



Introduction to Bash and Slurm

EDB/LEFE training - session 1



Laboratoire écologie
fonctionnelle
et environnement

Anne-Sophie Benoiston
anne-sophie.benoiston@univ-tlse3.fr
Jordi Salmona
jordi.salmona@gmail.com
Imogen Dumville
imogen.dumville@gmail.com

Who are we?

Anne-Sophie Benoiston



Bioinformatics engineer



Jordi Salmona



Post-doc

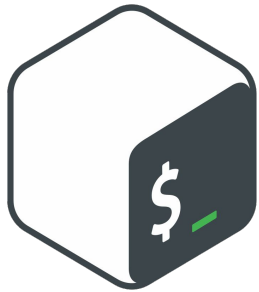
Imogen Dumville



M.Sc. Intern



Organization of the session



BASH
THE BOURNE-AGAIN SHELL

9:30 - 12:00



13:30 - 16:00

theory + exercises



10:40
14:40



BASH
THE BOURNE-AGAIN SHELL

Access the cluster to practice



Remotely accessing servers: putty, ssh, scp, FileZilla, MobaXterm

Command line window interfaces:

- Putty or MobaXterm (Windows), + `ssh` command
- Command line terminal (Linux/Apple) + `ssh` command

File transfer:

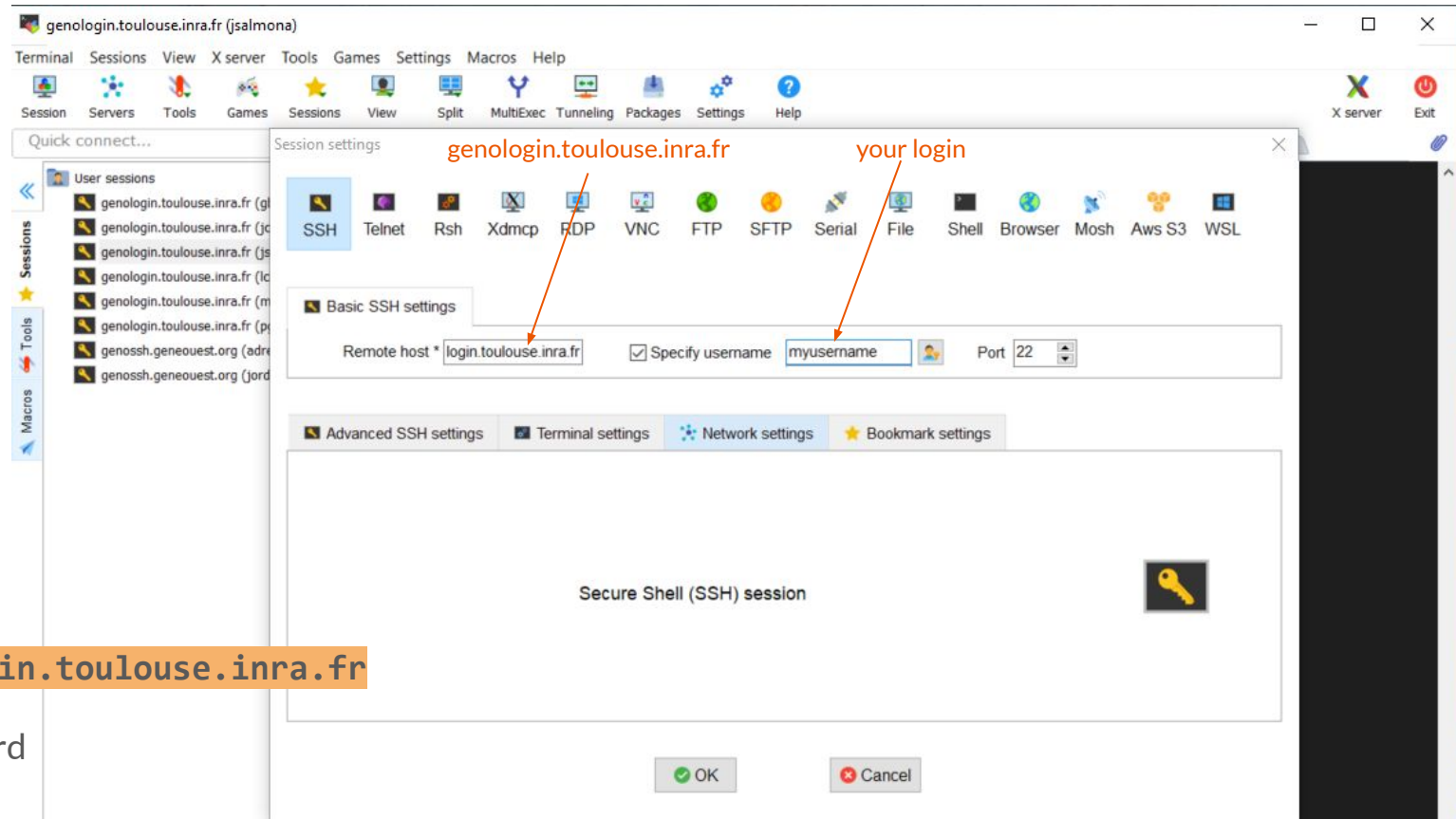
- FileZilla (all OS), MobaXterm, winscp (windows)
- `scp` command
- from folder interface (linux)

Vocabulary:

- FTP: file transfer protocol
- SSH: secure shell
- SCP: secure copy

This course: MobaXterm chosen for its versatility (for Windows users)

Practice to learn: connect to your Genotoul account



Open a terminal

Connect with ssh:

```
ssh login@genologin.toulouse.inra.fr
```

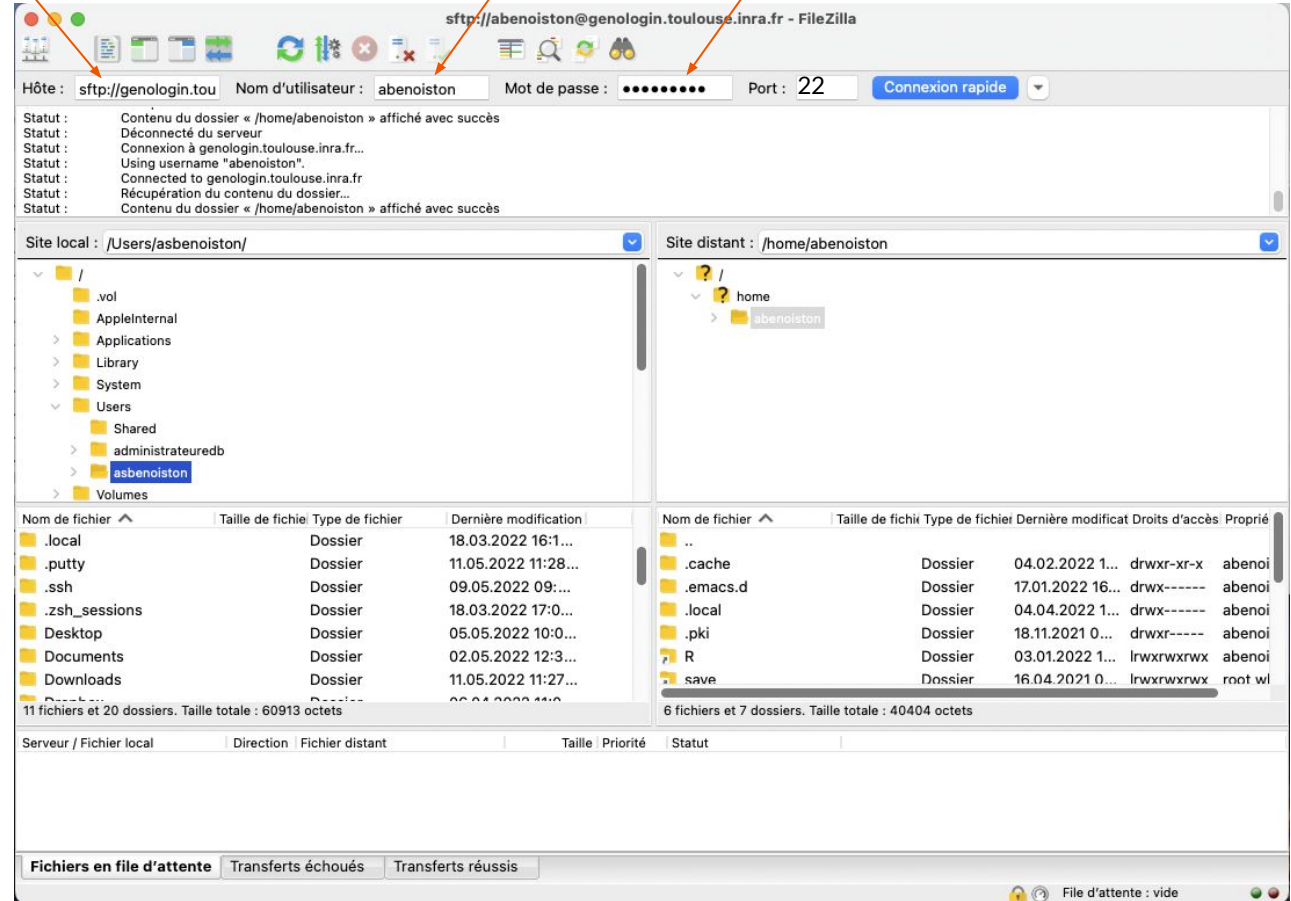
-> Enter your password

FileZilla

genologin.toulouse.inra.fr

your login

your password





Download data

For the purpose of the exercises, download some data following one of these solutions:

- Enter this address in your **web navigator**: <https://mothur.s3.us-east-2.amazonaws.com/wiki/miseqsopdata.zip>
 - + unzip the file
 - + copy the folder in your work/ directory using FileZilla
- Use **wget** in your terminal: `cd work/`
then **wget** <https://mothur.s3.us-east-2.amazonaws.com/wiki/miseqsopdata.zip>
and finally unzip the file **unzip miseqsopdata.zip**

Structure of a Unix system



Dennis Ritchie and Ken Thompson, inventors of Unix, working on a PDP-11 (a “minicomputer”!) on which the first version of Unix was ran

Introduction to Bash



What is Bash?

A **command-line interpreter** (or shell): provides a command-line interface to interact with a Unix OS

Developed in 1989 from the Bourne shell (sh)

Default shell on most Unix systems



Brian Fox, the author of Bash



Syntax of a Bash command

`command` [-options] [arguments]

Examples:

`ls -l`

`head -n 30 sequence.fasta`

`cp -r ../exercise1 exercises/`

Get help on a specific command:

`man ls`



Tree structure of files

“/” root directory

“~” home directory

“.” current/working directory

“..” parent directory

“-” last working directory

Change directory: `cd [directory name]`



Tree structure of files

“/” root directory

“~” home directory

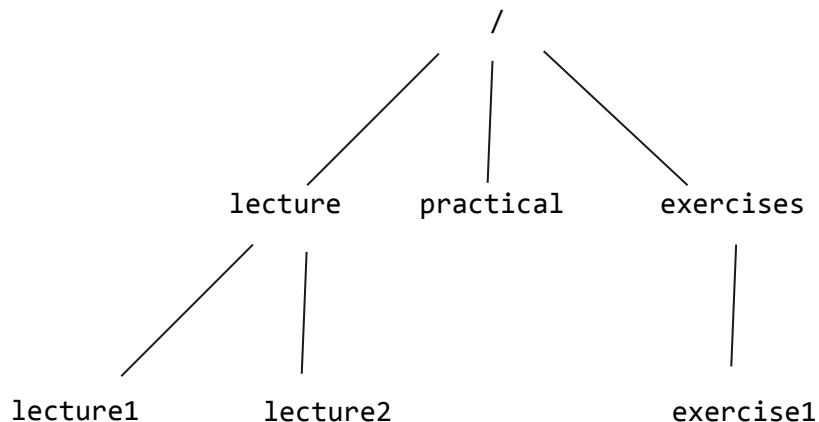
“.” current/working directory

“..” parent directory

“-” last working directory

Change directory: `cd [directory name]`

A small exercise:

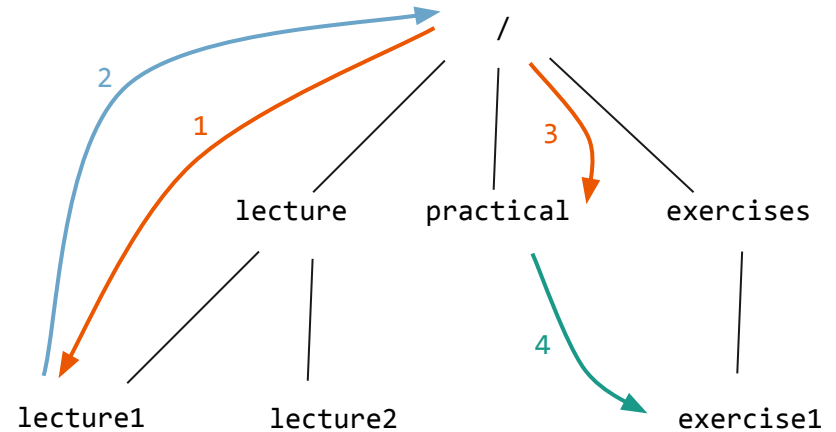


Tree structure of files

- "/" root directory
- "~" home directory
- "." current/working directory
- ".." parent directory
- "_" last working directory

Change directory: `cd [directory name]`

A small exercise:





Tree structure of files

`ls` (list directory contents)

`ls -lah`

`ls -lah work/`

`tree` (list contents of directories in a tree-like format)

`tree -dL 2`

`pwd` (print working directory)

`pwd`



Absolute vs. relative path

An absolute path begins with the **root** “/”

```
/home/abenoiston/work/MiSeq_SOP/F3D0_S188_L001_R1_001.fastq
```

A relative path starts *from where you are in the tree structure*

```
work/MiSeq_SOP /F3D0_S188_L001_R1_001.fastq
```

(path from /home/abenoiston/)



Manipulate files and directories

Create and remove files/directories

```
touch / rm [file_name]
```

```
mkdir / rmdir [dir_name]
```

Copy files/directories

```
cp [-r] [source] [destination]
```

```
cp file1 file2
```

```
cp -r dir1 dir2
```

Move/rename files/directories

```
mv [source] [destination]
```

```
mv file_name existing_dir_name
```

```
mv old_file_name new_file_name
```

```
mv old_file_name
```

```
existing_dir_name/new_file_name
```



Find files and directories

```
find [dir_name] [-options] [parameter]
```

```
find myfolder/ -name "*.fastq"
```

```
find -type d
```

```
find -type f
```



Display file content

```
cat [file_name]
```

```
more [file_name]
```

```
less [file_name]
```

```
head [file_name]
```

```
tail [file_name]
```



Piping and redirection

Appending commands together with a pipe

```
echo "hello world" | grep hello
```

Sending output of a command to a file

```
echo "hello world" > hello.txt
```

Appending output of a command to a file (no overwriting)

```
echo "hello world" >> hello.txt
```

Getting input to a command from a file

```
cat < hello.txt
```



Searching files and counting words

Finding the word “hello”

```
echo "hello world" | grep hello <- remember this?
```

Or from a file...

```
grep "hello" filename
```

Lots of options; line number (-n) ; only match (-o) ; lines before -B and after -A

Counting the number of lines/words/characters

```
wc (often with -l)
```

Tutorial 1

→ Follow the file **Morning_tuto_bash_1.sh** on
https://github.com/Imogen-D/Bases_De_Bash_Workshop

Owners and permissions



Owners and permissions

Owners:

1. The user who owns the file (u), usually the creator of the file
2. The group that owns the file (g). If a user is a member of a certain group that owns a file, the user will also have certain permissions on that file.
3. The others, the rest of the world (o). In short, everyone is neither the owner of the file, nor a member of the group that owns the file.



Owners and permissions

Permissions:

1. Reading a file (r)
2. Writing a file (w)
3. Executing a file (x)

Understand access rights

Show the permissions on the files in a directory:

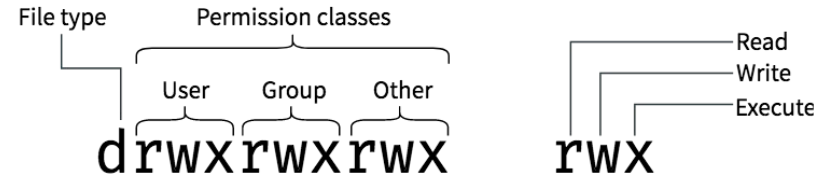
```
ls -l
```

Permissions appear like this:

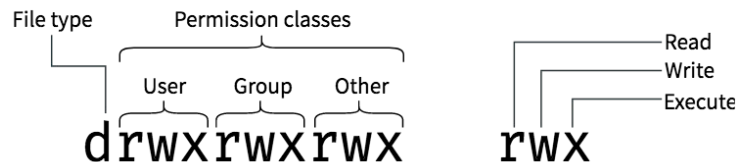
```
drwxr-xr-x
```

The first symbol indicates the nature of the file:

- d directory
- file
- l symbolic link



Change permissions



chmod (*change mode*, change permissions)

3 elements to choose:

1. To whom the change applies : u (user), g (group), o (others), a (all)
2. The change you want to make: +, -
3. The right we want to change: r, w, x

Examples :

```
chmod o-r myfile
```

```
chmod u+rwx,g+rx-w,o+r-wx myfile
```

chown (*change owner*) Example: **chown** toto fichier1

Download files



```
$ wget
```

Download files from online repositories: wget, curl

Wget: let's try with the WorldClim min temperature data

```
wget --no-check-certificate  
https://biogeo.ucdavis.edu/data/worldclim/v2.1/base/wc2.1_10m_tmin.zip
```

cURL: let's try with one of *Yersinia pestis*' reference genome

```
curl -L  
"https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/015/336/265/GCF_015336265.1_ASM1533626v1/GCF_015336265.1_ASM1533626v1_genomic.fna.gz" -o  
GCF_015336265.1_ASM1533626v1_genomic.fna.gz
```

More information: <https://linuxhint.com/what-is-the-difference-between-wget-vs-curl/>

Unzip files

Zip-Unzip files: tar, gzip, gunzip

gunzip: let's unzip the WorldClim data and the *Yersinia* genome

```
gunzip wc2.1_10m_tmin.zip
```

```
gzip GCF_015336265.1_ASM1533626v1_genomic.fna.gz
```

tar: let's compress folders of files for long term storage

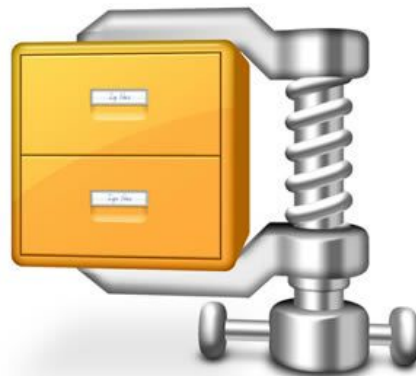
```
mkdir -p zipping ; mkdir -p zipping/zippings_{001..100}/
```

```
tar -zcvf my-archive-name.tar.gz zipping
```

```
du -chs ./*
```

How much disk space is saved by the compression?

More info: <https://lecrabeinfo.net/linux-compresser-decompresser-fichiers-dossiers-avec-tar-gzip-bzip2-xz.html>



Tutorial 2

→ Follow the file **Morning_tuto_bash_2.sh** on
https://github.com/Imogen-D/Bases_De_Bash_Workshop



Tutorial

```
cd ~/work
cd bash_slurm_training
mkdir data slurm # creates 3 folders
mv slurm slurm # oops we misspelled the folder, we rename it
cd data
wget https://mothur.s3.us-east-2.amazonaws.com/wiki/miseqsopdata.zip
unzip miseqsopdata.zip # files unzip in a folder named 'MiSeq_SOP'
cd MiSeq_SOP
ls -lh # have a look at the files - look at the structure of the output
# let's organize our data:
mkdir fastq_files
mv *.fastq fastq_files # moves all fasta files
pwd # check where we are
```



Tutorial

```
cd .. # go back to the parent folder
tree # have a look at how are organized the files in MiSeq_SOP folder
# now look at the content of the files:
head -n 20 MiSeq_SOP/fastq_files/F3D1_S189_L001_R1_001.fastq
head -n 5 MiSeq_SOP/fastq_files/*.fastq
less MiSeq_SOP/HMP MOCK.v35.fasta
# extract the first lines of this file and write them in another file:
head MiSeq_SOP/HMP MOCK.v35.fasta > head_file.txt
cat head_file.txt
echo "THE END" >> head_file.txt # print "THE END" at the end of the file
rm head_file.txt # delete the file
# change permissions for others on MiSeq_SOP:
chmod -R o-rwx MiSeq_SOP/
```

Get commands' recent HISTORY

Obtain the last command.s

use the keyboard's up arrow



Get commands' HISTORY

Print the last [thousand] commands' history

```
history
```

or the last n [25] commands

```
history 25
```

clear [-c] your commands' history (if you have something to hide)

```
history -c
```

save your commands' history to a permanent file

```
history 500 > ./the_true_history
```

more options with the manual's instructions

```
man history
```



Bas relief, cercueil, MUSEE SAINT-RAYMOND, MUSEE D'ARCHEOLOGIE DE TOULOUSE

Create links

Organizes your files and avoid duplicating them !

Soft links -s (--symbolic) option: like windows shortcut [`ln -s`], **can** point to a file or a directory on a different filesystem or partition.

```
dd if=/dev/zero of=my-2mb-file bs=100000 count=20
```

```
ln -s my-2mb-file my-2mb-file-softlink
```

 #just a light link

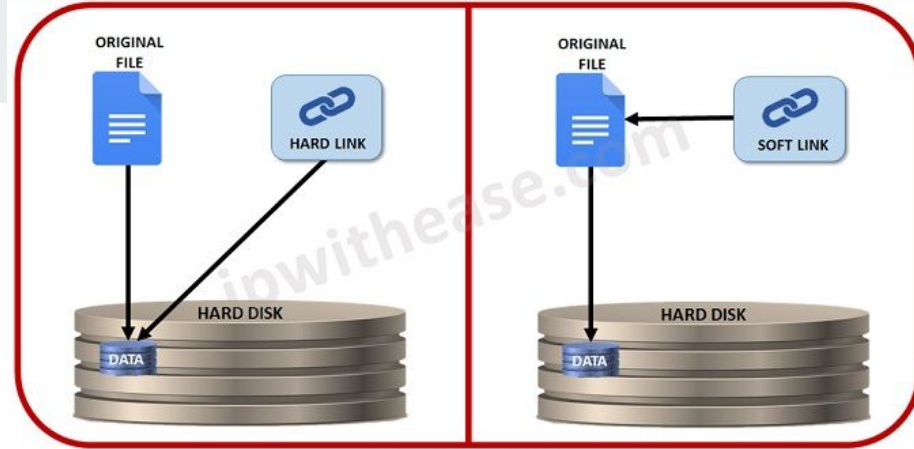
Hard links: an additional name/location for an existing file [`ln`], **not** for directories and files on a different filesystem or partition

```
ln my-2mb-file my-2mb-file-hardlink
```

 #copied the size but not using more space

```
ls -lh my-2mb-file*
```

```
du -chs ./
```



Jokers and regular expressions

Jokers, wildcards and regular-expressions

signification

joker (1 character)

joker (several characters)

character among ...

character except ...

characters ? *

bash

?

*

[...]

[!...]

\? *

list all files with extension .txt:

```
ls *.txt
```

list files for individuals from SX00 to SX09

```
touch individual_SX{00..25}.txt
```

```
ls individual_SX0?.txt
```



list SX and SY files from individual 01:

```
touch individual_SY{00..25}.txt
```

```
ls individual_S[XYZ]01.txt
```

What is that one doing ????

```
ls individual_S[XYZ][!023]1.txt
```

Regular Expressions - Quick Reference Guide



Anchors

^	start of line
\$	end of line
\b	word boundary
\B	not at word boundary
\A	start of subject
\G	first match in subject
\z	end of subject
\Z	end of subject or before newline at end

Non-printing characters

\a	alarm (BEL, hex 07)
\cx	"control-x"
\e	escape (hex 1B)
\f	formfeed (hex 0C)
\n	newline (hex 0A)
\r	carriage return (hex 0D)
\t	tab (hex 09)
\ddd	octal code ddd
\xhh	hex code hh
\x{hhh...}	hex code hhh...

Generic character types

\d	decimal digit
\D	not a decimal digit
\s	whitespace character
\S	not a whitespace char
\w	"word" character
\W	"non-word" character

POSIX character classes

alnum	letters and digits
alpha	letters
ascii	character codes 0-127
blank	space or tab only
cntrl	control characters
digit	decimal digits
graph	printing chars -space
lower	lower case letters
print	printing chars +space
punct	printing chars -alnum
space	white space
upper	upper case letters
word	"word" characters
xdigit	hexadecimal digits

Literal Characters

Letters and digits match exactly	a x B 7 0
Some special characters match exactly	@ - = %
Escape other specials with backslash	\ \ \ \$ \ [

Character Groups

Almost any character (usually not newline)	.
Lists and ranges of characters	[]
Any character except those listed	[^]

Counts (add ? for non-greedy)

0 or more ("perhaps some")	*
0 or 1 ("perhaps a")	?
1 or more ("some")	+
Between "n" and "m" of	{n,m}
Exactly "n", "n" or more	{n}, {n,}

Alternation

Either/or	
-----------	--

Lookahead and Lookbehind

Followed by	(?=)
NOT followed by	(?!)
Following	(?<=)
NOT following	(?<!=)

Grouping

For capture and counts	()
Non-capturing	(?:)
Named captures	(?<name>)

Back references

Numbered	\n \gn \g{n}
Relative	\g{-n}
Named	\k<name>

Character group contents

x	individual chars
x-y	character range
[:class:]	posix char class
[^:class:]	negated class

Examples

[a-zA-Z0-9_]
[[:alnum:]]

Comments

(?#comment)

Conditional subpatterns

(?(condition)yes-pattern)
(?(condition)yes no-pattern)

Recursive patterns

(?n)	Numbered
(?0) (?R)	Entire regex
(?&name)	Named

Replacements

\$n	reference capture
-----	-------------------

Case foldings

\u	upper case next char
\U	upper case following
\l	lower case next char
\L	lower case following
\E	end case folding

Conditional insertions

(?n:insertion)
(?n:insertion:otherwise)

Regular-expressions

Regex are very useful for complex requests

start of a line

alphanumeric character string

other characters allowed

character string min,max length

character .

end of a line

^

[A-Z0-9]

[._%+]

{2,}

\.

\$

Get properly formatted emails from a file

```
echo my-professional-email@bushpig-hunter.wwf > my_emails_file.txt  
grep -i -E '^[A-Z0-9._%+]+@[A-Z0-9.-]+\.[A-Z]{2,}$' *.txt
```

More information

<https://www.regular-expressions.info/>



Tutorial 3

→ Follow the file **Morning_tuto_bash_3.sh** on
https://github.com/Imogen-D/Bases_De_Bash_Workshop

Text editing



Text Editor Commands

Two major commands:

Sed

- Syntax from an original text editor 'ed' ; **Stream editor**
- 1973-1974 by Lee E. McMahon of Bell Labs
- In-line text file editor - primarily for strings in files
- Using regular expression (**regex**)
- Substitution

Awk

Common Regular Expressions

Character	Kind	Explanation
.	Point	any character
[...]	hooks	character class : all the characters listed in the class, with the possibility of ranges whose limits are separated by "-". Ex: [0-9a-z] for all lowercase alphanumeric, or [0-Z] for all characters in the Unicode table between "0" and "Z", i.e. uppercase alphanumeric plus ".,<=>?@" [2].
[^...]	brackets and circumflex	complemented class : all characters except those listed.
^	circumflex	marks the start of the string or line.
\$	dollar	marks the end of the string or line.
	vertical bar	alternative - or recognizes one or the other
(...)	parentheses	capturing group : used to limit the scope of a mask or alternative
*	asterisk	0, 1 or more occurrences
+	more	1 or more occurrences
?	questioning	0 or 1 occurrence
{...}	braces	count: determines a number of characters fulfilling the criteria it follows. Ex: a{2} two occurrences of "a", a{1,10} (without spaces) between one and ten.
(? P<nom>pattern)	Named subpattern	names the result of a capturing group by name.

Sed Syntax

- Most common: substitution

```
sed [options] 's/REGEX/REPLACEMENT/[FLAGS]' file
```

```
sed [options] 's/foo/bar/[FLAGS]' file
```

Flags: g = global, [1:9] = number match, p = print

- Removal of lines

```
sed [options] '1,3d'
```

- Printing lines

```
sed -n 5,30p file
```

Options: -i, -n (often in combination with p) -> and more

Try this -

```
echo 'abc' | sed 's/abc/xyz/'
```

And what is this one doing?

```
echo 'abc' | sed 's/./xyz/g'
```




Text Editor Commands

Two major commands:

Sed

Awk

- More applications ; Turing-Complete
- 1977 at Bell Laboratories
- Named by: Alfred Aho, Peter Weinberger, and Brian Kernighan
- Often used on files with structure i.e. columns





Awk Syntax

Uses 'records' and 'fields' - Great for tabulated data - Default field separators is any whitespace

```
awk '
    BEGIN { actions }

    /pattern/ { actions }

    .....

    END { actions }
' filenames
```



Awk Syntax

Print command

`awk print $1, $3` shows first and third fields

Variables: NF;FS;OFS

Example

```
fish|Eating|2
frog|Eating|10
crocodile|Sleeping|100
fish|Sleeping|200
```

```
awk -F '|' '$1 ~ /fish/ {sum += $3} END {print sum}' inputfilename
```

```
awk -F '|' '$1 ~ /^fish$/ {sum += $3} END {print sum}' inputfilename
```

```
awk -F '|' '$1 == "fish" {sum += $3} END {print sum}' inputfilename
```



Text Editor Commands

One more command: split

```
split [options] filename prefix ; -l or -b ; -d ; --additional-suffix=.xml
```

lines, bytes, numeric

Text Editor Practise I



```
mkdir exercise2 ; cd exercise2      #making directory and file
echo 1 2 1 > exercise2.txt
echo 1 2 3 >> exercise2.txt
head exercise2.txt                   #looking at file
```

```
sed -i 's/1/4/2' exercise2.txt      #changing the second instance of "1" to "4"
```

```
awk -F" " '{print $0, "the sum is "$1+$2+$3}' exercise2.txt > exercise2_sum.txt
#summing and creating new column
```

```
split -l 1 -d --additional-suffix .txt exercise2_sum.txt exercise2_ ; ls
#splitting the output file into separate numbered files
```

Conditional structures and loops

UNIX if statements

If statements only perform a task if a condition is true

Followed by **then** and/or **else** and/or **elif**

```
if [ $1 > 18 ]
then
    echo You may go to the party.
elif [ $2 == 'yes' ]
then
    echo You may go to the party but be back before midnight.
else
    echo You may not go to the party.
fi
```

Boolean Operators

```
if [ $USER == 'bob' ] || [ $USER == 'andy' ] #OR
then
    ls -alh
else
    ls
fi
```

```
if [ -r $1 ] && [ -s $1 ] #AND
then
    echo This file is useful.
fi
```

UNIX loops

While and For loops (+ until or select)

Followed by **do** and **done**

```
while read line
do
    echo $line
done < file.txt
```

Step Value

```
for i in 1 2 3 4 5
do
    echo "Welcome $i times"
done

for i in {0..10..2}
do
    echo "Welcome $i times"
done
```


Text Editor Practise II

```
wget https://raw.githubusercontent.com/Imogen-D/Bases_De_Bash_Workshop/main/exercise2script.awk
```

```
#!/bin/bash
for file in $@; do
    if [ -f $file ] ; then
        echo "File is: $file"          #print out filename
                                     #print the total number of times the number 1 appears in the file
        awk ' BEGIN { print "The number of times the number 1 appears in the file is:" ; }
              /^1/ { counter+=1 ; }
              END { printf "%s\n", counter ; }
              ' $file
    else
        #print error info incase input is not a file
        echo "$file is not a file, please specify a file." >&2 && exit 1
    fi
done
exit 0                                #terminate script with exit code 0 in case of successful execution
```

```
chmod +x exercise2script.awk; ./exercise2script.awk exercise2_02.txt
```

Text Editor Practice II

genologin.toulouse.inra.fr (idumville)

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Quick connect...

/work/idumville/exercise2/

Name	Size (KB)
exercise2script.awk	0

exercise2script.awk 0 Bytes

2022-05-12 09:31 idumville UMR5174 -rw-r--r--

Support

=====
Pour plus d'informations, consulter le site web :
<http://bioinfo.genotoul.fr/>

Pour toute demande de support, adressez-vous à :
support.bioinfo.genotoul@inrae.fr

2022-05-12 09:31 idumville UMR5174 -rw-r--r--

Informations sur le compte idumville (12-08-2022-2023-03-05)

=====
Home: 3.05M (0%)
Work: 584G (57%)
Save: 0.00 (0%)
CPU : 70241h (70%)
Expiration du compte: 12-08-2022
Expiration du mot de passe: 14-03-2023

Toute utilisation de notre infrastructure est soumise à notre charte: <http://bioinfo.genotoul.fr/wp-content/uploads/ChartPFBioinfoGenoToul.pdf>

Tout traitement lancé sur les serveurs genologin sera immédiatement interrompu par les administrateurs système.

```
idumville@genologin2 ~ $ cd /work/idumville/  
idumville@genologin2 /work/idumville $ ls  
pangolins  
idumville@genologin2 /work/idumville $ mkdir exercise2  
idumville@genologin2 /work/idumville $ touch exercise2.awk  
idumville@genologin2 /work/idumville $ rm exercise2.awk  
idumville@genologin2 /work/idumville $ cd exercise2/  
idumville@genologin2 /work/idumville/exercise2 $ touch exercise2script.awk  
idumville@genologin2 /work/idumville/exercise2 $
```

Remote monitoring

☐ Follow terminal folder

`chmod +x exercise2script.awk; ./exercise2script.awk exercise2_02.txt`

Bonus



Aliases

To set a shortcut command for a long command!

e.g. instead of typing `ls -lah`

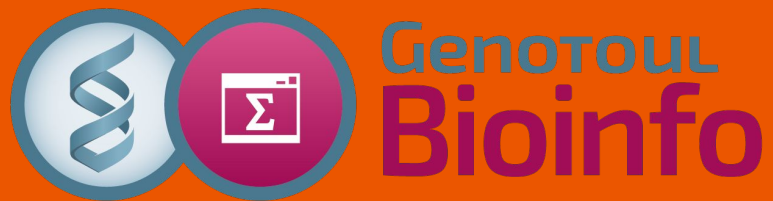
create an alias:

```
alias lh="ls -lah"
```

in your `.bash_profile` or `.bashrc` file (typically in your home directory `~`)

and finally reload the file using:

```
source ~/.bash_profile
```

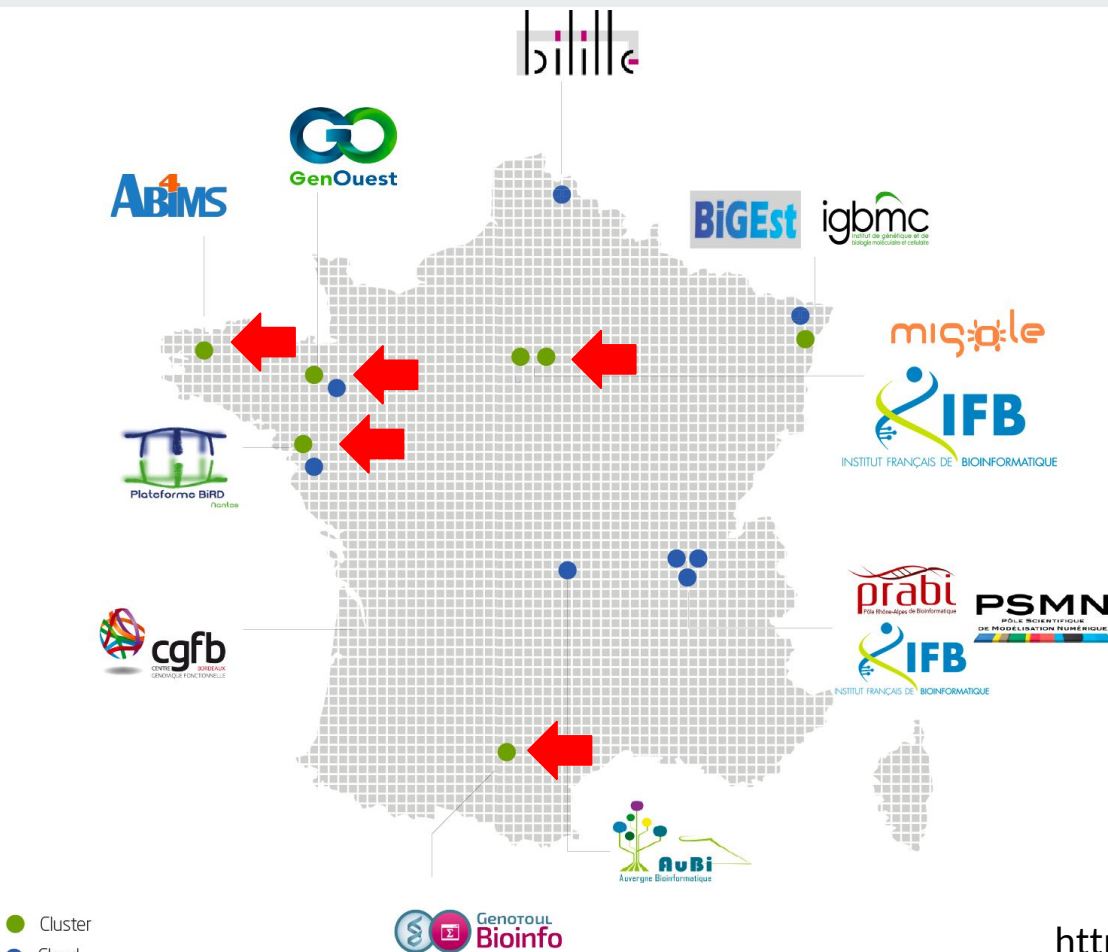


+



—

French Institute of Bioinformatics (IFB)



National infrastructure providing support, services, training

IT infrastructure

Cloud + Cluster

IFB-core Cluster since November 2018

4,300 cores, 2PB of storage

GenoToul since 2000

3,000 cores, >6PB of storage



INSTITUT FRANÇAIS DE BIOINFORMATIQUE

<https://www.france-bioinformatique.fr/en/home/>62

The Genotoul cluster

The Genotoul facility

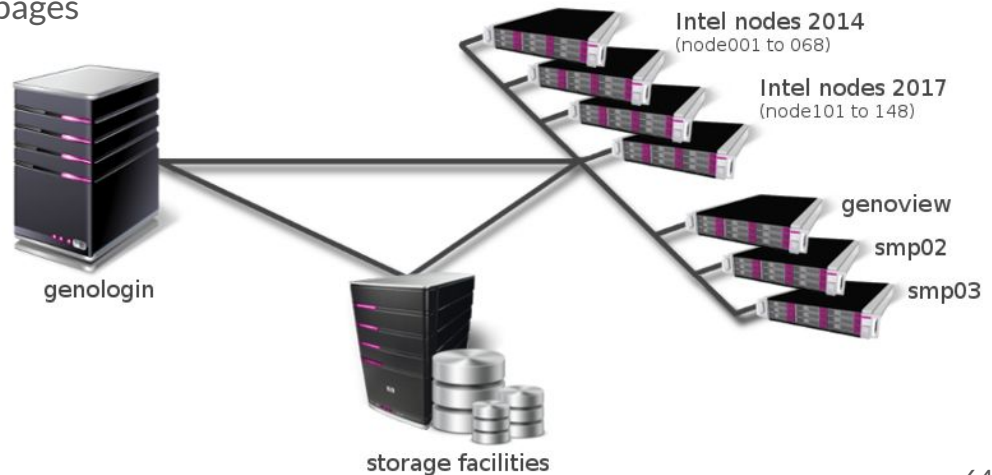
The closest facility geographically

A well maintained website with furnished FAQ pages

A reactive support team

Most genomic tools already installed

Most used facility in EDB

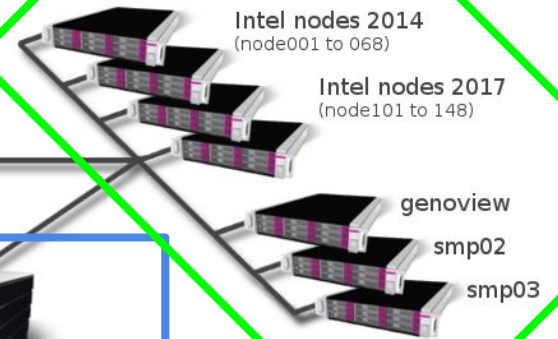
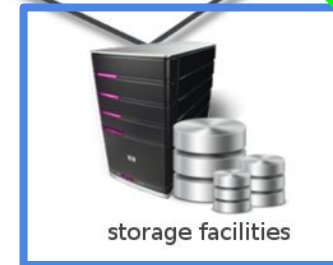


The Genotoul facility's structure

/home	100Mb, landing area + user-profile files
/work	1Tb temp. storage - working area
/save	250Gb back-up storage
/bank	biological databases
/usr/local/bioinfo/src	software modules
119 nodes with 3072 cores	to run your jobs



JOBS



The Genotoul facility's rights and use

/home

do not use except for profile-files

/work

jobs can write here

/save

jobs cannot write here

/bank

read only

/usr/local/bioinfo/src

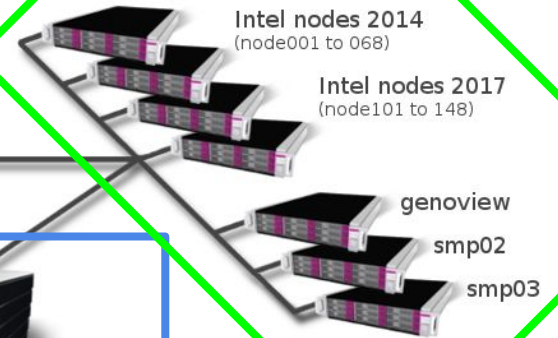
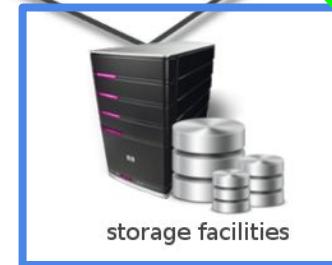
read only

119 nodes

to run your jobs only



JOBS





Assess used space /computational time

Reminder when opening an ssh connection

Assess used space (**du** =disk usage)

```
du -csh /DIR_NAME/USER_NAME/*
```

Assess used computational time

100,000 hours / user / year

```
squota_cpu
```

Need more time? File a request at:

<http://bioinfo.genotoul.fr/index.php/ask-for/resources-request/>

```
=====
Informations sur le compte jsalmona (10-05-2022 00:05)
=====
Home: 76.88M (1%)
Work: 706G (68%)
Save: 214.17G (85%)
CPU : 21983h (22%)
Expiration du compte: 31-08-2022
Expiration du mot de passe: 02-03-2023
=====
```



Find modules / softwares / packages

All soft are listed there: <http://bioinfo.genotoul.fr/index.php/resources-2/software/>

Also consult the gt FAQ webpage: http://bioinfo.genotoul.fr/index.php/faq/software_faq/

```
search_module soft_name #(or part of soft_name)
```

```
ls /usr/local/bioinfo/src | grep -i soft_name
```

```
module avail : display all available software installed on the cluster
```

```
module avail category/soft_name : display available versions for a specific application (with category in bioinfo,compiler,mpi or system) (case sensitive)
```

```
module avail -t 2>&1 | grep -i soft_name : display available versions for a specific application
```



Load - Unload modules / packages / softwares

Consult the gt FAQ webpage: http://bioinfo.genotoul.fr/index.php/fag/software_fag/

```
module load bioinfo/blast-2.2.26 # load a module
```

```
which blastall # verify the loaded module
```

```
# /usr/local/bioinfo/src/NCBI_Blast/blast-2.2.26/bin/blastall
```

```
module unload modulename# : unload or remove a module
```

```
module purge # unload or remove all modules
```

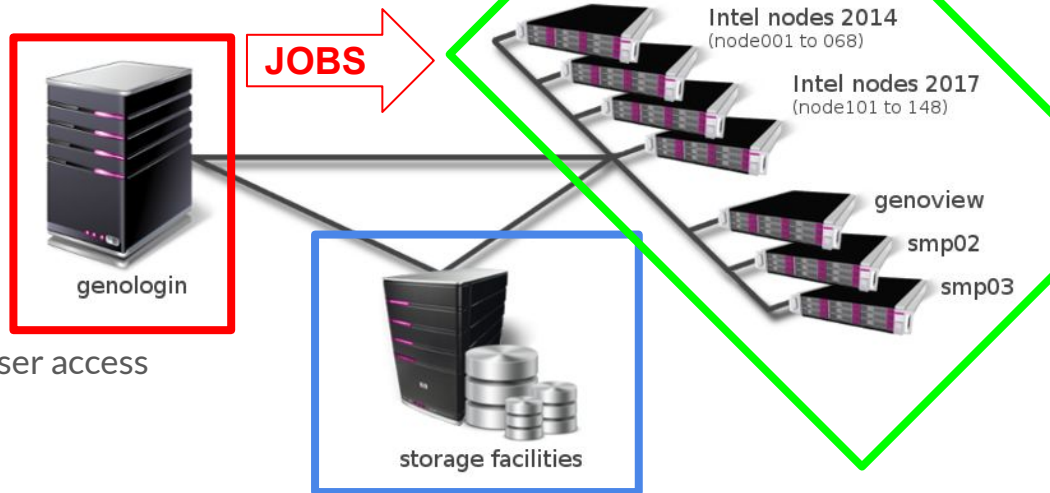
```
module help modulename # path to the very useful "How_to_use_SLURM_soft_name" file
```

Partitions

Partitions can be considered as **job queues**

With differing job size or time limit, or user access

On genotoul: **workq** is default



Queue	Access	Priority	Max time	Max slots
workq	everyone	100	4 days (96h)	3072
unlimitq	everyone	1	180 days	500
interq (runVisuSession.sh)	on demand		2 days (48h)	32
smpq	on demand		180 days	96
wflowq	specific software		180 days	3072

Introduction to SLURM



What is SLURM?

SLURM (Simple Linux Utility for Resource Management): software used to manage resources on the Genotoul cluster.

Book resources and launch the programs on the cluster nodes.

All commands begin with the letter **s**.

<https://slurm.schedmd.com/documentation.html>





SLURM vocabulary

CPU: the smallest unit of a computer processor (= a thread or a hyperthreaded core)

RAM: memory used by the processor to store the analyzed data

Task: the execution of a tool, may use several CPUs

Job: a resource reservation for a process

Job step: a "job step" is the part of a job that consists in executing a program

Task: a job step can use several "tasks" (if it can be parallelized)

Submit a job with SLURM



SBATCH & SRUN

Two main commands to submit a job on the cluster with SLURM:

- **sbatch** is used to book resources and submit a job script for later execution
- **srun** is used to submit a job for execution or initiate job steps in real time



sbatch & srun: the main options

- -cpus-per-task: advise the Slurm controller that ensuing job steps will require *ncpus* number of processors per task (default: 1)
- -mem: specify the real memory required per node (default: 2G)
- -mem-per-cpu: minimum memory required per allocated CPU
- -ntasks: number of tasks (default: 1)



Monitor a job

squeue: view information about jobs located in the Slurm scheduling queue

```
squeue -u $USER # permet de voir les jobs de votre utilisateur
```

```
squeue -u $USER -t RUNNING # permet de voir les jobs en cours d'exécution
```

```
squeue -u $USER -t PENDING # permet de voir les jobs en attente
```

```
squeue -j <jobid> # permet de voir l'état d'un job à partir de son job id
```



Monitor a job

`sacct` displays accounting data for all jobs and job steps in the Slurm job accounting log or Slurm database

```
sacct -j <job_id> # consulter les informations de bases du job job_id
```

```
sacct --start=2021-03-01 # consulter les informations de l'ensemble de ses jobs depuis le 1 mars 2021
```

```
sacct --format=JobID,JobName%30,State,Start,Elapsed,NodeList -j <job_id> # afficher des informations détaillées pour le job job_id
```



Killing your jobs

The `scancel` command:

Kill the specified job

```
scancel job_id
```

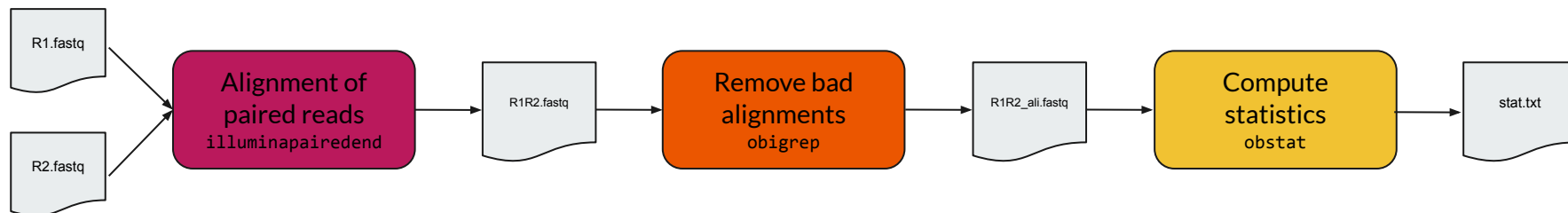
Kill all job launched by the specified user

```
scancel -u username
```

Align reads from metabarcoding data

We will use programs from the obitools (set of programs specifically designed for analyzing NGS data in a DNA metabarcoding context):

1. **illuminapairedend**: aligns the two reads of a paired-end library sequenced using an Illumina platform
2. **obigrep**: selects a subset of sequence records from a sequence file
3. **obistat**: computes basic statistics for attribute values of sequence records





Align reads from metabarcoding data

```
cd ~/work/bash_slurm_training/slurm
```

```
mkdir metabarcoding
```

```
cd metabarcoding
```

```
mkdir analysis results # organize the folder
```

```
cd analysis
```

```
wget https://raw.githubusercontent.com/Imogen-D/Bases\_De\_Bash\_Workshop/main/align\_reads\_basic.sh
```

```
wget https://raw.githubusercontent.com/Imogen-D/Bases\_De\_Bash\_Workshop/main/align\_reads\_intermediate.sh
```

```
wget https://raw.githubusercontent.com/Imogen-D/Bases\_De\_Bash\_Workshop/main/align\_reads\_difficult.sh
```

```
cd ..
```



Align reads from metabarcoding data - basic

align_reads_basic.sh

```
#!/bin/bash

module load bioinfo/obitools-v1.2.11

# align reads that have a minimum alignment score of 40 + concatenate other reads:
srun illuminapairedend --score-min=40 -r ../../data/MiSeq_SOP/fastq_files/F3D0_S188_L001_R1_001.fastq
../../data/MiSeq_SOP/fastq_files/F3D0_S188_L001_R2_001.fastq > results/F3D0_S188_L001_R1R2_001.fastq
# remove unaligned reads:
srun obigrep -p 'mode!="joined"' results/F3D0_S188_L001_R1R2_001.fastq > results/F3D0_S188_L001_R1R2_001.ali.fastq
# compute stats:
srun obistat -c score results/F3D0_S188_L001_R1R2_001.ali.fastq > results/F3D0_S188_stats.txt
```

Terminal:

```
$ sbatch --cpus-per-task=1 --mem=2M analysis/align_reads_basic.sh
```



Align reads from metabarcoding data - intermediate

align_reads_intermediate.sh

```
$ sbatch analysis/align_reads_intermediate.sh
```

```
#!/bin/bash
#SBATCH -p workq
#SBATCH --mem=2M
#SBATCH --cpus-per-task=1
#SBATCH -o align_reads_output_%j.out
#SBATCH -e align_reads_error_%j.err
#SBATCH --mail-user=anne-sophie.benoiston@ird.fr
#SBATCH --mail-type=BEGIN,END,FAIL

module load bioinfo/obitools-v1.2.11

fastq_R1="../../data/MiSeq_SOP/fastq_files/F3D141_S207_L001_R1_001.fastq"
fastq_R2="../../data/MiSeq_SOP/fastq_files/F3D141_S207_L001_R2_001.fastq"
ipe_res="results/F3D141_S207_L001_R1R2_001.fastq"
grep_res="results/F3D141_S207_L001_R1R2_001.ali.fastq"
stat="results/F3D0_S188_L001_stat.txt"

srun -J "alignment" illuminapairedend --score-min=40 -r "${fastq_R1}" "${fastq_R2}" > "${ipe_res}"
srun -J "obigrep" obigrep -p 'mode!="joined"' "${ipe_res}" > "${grep_res}"
srun -J "stat" obistat -c score "${grep_res}" > "${stat}"
```

Parallelize



Submitting jobs as an array

Lots of samples or files? -> Submit a job array, multiple jobs to be executed with identical parameters.

Multiple values be specified using a comma separated list and/or a range of values with a « - » separator.

Example :

```
sbatch --array=1-10
```

```
--array=0,6,16-32
```

```
--array=0-16:4 #a step of 4
```

```
--array=1-10%2 #a maximum of 2 simultaneously running task
```



Submitting jobs as an array

Lots of samples or files? -> Submit a job array, multiple jobs to be executed with identical parameters.

Multiple values specified using a comma separated list and/or a range of values with a « - » separator.

Example :

```
$ sbatch --array=1-1000  
#SBATCH --array=1-1000  
  
LINE=$(sed -n "$SLURM_ARRAY_TASK_ID"p File.txt)  
echo $LINE  
  
call-program-name-here $LINE
```

https://help.rc.ufl.edu/doc/SLURM_Job_Arrays



Align reads from metabarcoding data - difficult

The idea is to align all fastq files at the same time.

We will take advantage of SLURM arrays to do that :)



Align reads from metabarcoding data - difficult

We have fastq files for 20 samples in `~/work/bash_slurm_training/data/MiSeq_SOP/fastq_files`

So we have to create an array with 6 jobs

Align reads from metabarcoding data - difficult

align_reads_difficult.sh

```
#!/bin/bash
#SBATCH -p workq
#SBATCH --mem=2M
#SBATCH --array=0-19
#SBATCH -o align_reads_output_%j.out
#SBATCH -e align_reads_error_%j.err
#SBATCH --mail-type=BEGIN,END,FAIL

module load bioinfo/obitools-v1.2.11

data_folder="../../data/MiSeq_SOP/fastq_files"
results_folder="results"
fastq_R1=("${data_folder}"/*_R1_001.fastq)
fastq_R1_id=$(basename -s .fastq "${fastq_R1[$SLURM_ARRAY_TASK_ID]}")
fastq_R2=("${data_folder}"/*_R2_001.fastq)
fastq_R2_id=$(basename -s .fastq "${fastq_R2[$SLURM_ARRAY_TASK_ID]}")
ipe_res=$(echo "$fastq_R1_id.fastq" | sed 's/R1/R1R2/g')
grep_res=$(echo $ipe_res | sed 's/./fastq/.ali.fastq/g')
obi_res=$(echo "$fastq_R1_id.fastq" | sed 's/R1_001.fastq/stat.txt/g')

srun -J "${fastq_R1_id} alignment" illuminapairedend --score-min=40 -r "$data_folder"/"$fastq_R1_id.fastq" "$data_folder"/"$fastq_R1_id.fastq" >
"$results_folder"/"${ipe_res}"
srun -J "${fastq_R1_id} obigrep" obigrep -p 'mode!=joined' "$results_folder"/"${ipe_res}" > "$results_folder"/"${grep_res}"
srun -J "${fastq_R1_id} obistat" obistat -c score "$results_folder"/"${grep_res}" > "$results_folder"/"${obi_res}"
```

\$ sbatch analysis/align_reads_difficult.sh

Use R on the cluster



Use R in an interactive mode

Check-out the tutorial: http://www.nathalieviaianeix.eu/doc/pdf/tutoR_cluster.pdf

1. get connected to a node

```
srun --pty # do not run R on the master
```

2. load the R module of the version of your choice

```
module load system/R-3.5.1 #(for the last R version)
```

3. launch R

```
R
```



Use R in a submitted job

Check-out the tutorial: http://www.nathalieviaianeix.eu/doc/pdf/tutoR_cluster.pdf

1. load the R module of the version of your choice

```
module load system/R-3.5.1
```

2. launch R script

```
R my_script.R my_parameter1 my_parameter2
```



Find R packages

```
search_R_package package_name
```



Useful resources

Linux/Bash:

- [Linux - genotoul-bioinfo](#)
- <https://doc.ubuntu-fr.org/accueil>

Sed:

- https://www.gnu.org/software/sed/manual/html_node/Command_002dLine-Options.html
- <https://www.grymoire.com/Unix/Sed.html>

Awk:

- https://www.funtoo.org/Awk_by_Example_Part_1
- <https://www.grymoire.com/Unix/Awk.html>
-

Using a cluster:

- Cluster Genotoul: training, FAQ, ask for resources... <http://bioinfo.genotoul.fr/>
- [Cluster - genotoul-bioinfo](#)
- [Utilisation d'un cluster de calcul](#) (IFB)
- <https://www.cism.ucl.ac.be/Services/Formations/slurm/2016/slurm.pdf>