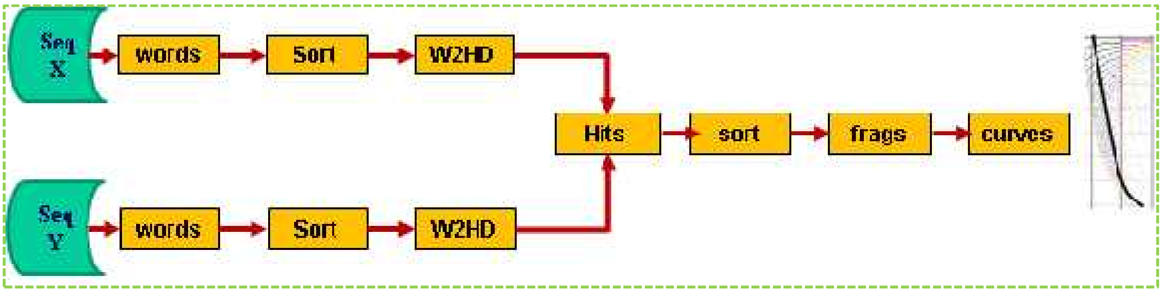
5. For benchmarking purposes you can always use the "time" command in the Linux-UNIX environment.For instance you can invoke the first script using "time ejeStep0.csh". At the end of execution the time use to be reported in three categories (full elapsed time; time running in user mode; and time running in kernel mode)--- visit "man time" for details):

```
real    0m0.006s
user    0m0.004s
sys     0m0.000s
```

6. The backslash symbol –in the command lines in this document- represents the line continues (remove the newline) as the standard of fact in Linux-UNIX

2.3 The workflow

In this exercise we will use the software developed by UMA group to obtain HSPs (High Scoring Pairs) and the dotplot viewer enhanced by RISC software. To remove the limits of memory capacity, the first procedures are an out-of-core implementation. The complete WF is composed of a set of procedures. The initial ones (see left-branches in the image) must be carried out once for each sequence. The second part of the WF is invoked once for each pair of sequences.



The process starts with two sequences to be compared.

For each of the sequences (masked or unmasked) the collection of k-mers (words of size K) is obtained (by sequence position) and ordered (sort) by word content. This output is used to build-up a hash-table or **dictionary of words** for each sequence.

Next, the identical words are used to identify the **hits**. This will be executed for both forward and reverse complement sequence. At this point the **hits** are used as seeds to extend the local un-gapped alignments (fragments) being this step the most computationally expensive.

To have the total number of fragments we will combine the group of hits from both forward and reverse complement strands and finally we will visualize the genomic comparison result as a Dot Plot and a phylogenetic tree.

(Please refer to supplementary information for details).

3. Guided Exercise

The exercise consists on a comparison between 6 different Mycoplasma genomes belonging to three species (*bovis*, *fermentans* and *agalactiae*). We will perform the comparison for both the forward and the reverse complement strand.

> See: Franciele Maboni Siqueira, et al. (2012); “New insights on the biology of swine respiratory tract mycoplasmas from a comparative genome analysis”; *BMC Genomics* 2013,14:175doi:10.1186/1471-2164-14-175
> The electronic version of this article is the complete one and can be found online at: http://www.biomedcentral.com/1471-2164/14/175

The exercise has been already tested and the intermediate/final files from the different steps are available to be used as control in case of need.

Now we are ready to start the exercise!!

[S1] As first action after login, proceed to **create your own working directory** with ‘`mkdirmydirectory`’ command.

Go to your directory ‘`cd mydirectory`’. All your actions will be described in this document as you would be executing from this position

Server	mango.ac.uma.es
User	student01
Password	[redacted]

[S2]Creating direct access to data files.

In order to have a direct access to the data files in your working directory you must execute the script “ejeStep0.csh” that creates a soft-links to the original datafiles.

ejeStep0 script needs two parameters: the directory with the files and a prefix for the names of the links (here it is assumed **Myco** is used as prefix)

```
    ejeStep0.csh “/home/mrsbm/data/N*"Myco
```

This will create soft-links to all files (starting with N) in the `/home/mrsbm/data` directory and will include the links in your working directory (remind ‘`ls-l pathway`’command can be used to visualize directory contents); and results are available in Annex 2 --- hereafter this recommendation will be omitted)

[S3]Creation of the k-mer dictionaries.

For those sequences we want to comparewe need the collection of k-mers or words. We execute the “ejeStep1.csh”algorithm.As the comparison will be performed for both forward and reverse complement strand, we also need to generate the k-mers dictionaries for the reverse complement strand for at least one of the sequences to be compared. In this exercise we will provide you with the reverse complement strand (*r.fasta).

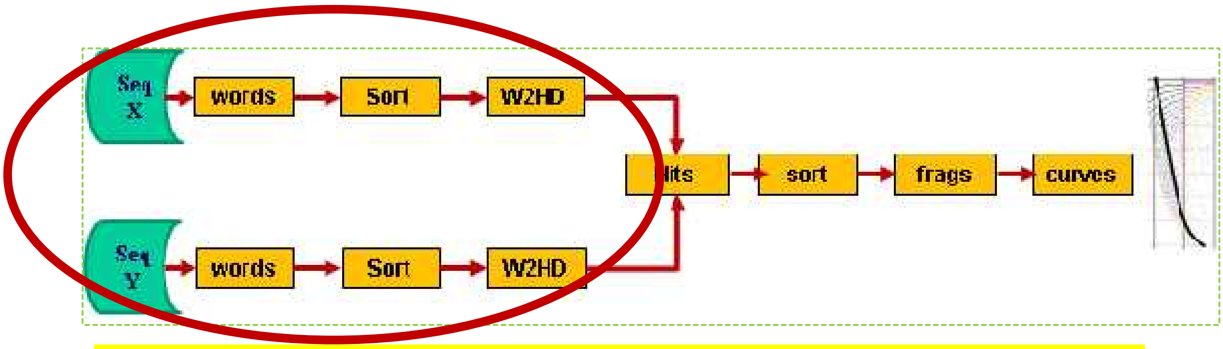
However, you have the possibility to get the reverse complement sequence yourself by executing the algorithm getReverseComplement.pl under /home/mrsbm/bin

The execution command lines are

```
ejeStep1.csh   Myco01.fasta
ejeStep1.csh   Myco02.fasta
ejeStep1.csh Myco02r.fasta
```

After the execution,in the working directory, 3 new files for each input will be generated, the extensions corresponds to files for the generated dictionary of kmers, naming [*.d2hW], [*.d2hp] and [*.words.sort] where “*” means a “prefix” used to define the dictionary content.

File name	Content
[*.d2hW]	Collection of words
[*.d2hp]	Positions on the Hash table
[*.words.sort]	Sorted words



**TEACHER: display the content of script1 (ejeStep1.csh) – identify the commands.*

And finally, in our case a total of 9 files:

Sequence 1	**Sequence2**	**Sequence3** (rev.comp. Seq. 2)
Myco01.d2hW	Myco02.d2hW	Myco02r.d2hW
Myco01.d2hP	Myco02.d2hP	Myco02r.d2hP
Myco01.words.sort	Myco02.words.sort	Myco02r.words.sort

[S4] Getting fragments from hits (sequences comparison)

We will use“ejeStep2.csh” script.First we will compare just forward against forward strands. Then we will do the comparison between the forward against reverse strand of the Sequence2. The script needs some parameter for the different steps.

These parameters are the following:

```
[script][Prefix-Dictionary1][ Prefix-Dictionary 2] \
      [Min_Length][%similarity][k/4 word size] \
      [fixed length 1=yes;2=no] [direction f=forward r=reverse]
```

The script manages to complete the extension of dictionary files from the prefixes.

The execution line for comparison of forward against forward strand and forward against reverse are:

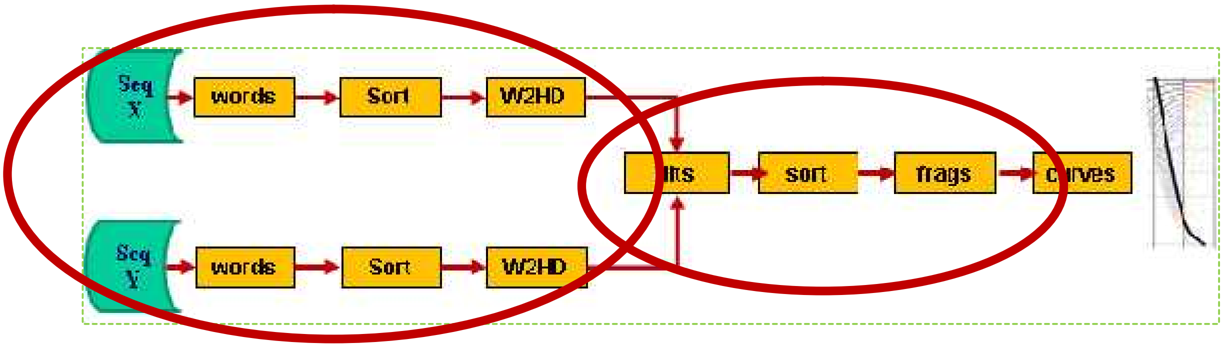
```
      ejeStep2.csh Myco01 Myco02 65 30 8 1 f
      ejeStep2.csh Myco01 Myco02r 65 30 8 1 r
```

In this case, the parameters mean we are performing a comparison between Myco01 and Myco02 filtering fragments with a fragment of minimal less than to 65, minimal similarity of 30% and a word length of 32 (8*4) kmer with length 8 on the forward (f) and the reverse (r) strand.

*TEACHER. Explain the parameters used in ejeStep2.csh

Once executed, 4 new files for each comparison will be automatically generated:

File name	Content
[*-K8.hits.order]	Hits between the two forward sequences
[*r-K8.hits.order]	Hits between the forward and the revere complement sequences
[*-K8.hits.order.filter]	Ordered hits according to length between the two forward sequences
[*r-K8.hits.order.filter]	Ordered hits according to length the forward and the revere complement sequences
[*-L65-S30-K8.sf.frags]	Common fragments between the two forward sequences
[*-L65-S30-K8.sr.frags]	Common fragments between the forward and the revere complement sequences
[*sf.frags.INF]	Information about the number of hits and common hits for the forward sequences
[*sr.frags.INF]	Information about the number of hits and common hits for the forward and reverse complement sequences



[S5]Combining forward and reverse fragments

To obtain a unique file for both, forward-forward and forward-reverse complementstrandsexecute `[script] [file1.frags] [file2.frags] [outputName]`

The execution line is

```
combineFrags Myco01-Myco02-L65-S30-K8-sf.frags  \
Myco01-Myco02r-L65-S30-K8-sr.frags  \
Myco01Myco02TFrags
```

[S6]Visualization (ongoing work)

We provide two ways to visualize the fragments
1.- Static image: by using the algorithm “af2pngrev” with execution line:

`[af2pngrev][file1TotalFragments][OutputName][nameX][nameY] Colorstyle` (use 1 as default color style)

af2pngrevMyco01Myco02TFrags Myco01Myco02TFrags.PNG Myco01 Myco02 1

File generated: Myco01Myco02TFrags.png

Download the generated .PNG file to your local computer and use a standard viewer to observe the common fragments between the two sequences.

2.- Interactive visualization: using our java (jar) application (work in progress) (see – at the end of this document) the “Comparative Genomics Analysis Tool(Draft version)” user guide.

[S7]Distance matrix and phylogenetic tree

1. Distance matrix.

The final objective in this exercise is to generate a phylogenetic tree so we can infer similarities between the different species. To do so we have to calculate the inter-genome distance between the 2 sequences by executing the “gDistanceNEW” program.

The execution lines are

```
      gDistanceNEW Myco01-Myco02-L65-S30-K8-sf.frags 0.8 0.5
      gDistanceNEWMyco01-Myco02-L65-S30-K8-sr.frags 0.8 0.5
```

The parameters are the following

```
   [script] [frag.file-forward][Translocat.penalty] [Invers.penalty]
```

Teacher explain what do the parameters for distance matrix calculation mean

For this example we will weigh the translocations with a 0.8 over the normal, and the inversions of a 0.5 though these parameters can change.

We shall obtain the following distance matrix for all 6 genomes.

	Myco01	Myco02	Myco03	Myco04	Myco05	Myco06
Myco01	0	0.35	3.76	1.10	3.77	1.08
Myco02	0.35	0	4.01	1.08	4.03	1.12
Myco03	3.76	4.01	0	4.29	0.28	4.17
Myco04	1.10	1.08	4.29	0	4.40	0.49
Myco05	3.77	4.03	0.28	4.40	0	4.17
Myco06	1.08	1.12	4.17	0.49	4.17	0

Genomic distances matrix (available at /home/mrsbm/data)

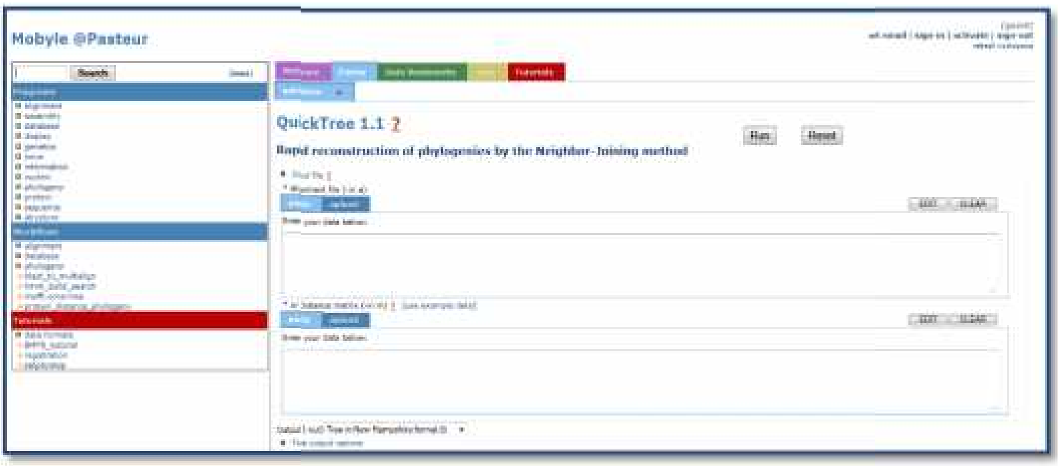
2. Phylogenetic tree.

Note: Java software must be installed in your local computer. Please refer to the annexes.

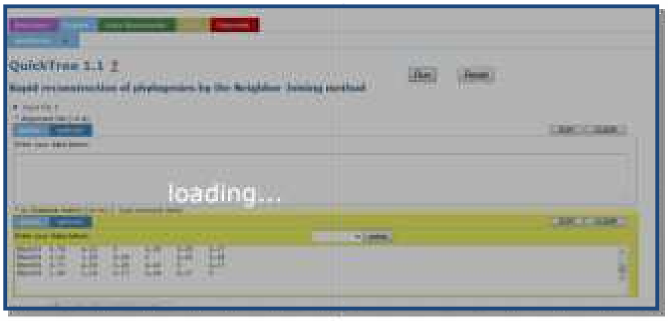
We will create the phylogenetic tree file following the instructions under the mobile Pasteur portal for bioinformatics analysis.

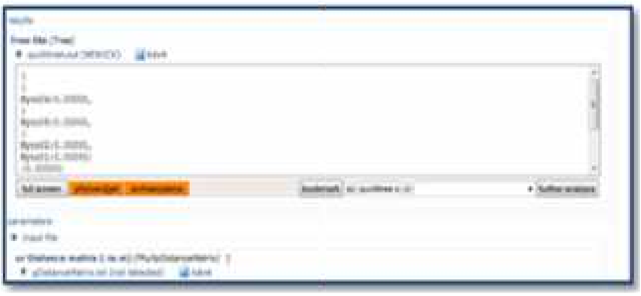
In order to do so we need to have our genetic distances matrix computed for all 6 genomes. Download this table from /home/mrsbm/data/mycoDistMat.txt to your local machine through putty.

Go to http://mobyle.pasteur.fr/cgi-bin/portal.py#forms::quicktree and copy / paste the "mycoDistMat.txt" text file containing the inter-.genome distances. A similar interface will be displayed in your screen:


Mobyle @Pasteur
Search
QuickTree 1.1
Run
Reset
Rapid reconstruction of phylogenies by the Neighbor-Joining method
Tutorials

Select as input "Distance matrix" and upload the MycoDistMat.txt file.

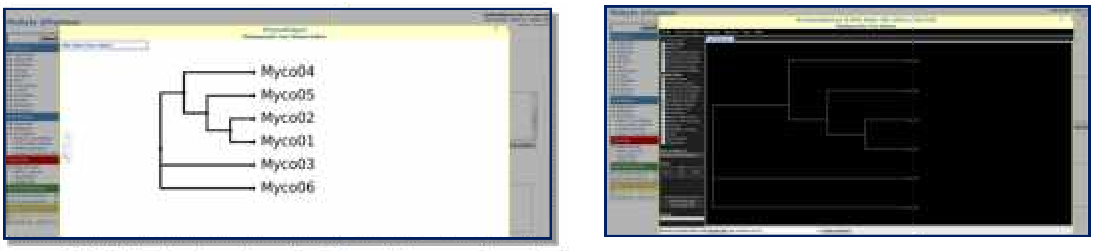

QuickTree 1.1
Run
Reset
Rapid reconstruction of phylogenies by the Neighbor-Joining method
loading...



Once our data file is uploaded we RUN the program leaving the default settings and parameters.

The program will generate a quicktree.out (NEWICK) file. Save this file to your local working directory for further visualizations. You can also download it from /home/mrsbm/data/quicktree.out to your local machine.

Now we can visualize the generated phylogenetic tree by using *Phylowidget* or *archaeopteryoptions*

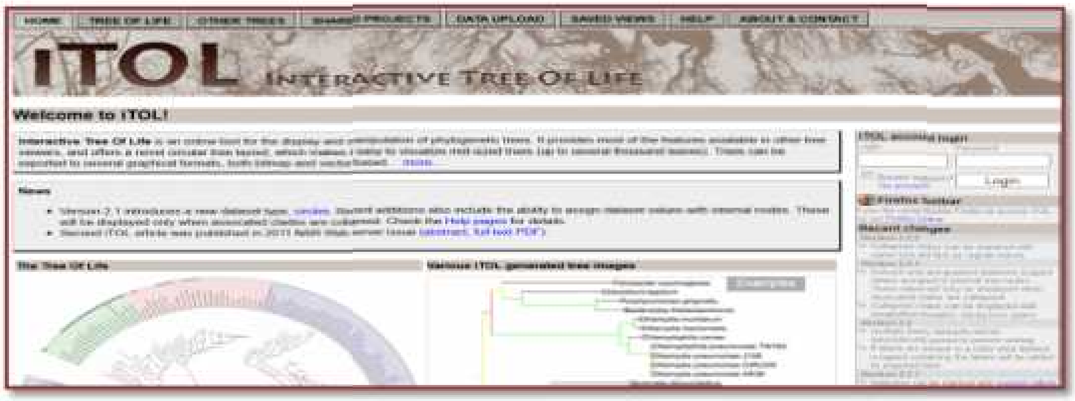


Mobile display showing the phylogenetic relationship between the 6 mycoplasmas genomes

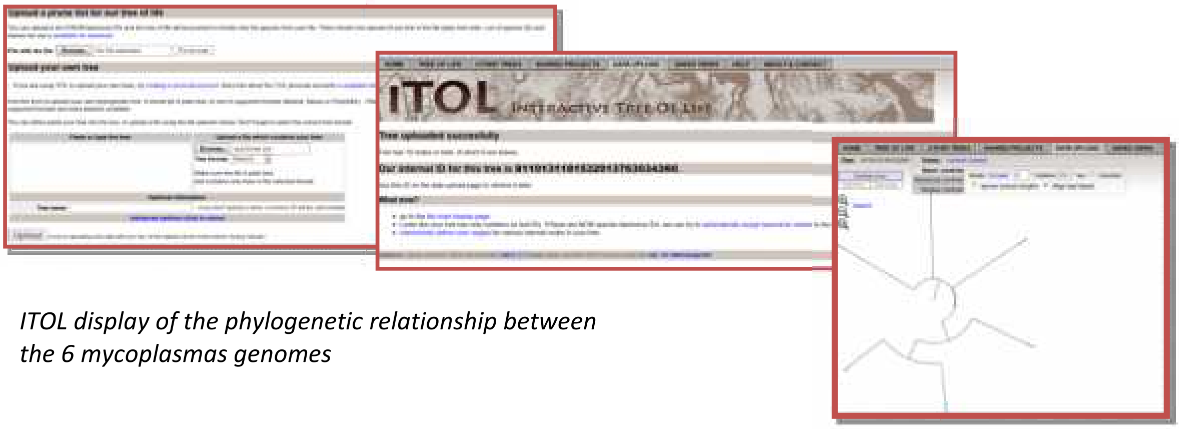
3. Other visualization options *(not to be discussed in the seminar)*

Additionally we can visualize the phylogenetic relationship between the 6 mycoplasmas by **ITOL**, the Interactive Tree of Life.

For this purpose we need to upload the NEWICK file (quicktree.out) with the quicktreevalues obtained before. Go to http://itol.embl.de/index.shtml



Click on "Data upload" on the menu and chose the option "upload your own tree". You can either paste the values from the quicktree file or browse the file from your local machine.



ITOL display of the phylogenetic relationship between the 6 mycoplasmas genomes

Málaga / Linz, August 2013

Mr.SymBioMath
High Performance, Cloud and Symbolic Computing in Big-Data Problems applied to Mathematical Modeling of Comparative Genomics
EU FP7 IAPP Project Nr. 324554

Summer School

Bioinformatics, Biomedicine & Cloud Computing

COMPARATIVE GENOMICS GUIDED EXERCISE

ANNEXES

Version v1: August 2013
Developed by:Mr.SBM team
Report incidents to:
ortrelles@uma.es

ANNEX 1: Software Installation

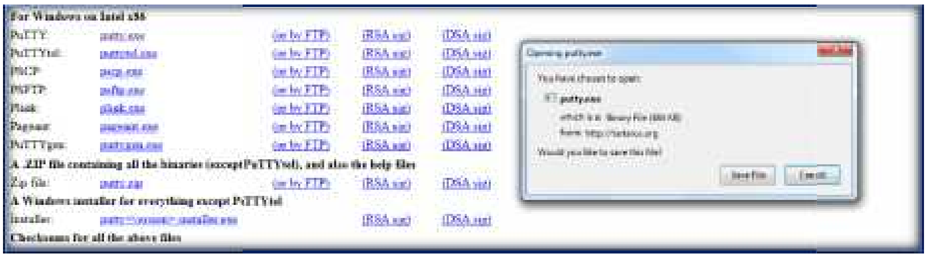
Before starting with the exercise we have to make sure we do have all plugins and required software.

Contents

1. Installing Putty

Putty is an *ssh* client that allows us to connect to a remote server through our local machine

Go to http://www.chiark.greenend.org.uk/~sgtatham/putty/
Click on download and choose the binaries for your operative system (is recommended to download the latest development snapshot) and save the file.



For Windows on Intel x86

PuTTY:	putty.exe	(or by FTP)	(RSA sig)	(DSA sig)
PuTTYtel:	puttytel.exe	(or by FTP)	(RSA sig)	(DSA sig)
PSCP:	pscp.exe	(or by FTP)	(RSA sig)	(DSA sig)
PSFTP:	psftp.exe	(or by FTP)	(RSA sig)	(DSA sig)
Plink:	plink.exe	(or by FTP)	(RSA sig)	(DSA sig)
Pageant:	pageant.exe	(or by FTP)	(RSA sig)	(DSA sig)
PuTTYgen:	puttygen.exe	(or by FTP)	(RSA sig)	(DSA sig)

A .ZIP file containing all the binaries (except PuTTYtel), and also the help files

Zip file:	putty.zip	(or by FTP)	(RSA sig)	(DSA sig)

A Windows installer for everything except PuTTYtel

Installer:	putty-[illegible]-installer.exe		(RSA sig)	(DSA sig)

Checksums for all the above files

Opening putty.exe

You have chosen to open:

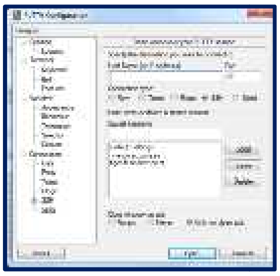
putty.exe

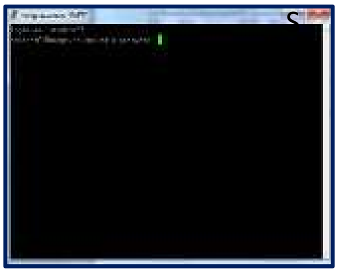
which is a: Binary File

from: http://tartarus.org

Would you like to save this file?

Run the software and follow the instructions.
Lunch Putty: Now we specify the destination we want to connect to by writing the host name and port number. Click on "*open*" to open a terminal.



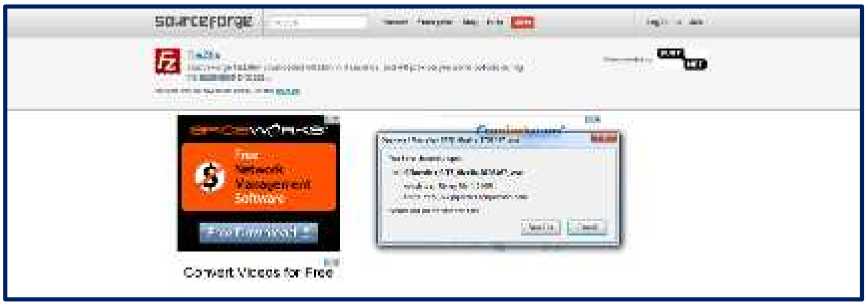


Finally we log in by typing the user name and password provided in this course

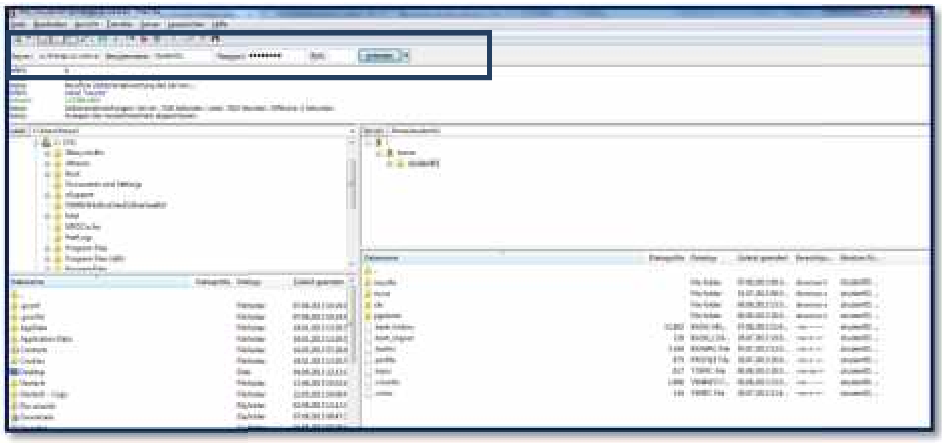
2. Installing Filezila

There are several FTP free open source clients. One of the most used is FTP client Filezilla. It allows us to transfer files between our local computer and a server on the Internet.

Go to the site https://filezilla-project.org/
Click on download and choose the binaries for your operative system and save the file.

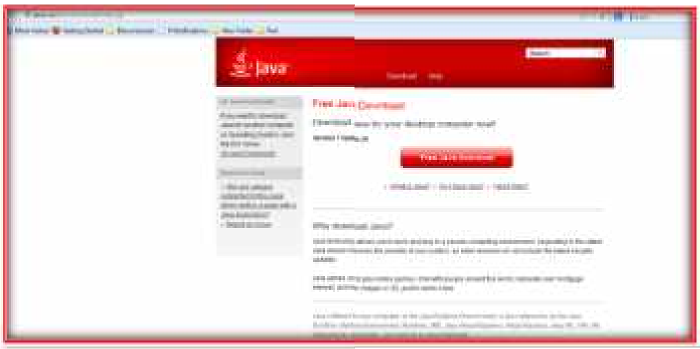


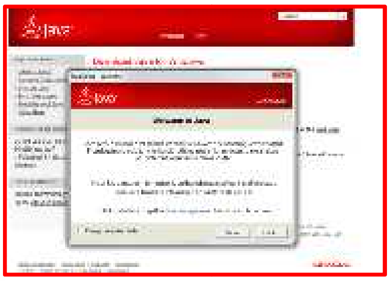
Run the software and follow the instructions.
Lunch Filezilla: Open the ftp client and log in by typing the host name, user, password and port number.



3. Installing Java

In order to visualize the phylogenetic trees we do need to have preinstalled some plugins. Go to http://java.com/en/download/index.jsp Download and follow the instructions.





ANNEX 2: Output files

This annex contains the files that are obtained in each execution step in the comparative genomic exercise.

Contents

[S1] Create your own working directory

Command lines

```
mkdirmydirectory
ls –l
```

```
drwxrwxr-x 2 student01 student01 4096 jul 30 15:04 nourita
drwxrwxr-x 2 student01 student01 4096 jul 30 14:28 oscar
```

[S2]Renaming and Creating direct access to data files.

Command lines

```
ejeStep0.csh "/home/mrsbm/data/N*" Myco
ls-l
```

```
lrwxrwxrwx 1 student01 student01 34 jul 31 10:24 Myco01.fasta -> /home/mrsbm/data/NC-
009497.1.fasta

lrwxrwxrwx 1 student01 student01 34 jul 31 10:24 Myco02.fasta -> /home/mrsbm/data/NC-
013948.1.fasta

lrwxrwxrwx 1 student01 student01 44 jul 31 10:24 Myco02r.fasta -> /home/mrsbm/data/NC-
013948.1-revercomp.fasta

lrwxrwxrwx 1 student01 student01 34 jul 31 10:24 Myco03.fasta -> /home/mrsbm/data/NC-
014552.1.fasta

lrwxrwxrwx 1 student01 student01 44 jul 31 10:24 Myco03r.fasta -> /home/mrsbm/data/NC-
014552.1-revercomp.fasta

lrwxrwxrwx 1 student01 student01 34 jul 31 10:24 Myco04.fasta -> /home/mrsbm/data/NC-
014760.1.fasta

lrwxrwxrwx 1 student01 student01 44 jul 31 10:24 Myco04r.fasta -> /home/mrsbm/data/NC-
014760.1-revercomp.fasta

lrwxrwxrwx 1 student01 student01 34 jul 31 10:24 Myco05.fasta -> /home/mrsbm/data/NC-
014921.1.fasta
```

```
lrwxrwxrwx 1 student01 student01 44 jul 31 10:24 Myco05r.fasta -> /home/mrsbm/data/NC-
014921.1-revercomp.fasta

lrwxrwxrwx 1 student01 student01 34 jul 31 10:24 Myco06.fasta -> /home/mrsbm/data/NC-
015725.1.fasta

lrwxrwxrwx 1 student01 student01 44 jul 31 10:24 Myco06r.fasta -> /home/mrsbm/data/NC-
015725.1-revercomp.fasta
```

[S3]Creation of the k-mer dictionaries.

Command lines

```
             ejeStep1.csh Myco01.fasta
             ejeStep1.csh Myco02.fasta
             ejeStep1.csh Myco02r.fasta
```

```
Myco01.d2hW              Myco02.d2hW           Myco02r.d2hW
Myco01.d2hP              Myco02.d2hP           Myco02r.d2hP
Myco01.words.sort        Myco02.words.sort     Myco02r.words.sort
```

[S4] Getting fragments from hits (sequences comparison)

Command lines

```
      ejeStep2.csh Myco01 Myco02 65 30 8 1 f
      ejeStep2.csh Myco01 Myco02r  65 30 8 1 r
```

```
Myco01-Myco02-K8.hits.order                Myco01-Myco02r-K8.hits.order
Myco01-Myco02-K8.hits.order.filter         Myco01-Myco02r-K8.hits.order.filter
Myco01-Myco02-L65-S30-K8-sf.frags          Myco01-Myco02r-L65-S30-K8-sr.frags
Myco01-Myco02-L65-S30-K8-sf.frags.INF      Myco01-Myco02r-L65-S30-K8-sr.frags.INF
```

[S5]Combining forward and reverse fragments

Command lines

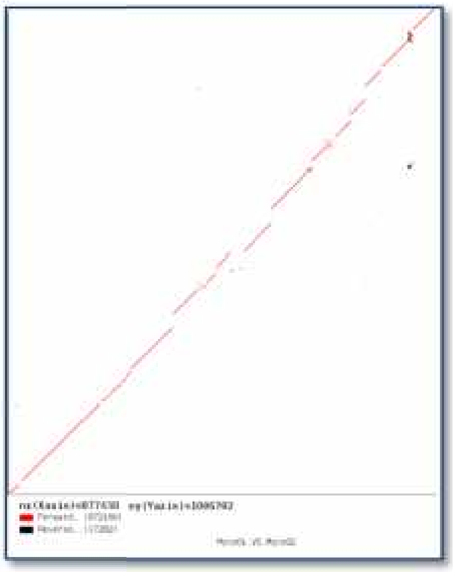
```
      combineFrags   Myco01-Myco02-L65-S30-K8-sf.frags\
      Myco01-Myco02r-L65-S30-K8-sr.frags\
                                     Myco01Myco02TFrags
                   ls -l
```

```
-rw-rw-r-- 1 student01 student01  7019240 jul 31 10:44 Myco01.d2hP
-rw-rw-r-- 1 student01 student01 13913984 jul 31 10:44 Myco01.d2hW
lrwxrwxrwx 1 student01 student01       34 jul 31 10:24 Myco01.fasta -> /home/mrsbm/data/NC-
009497.1.fasta
-rw-rw-r-- 1 student01 student01  6037440 jul 31 11:18 Myco01-Myco02-K8.hits.order
-rw-rw-r-- 1 student01 student01    91160 jul 31 11:18 Myco01-Myco02-K8.hits.order.filter
-rw-rw-r-- 1 student01 student01    25384 jul 31 11:09 Myco01-Myco02-L65-S30-K8-sf.frags
-rw-rw-r-- 1 student01 student01      429 jul 31 11:09 Myco01-Myco02-L65-S30-K8-sf.frags.INF
-rw-rw-r-- 1 student01 student01    25384 jul 31 11:18 Myco01-Myco02-L65-S30-K8-sr.frags
-rw-rw-r-- 1 student01 student01      429 jul 31 11:18 Myco01-Myco02-L65-S30-K8-sr.frags.INF
-rw-rw-r-- 1 student01 student01    50760 jul 31 11:27 Myco01Myco02TFrags
-rw-rw-r-- 1 student01 student01 14038480 jul 31 10:44 Myco01.words.sort
-rw-rw-r-- 1 student01 student01  8053352 jul 31 11:04 Myco02.d2hP
-rw-rw-r-- 1 student01 student01 15138768 jul 31 11:04 Myco02.d2hW
lrwxrwxrwx 1 student01 student01       34 jul 31 10:24 Myco02.fasta -> /home/mrsbm/data/NC-
013948.1.fasta
-rw-rw-r-- 1 student01 student01  8053352 jul 31 10:59 Myco02r.d2hP
-rw-rw-r-- 1 student01 student01 15138768 jul 31 10:59 Myco02r.d2hW
lrwxrwxrwx 1 student01 student01       44 jul 31 10:24 Myco02r.fasta -> /home/mrsbm/data/NC-
013948.1-revercomp.fasta
-rw-rw-r-- 1 student01 student01 16106704 jul 31 10:59 Myco02r.words.sort
-rw-rw-r-- 1 student01 student01 16106704 jul 31 11:04 Myco02.words.sort
lrwxrwxrwx 1 student01 student01       34 jul 31 10:24 Myco03.fasta -> /home/mrsbm/data/NC-
014552.1.fasta
lrwxrwxrwx 1 student01 student01       44 jul 31 10:24 Myco03r.fasta -> /home/mrsbm/data/NC-
014552.1-revercomp.fasta
lrwxrwxrwx 1 student01 student01       34 jul 31 10:24 Myco04.fasta -> /home/mrsbm/data/NC-
014760.1.fasta
lrwxrwxrwx 1 student01 student01       44 jul 31 10:24 Myco04r.fasta -> /home/mrsbm/data/NC-
014760.1-revercomp.fasta
lrwxrwxrwx 1 student01 student01       34 jul 31 10:24 Myco05.fasta -> /home/mrsbm/data/NC-
014921.1.fasta
lrwxrwxrwx 1 student01 student01       44 jul 31 10:24 Myco05r.fasta -> /home/mrsbm/data/NC-
014921.1-revercomp.fasta
lrwxrwxrwx 1 student01 student01       34 jul 31 10:24 Myco06.fasta -> /home/mrsbm/data/NC-
015725.1.fasta
lrwxrwxrwx 1 student01 student01       44 jul 31 10:24 Myco06r.fasta -> /home/mrsbm/data/NC-
015725.1-revercomp.fasta
```

[S6]Visualization

Command line

```
af2pngrev  Myco01Myco02TFrags  \
Myco01Myco02TFrags.PNG\
Myco01 Myco02 1
```



[S7]Distance matrix and phylogenetic tree

1.- Distance matrix.

In this step we do not generate any file but a value that appears once we execute the following

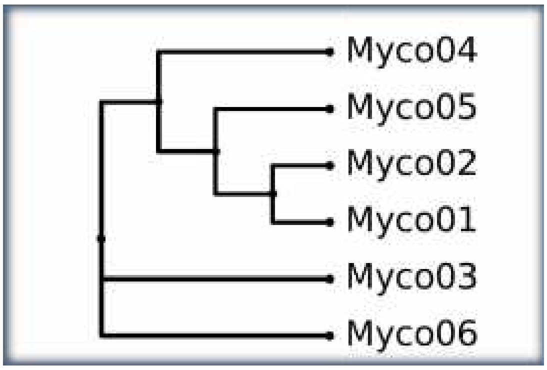
Command line

```
            gDistanceNEW Myco01Myco02TFrags 0.8 0.5

Distance: 0.060648
```

2.- Phylogenetic tree.

We need to have more than 2 genomic distances computed to create a phylogenetic tree, in this case we provide the user with all pre-computed genomic distances values corresponding to the 6 mycoplasmas genomes.



Mr.SymBioMath
High Performance, Cloud and Symbolic Computing in Big-Data Problems applied to Mathematical Modeling of Comparative Genomics
EU FP7 IAPP Project Nr. 324554

MARIE CURIE ACTIONS

SEVENTH FRAMEWORK PROGRAMME

Summer School

Bioinformatics, Biomedicine & Cloud Computing

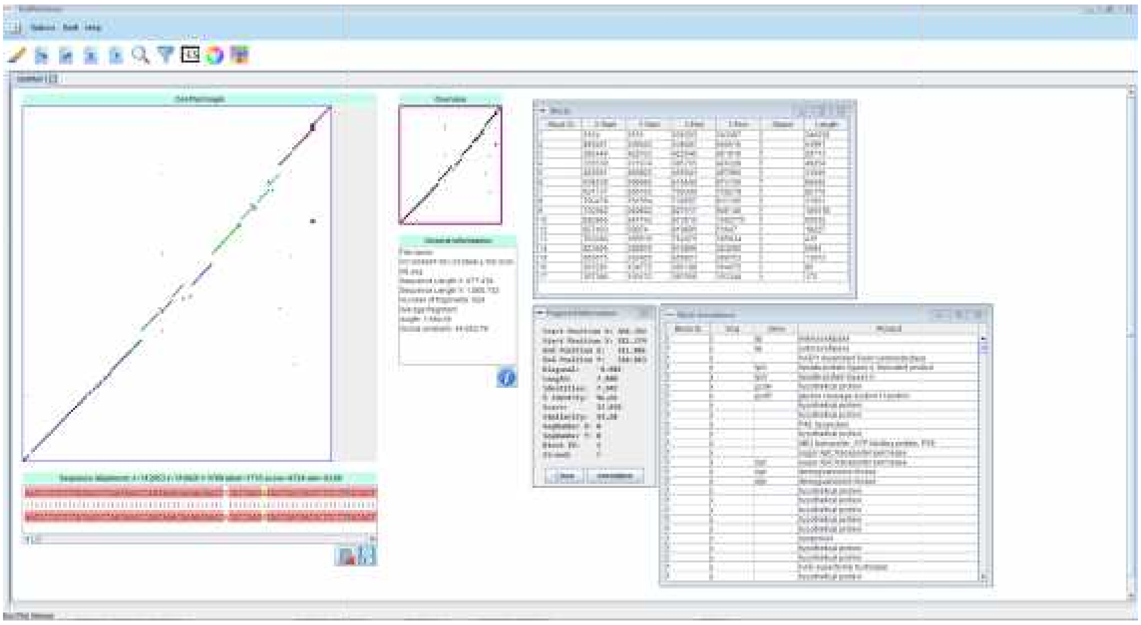
Visualization

GUIDED EXERCISE

ComparativeGenomics Analysis Tool

(Draft version)

Version v1: August 2013
Developed by:Mr.SBM team
Report incidents to:ortrelles@uma.es



CGTA: Comparative Genomics Analysis Toolis a first prototype for the Inter-Genome comparison viewer. It is also planed to support multiple genome and meta-genome comparison studies

This version is deployed as a first Beta version and it is an ongoing work being carried out in the Mr.SBM project

Step 0: Prepare some additional files: blocks and annotations

The HSP workflow starts with two sequences and ends with a collection of HSPs containing information obtained from the comparison procedure (i.e. coordinates, score, identities, etc.).

A set of post-processing algorithms have been developed to provide more useful information; such as, synteny blocks, functional annotations, gapped-fragments, etc. Al these files are available for download.

Step 1: Download the software and data files for the exercise

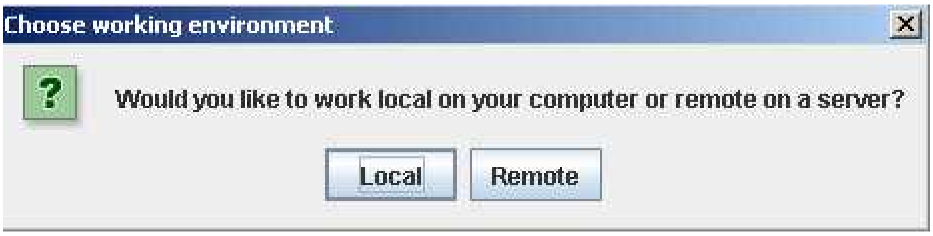
Software: dotplot.jar from XXXX
Data files: In this exercise the following files will be used:
1) Mycoplasma: NC-009497 vs. NC-013948 strains

File	**Content**
`NC-009497-NC-013948-L100-S30-K8.seg`	HSP including blocks identification
`NC-009497.FASTA`	sequence (X)
`NC-013948.FASTA`	sequence (Y)
`NC-009497-NC-013948-L100-S30-K8.GFF`	Functional annotations

2) Fungi analysis:

File	**Content**
	HSP including blocks identification
	sequence (X)
	sequence (Y)
	Functional annotations

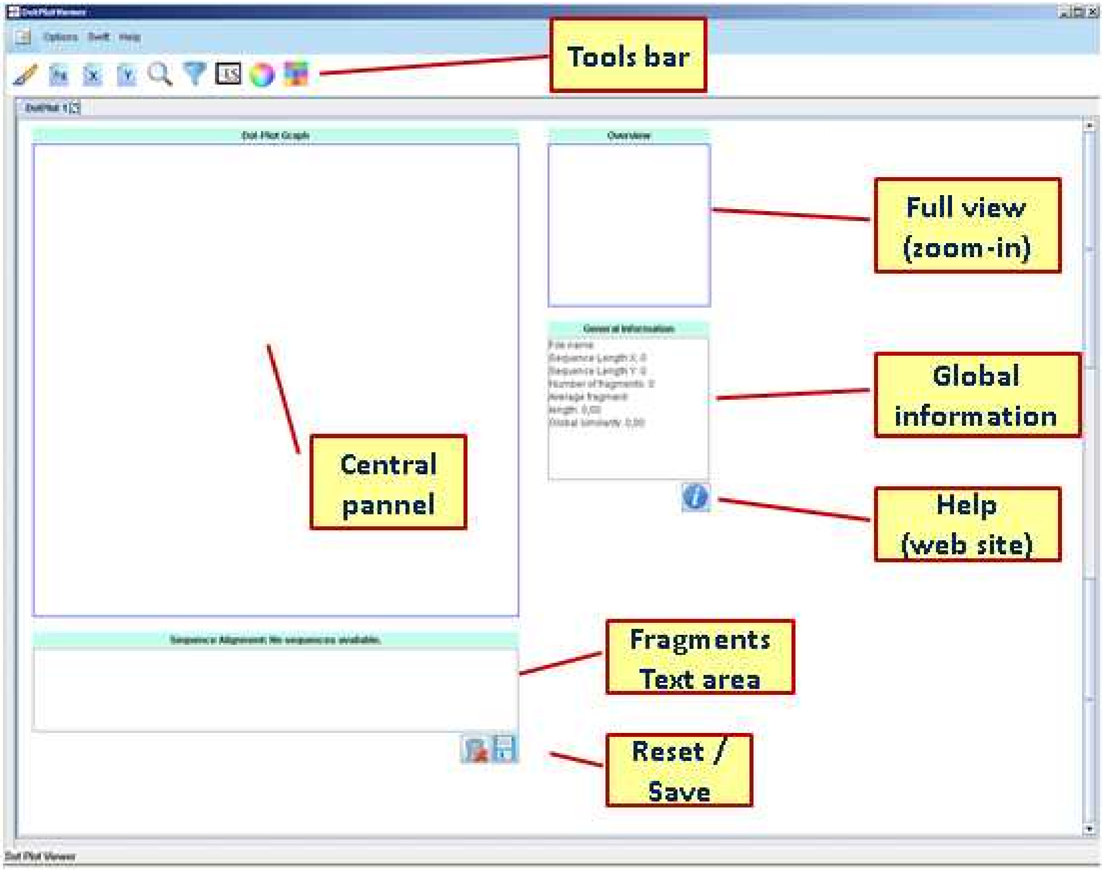
Step 2: invoke the program and choose the “Local” mode



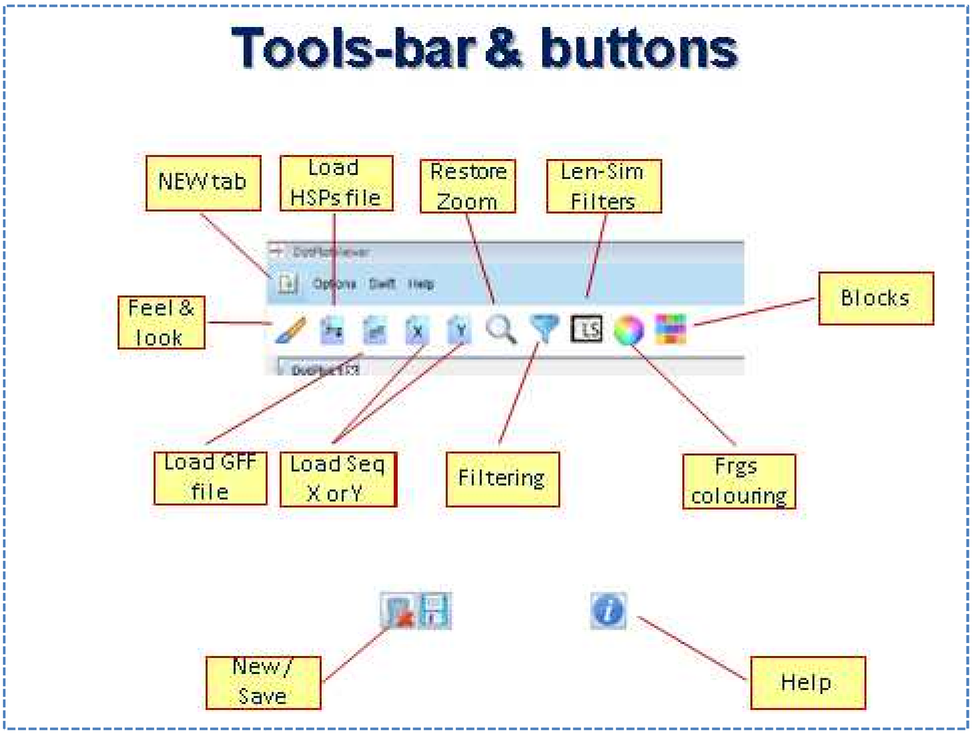
Choose working environment

Would you like to work local on your computer or remote on a server?

The main screen will be displayed

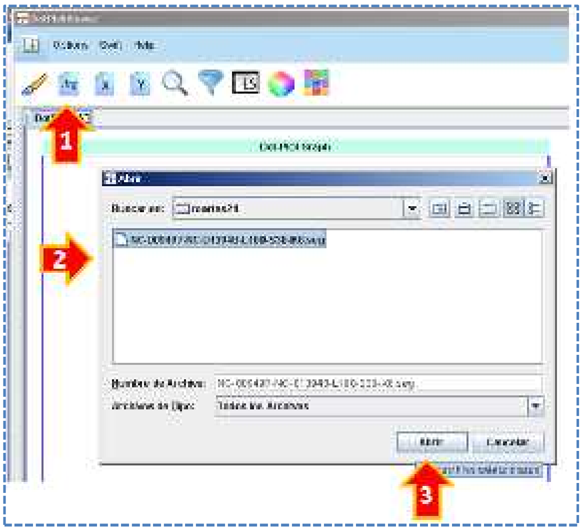


File name:
Sequence Length X: 0
Sequence Length Y: 0
Number of fragments: 0
Average fragment length: 0,00
Global similarity: 0,00



Tools-bar & buttons

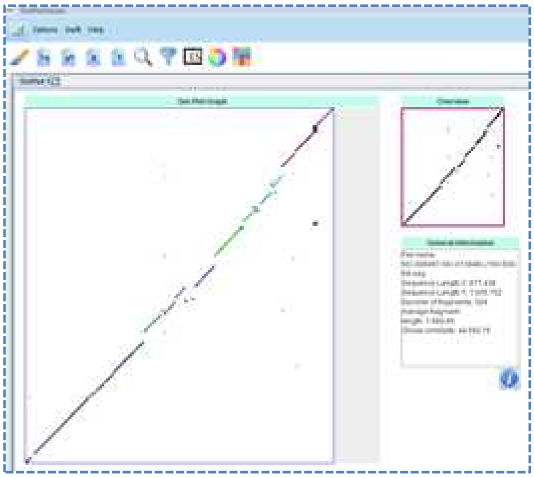
Step 3:Getting Started



Load Fragments (HSP) from a file
Use the "Frg" button
Browse your file system
Open the selected file
(Use Mycoplasma data in this exercise)

Note: the Fragments file must comply with the fragments data structure of the workflow

The following images correspond to the collection of files described in Step 1



The collection of HSPs (Fragments) is displayed on the main screen together with the global information and the full view

Observe, at this moment no information about the sequences is available, thus the fragments text-area is empty.

Hands-on: play with "zoom, filtering, block coloringand frags information" options

Step 4: Load the sequences. Use the corresponding X / Y buttons to load the sequences for this HSP collection. Browse your file system and load the files

Note: the Sequences must be in FASTA format

At this point the "fragments text-area" becomes active and the sequences are displayed in it.

This area is more interesting when used to display the fragment composition. Click in on particular fragment and observe the un-gapped alignment (text area) and the fragment information

Step 5: Load functional annotations. Use the corresponding X / Y buttons to load the sequences for this HSP collection. Browse your file system and load the GFF file corresponding to this experiment.

Apparently nothing changes, however, now the blocks information is available and this fact is reflected in:
a) when you click on the “Colouring button”. At this moment the “Block summary table”
b) when you click on a Fragment, and the fragment information is displayed the “Annotation” button is enabled, so, you can click it and the Annotations for this specific block is displayed (remember, block number 0 do not have annotations)