



ABIMS⁴

Cluster Initiation

Cycle de formation 2019

Gildas Le Corguillé

Crédit: Alexandre Cormier, Camille Vacquié, Gildas Le Corguillé



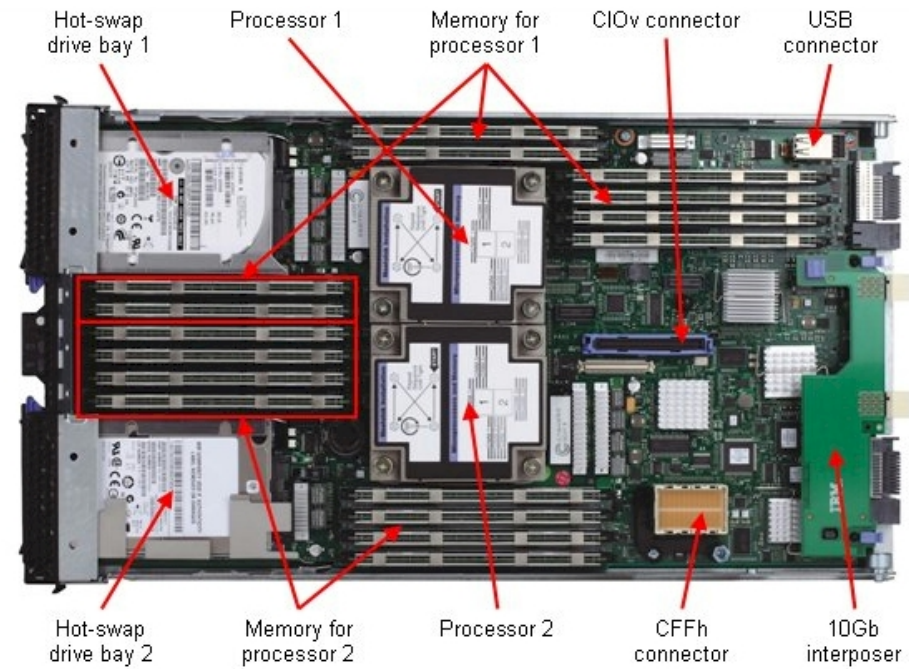
<http://abims.sb-roscoff.fr/resources/cluster/howto>

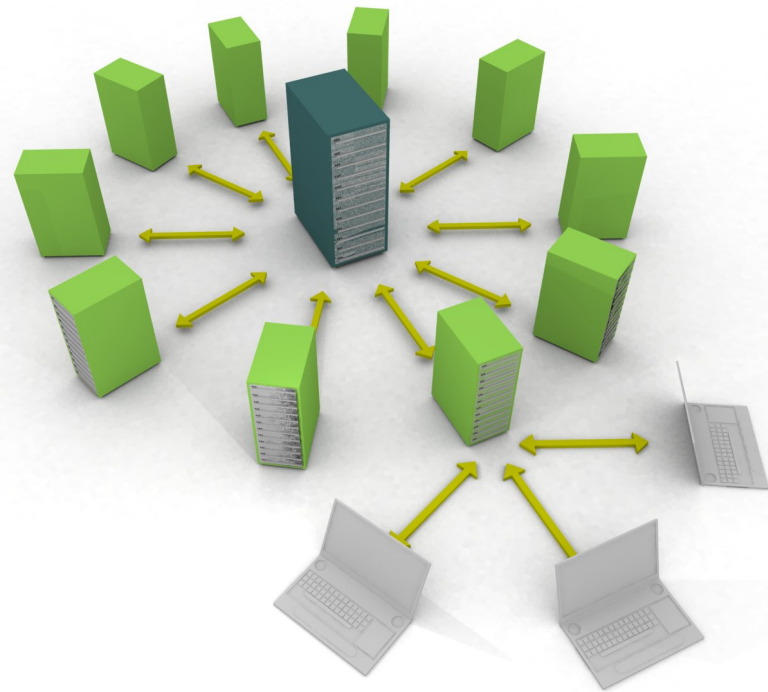
Principle

- Aggregation of computers / machines
 - Machine = node
- Distributed computing + shared access
- Transparent management for users
- Community system → rules!



Hardware





DISTRIBUTED COMPUTING

Distributed computing

- Distribution
 - Make a job as atomic as possible
 - Simple and robust
 - Linear gain
- Generate independent tasks
 - Split the data
 - Change parameters

Multithreading

- Thread

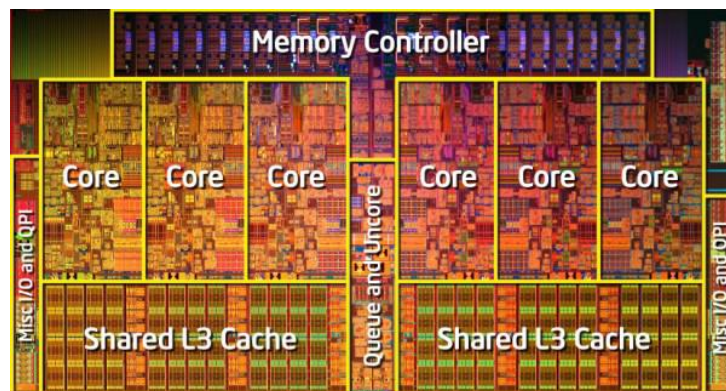
- Tasks running on the same machine but on several CPU or core
- Shared memory
- Nonlinear gain
 - For Blast, 4 CPU max



- Ex: Bowtie2, CLC, Blast,...

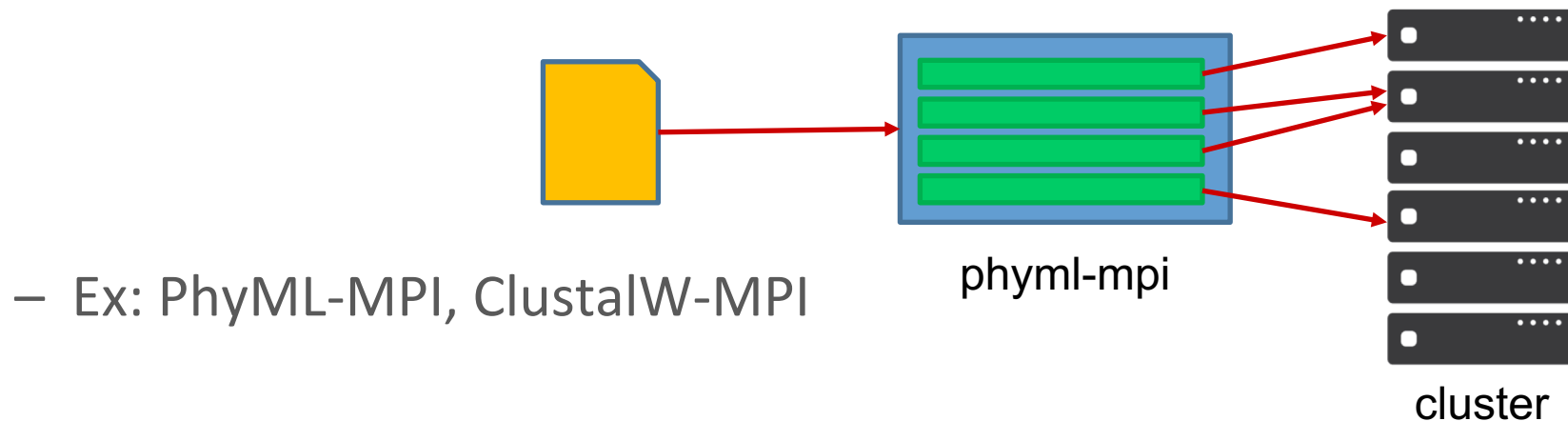
bowtie2

cluster



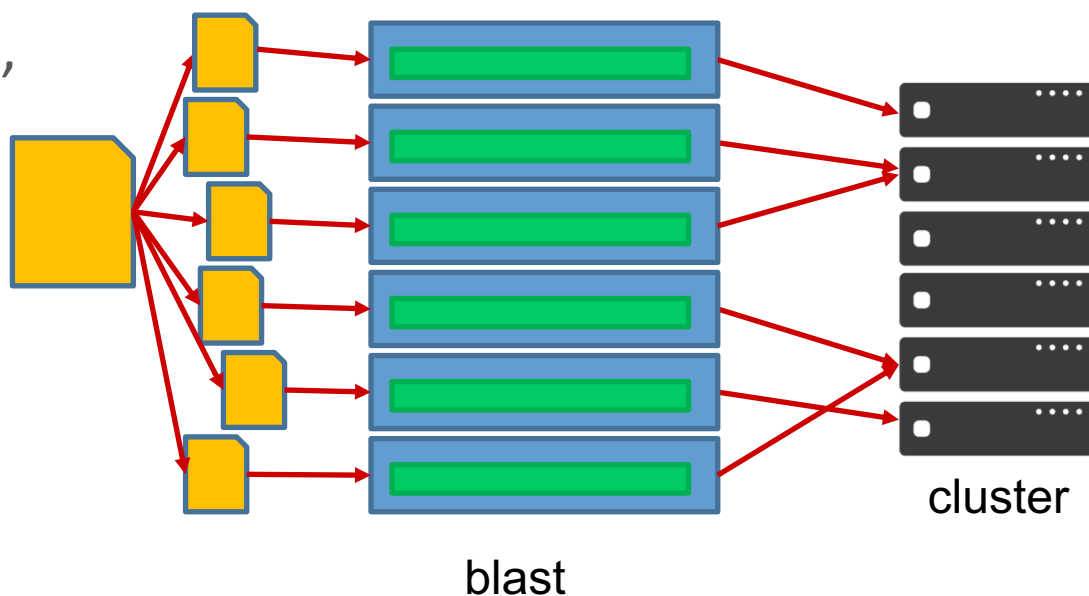
Parallel computing

- MPI (Message Parsing Interface)
 - Tasks are running on different machines
 - Communication between tasks over the network
 - Variable gain. Nonlinear in general

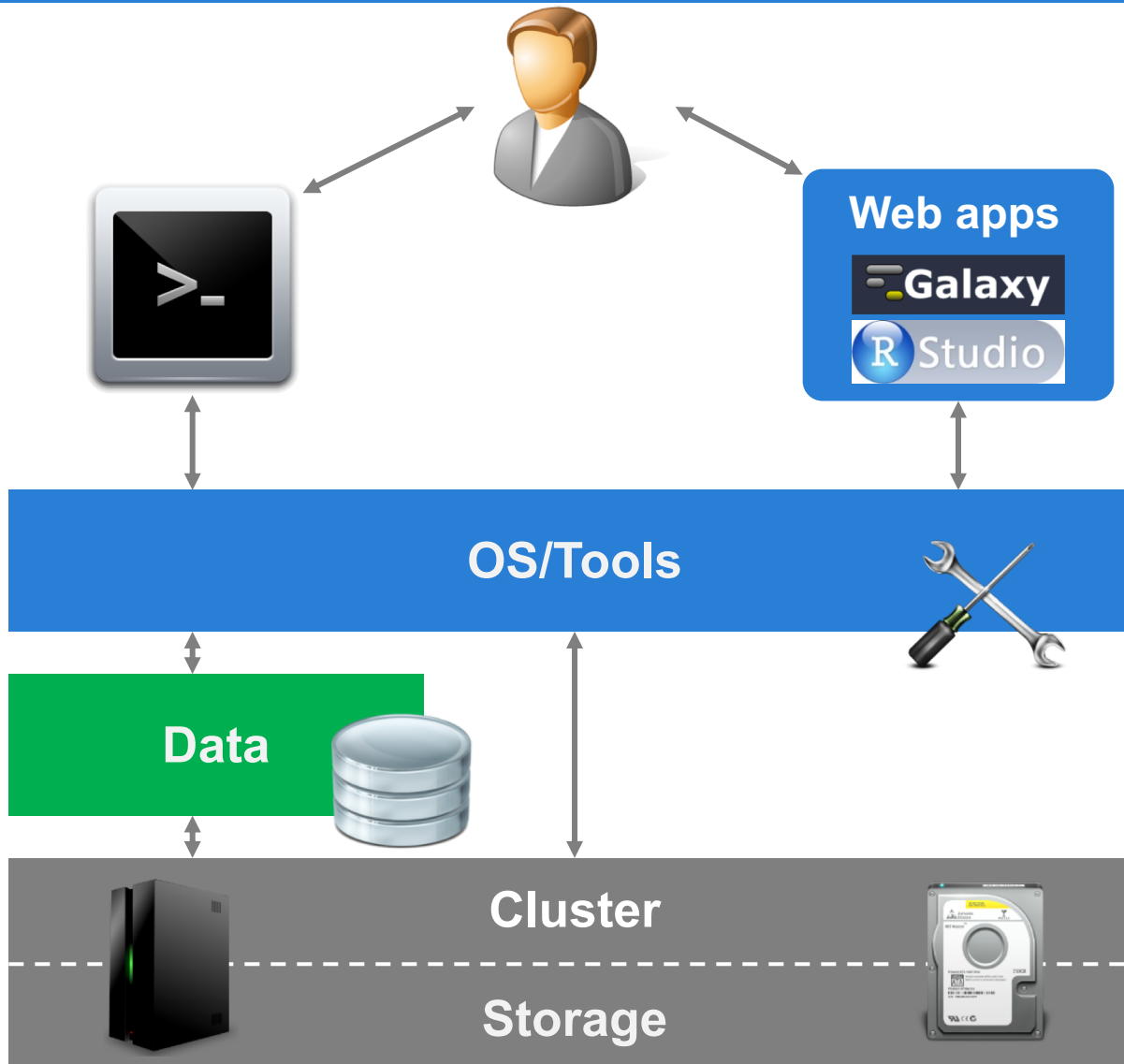


Multithreading

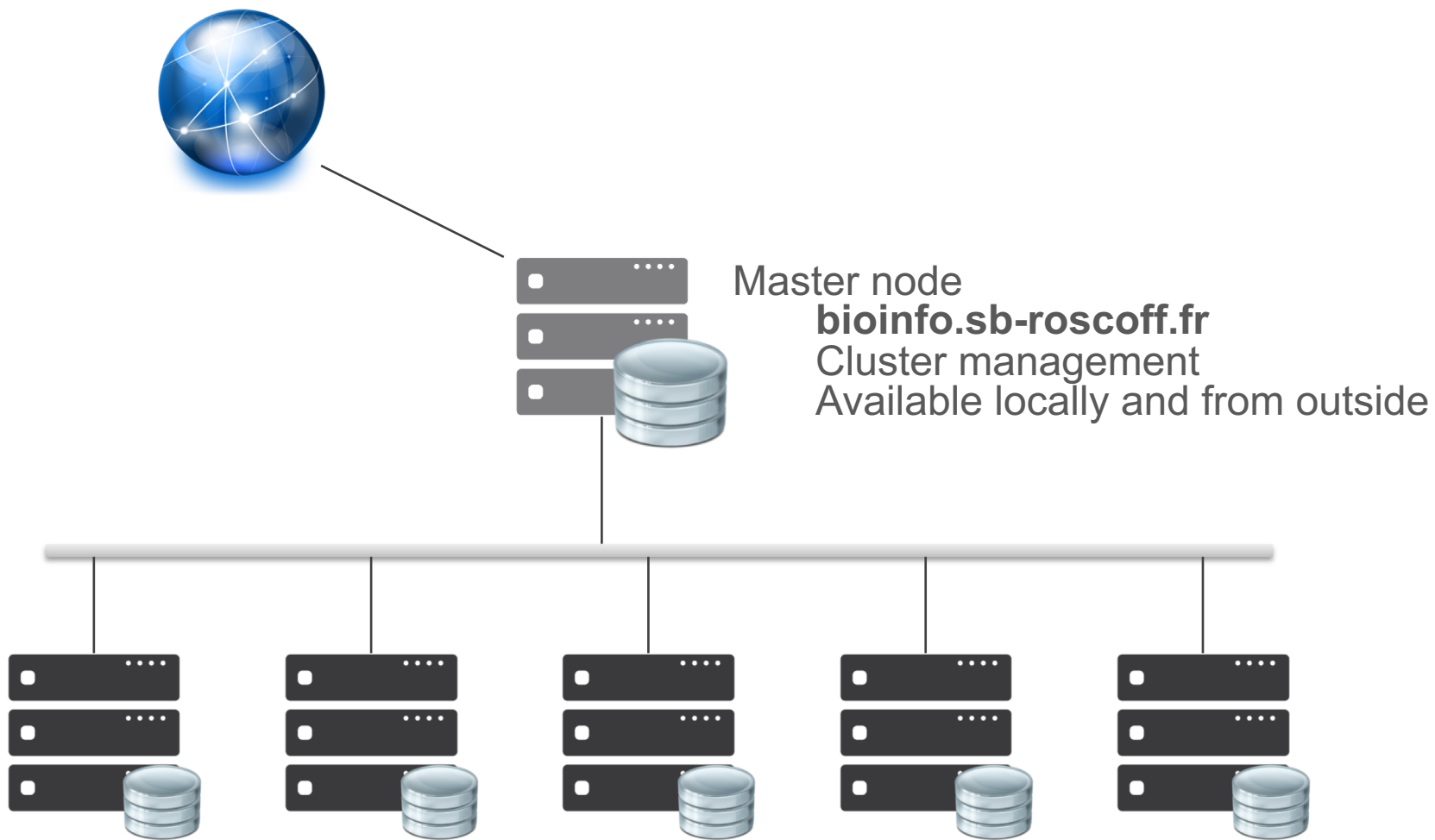
- Splited file -> Job array
 - 1 input file -> n sub-files
 - Each sub-file can be process independently
 - Linear gain
 - Ex: Blast, Diamond, Interproscan ...



Environment



Cluster



Cluster

DIFFERENTS TYPES OF NODE

Group	Node	Type	Function	Processor	RAM
@@bigcpu	n40-n55	Dell blade M630	Multithreaded traitements	Intel 32 x 2.4 GHz	128 Go
	n60-n75	Dell blade M605	Reserved for some applications	AMD 8 x 2.4 GHz	32 Go
@@bignode	n76-n79	Dell R815	Multithreaded traitements Memory usage	AMD 48 x 2.2 GHz	256 Go
	n56-n59		Multithreaded traitements Memory usage	AMD 48 x 2.2 GHz	256 Go
@@bigcpu	n80-n95	Dell C6220	Multithreaded traitements	Intel 32 x 2.2 GHz	128 Go
@@bigmem	n99	Dell R910	Memory usage	Intel 40 x 2.0 GHz	1 To



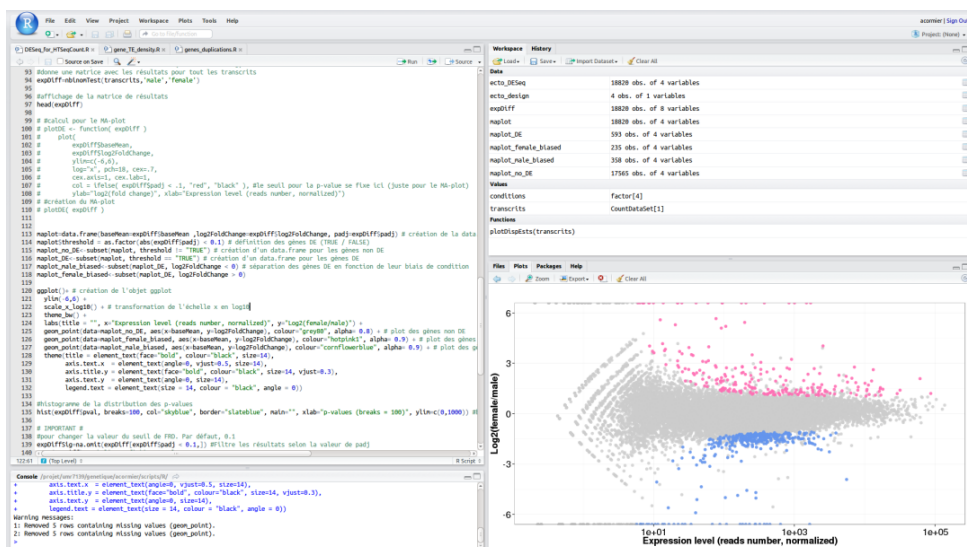
- Command-line
 - Knowledge in Unix / Bash
 - Integrated in computer resources and storage



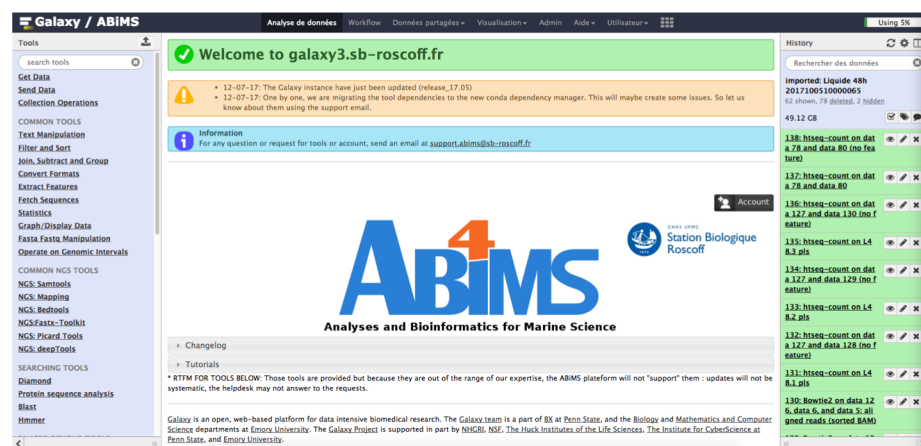
```
acormier@n0:/tmp
drwx----- 2 root      root      16384 avr 15 2009 lost+found
srwxr-xr-x  1 caron     sib       0 oct 30 2012 mapping-caron
srwxr-xr-x  1 corre     sib       0 avr 18 14:08 mapping-corre
srwxr-xr-x  1 ewcorre    lbm       0 jun 25 15:37 mapping-ewcorre
srwxr-xr-x  1 jkervellec sib      0 fév 22 13:52 mapping-jkervellec
srwxr-xr-x  1 jmaroumougom sib     0 nov  3 2011 mapping-jmaroumougom
srwxr-xr-x  1 ndebs      lbm       0 jun 28 15:11 mapping-ndebs
srwxr-xr-x  1 root       root      0 avr 17 2009 mapping-root
srwxr-xr-x  1          5000 root      0 avr 28 2009 mapping-toto
drwx----- 2 llegrand   inra      4096 jui  5 10:28 mozilla-media-cache
drwxr-xr-x 258 hfcollector application 20480 jui  6 04:48 ODV_hfcollector
drwx----- 3 nhenry     eppo      4096 jui  2 13:43 openmpi-sessions-nhenry@n0.sb-roscoff.fr_0
drwx----- 2 acormier   genetique 4096 mai  3 20:57 orbit-acormier
drwx----- 2 cock       genetique 4096 avr  7 13:18 orbit-cock
drwx----- 2 corre      sib       4096 jun 24 09:58 orbit-corre
drwx----- 2 ewcorre    lbm       4096 jui  3 13:50 orbit-ewcorre
drwx----- 2 gdm        gdm       4096 avr  5 18:30 orbit-gdm
drwx----- 2 hfcollector application 4096 jui  4 10:16 orbit-hfcollector
drwx----- 2 lecorguille sib       4096 jun 20 02:48 orbit-lecorguille
drwx----- 2 llegrand   inra      4096 jui  5 10:35 orbit-llegrand
drwx----- 2 mhoebeke    sib       4096 jun 25 13:30 orbit-mhoebeke
drwx----- 2 ndebs      lbm       4096 jun 28 16:17 orbit-ndebs
drwx----- 2 stage02     stage     4096 mai 13 10:40 orbit-stage02
drwx----- 2 wcarre      sib       4096 jun 20 09:53 orbit-wcarre
srwxr-xr-x  1 ewcorre    lbm       0 nov 10 2011 OSL_PIPE_6108_SingleOfficeIPC_eebd8121e860c31ca9a23ed86a44ce
drwxr-xr-x  4 root       root      4096 sep 30 2009 perl5
drwxr-xr-x  2 acormier   genetique 4096 jun 25 15:58 perl_formation
drwxr-xr-x  2 mhoebeke    sib       4096 jun 25 11:45 phyloclusters
srwxr-xr-x  1 root       root      0 avr 11 2012 sfcblLocalSocket
drwxr-xr-x  5 root       root      4096 fév 20 2010 sge
-r-----  1 root       root      3066 mai 23 2011 shadow
-rw-r--r--  1 root       root     10978 jui  5 23:00 stat_sge.txt
drwxr-xr-x  3 root       root      4096 jun 26 2012 toto
[acormier@n0 tmp]$
```

Web applications

- Web interface
 - Galaxy
 - R-studio: dedicated to R



<http://r.sb-roscoff.fr/>



<http://galaxy3.sb-roscoff.fr/>



CONNEXION AND STORAGE

Prerequisite

- Account
 - <http://abims.sb-roscoff.fr/account>
 - support.abims@sb-roscoff.fr
- Email
- X11 terminal
 - Windows: Putty or MobaXterm (X11)
 - Mac OS : XQuartz (X11)
 - Linux: integrated
- Text editor
 - Vim, nano, gedit, emacs...
- SFTP client

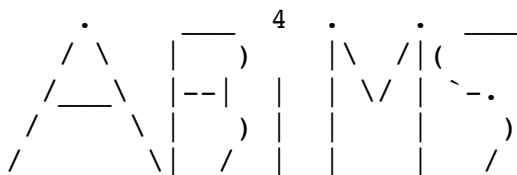


Connection

```
$ ssh -Y foobar@bioinfo.sb-roscoff.fr # -Y → for graphic (X11) flux redirection
```

```
$ ssh -Y foobar@bioinfo.sb-roscoff.fr # -Y → for graphic (X11) flux redirection
```

Last login: Wed May 17 13:39:19 2017 from 192.168.4.180



Analysis and Bioinformatics for Marine Science

<http://abims.sb-roscoff.fr> - support.abims@sb-roscoff.fr

Please have a look at the training material:

http://abims.sb-roscoff.fr/sites/abims.sb-roscoff.fr/files/formation_2016/formation_cluster_v5.1.pdf

IMPORTANT:

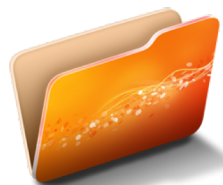
- n221: Never launch job on this server -> Use a qlogin
- /home: Never launch job from this space
- /projet: Use your /projet folder for its performance, its volumetry and its independence from the /home space
- /scratch: For your huge temporary files, please use /scratch but note that files older than 30 days are automatically deleted

CITATION: Please cite the platform ABiMS in the Acknowledgement of your future publication

WARNING: 27/03/2017 - We are facing some issues with the disk server NZ.
We will inform you as soon as we get further information.
Meanwhile, we can expect some random interruptions. Sorry for that!

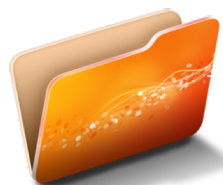
WORKING DIRECTORIES

- Personal data
- Shared data:
 - By team / group
 - By UMR
 - For a community
 - Public data
- Databank
 - Genbank, Uniprot, InterPro banks, etc.
 - Format/Index : Fasta, Blast, Bowtie2, BWA, Daemon, etc.
 - Private & Public



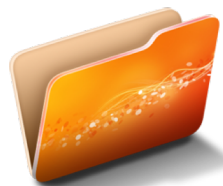
projet

- per person
- by team
- by subject



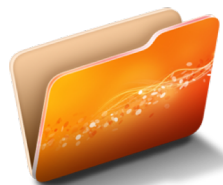
home

- only for connexion (Environment variable)



db

- Databank (Blast, Genbank, Interpro...)

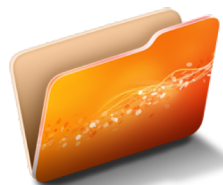


Galaxy

- ftp
- files

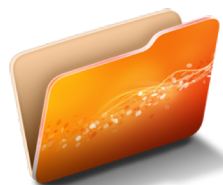


Workspace



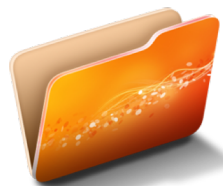
projet

→ nz
→ nz11



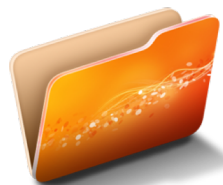
home

→ brazil



db

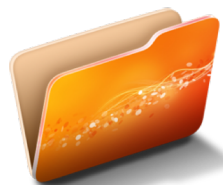
→ banko



Galaxy

→ cfs10

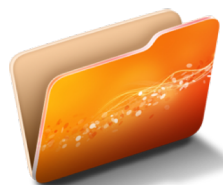




projet



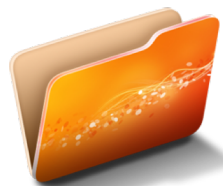
Partial backup



home



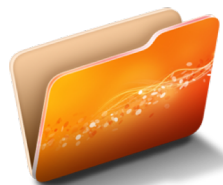
Partial backup



db



No backup



Galaxy



No backup



scratch
scratch2

→ nz

→ nz11



20 To

Space for all the primary analysis - generated huge amount of temporary/useless files

- **Mutualised storage** between all users
- Data are **not backed up**
- All files older than **90 days** are automatically **deleted**

Working directory: ~~home~~

When I'm connecting, I arrive in my:



home

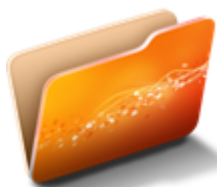
```
$ pwd #print working directory
```

```
/home/fr2424/sib/foobar
```

Not for storage / computing

Working directory: scratch

I have to work in:



scratch

For all analysis

```
$ cdscratch
```

```
$ pwd
```

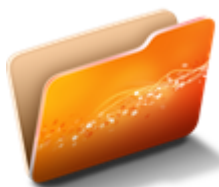
```
/scratch/fr2424/sib/foobar
```

- Structuration:
→ by user
- Shared between all nodes
- Available from outside

- No particular structuration
- Don't forget this:
 - **All files older than 90 days are automatically deleted** (based on the last modification date)

Working directory: projet

I have to save my results in:



projet

To store raw data, final
results and scripts

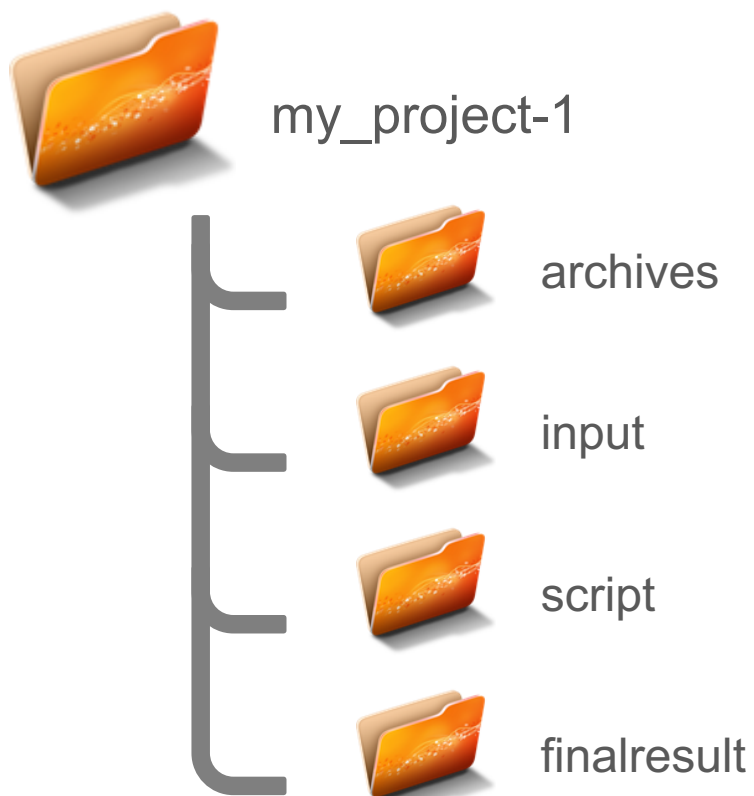
```
$ cdprojet #alias for fast moving in my project directory
$ pwd

/projet/fr2424/sib/foobar
```

- Structuration:
 - by team: /projet/fr22424/sib/foobar
 - by subject: /projet/abims/ectocarpus
- Shared between all nodes
- Available from outside

Working directory: projet – structure

Each project needs to have particular folders:



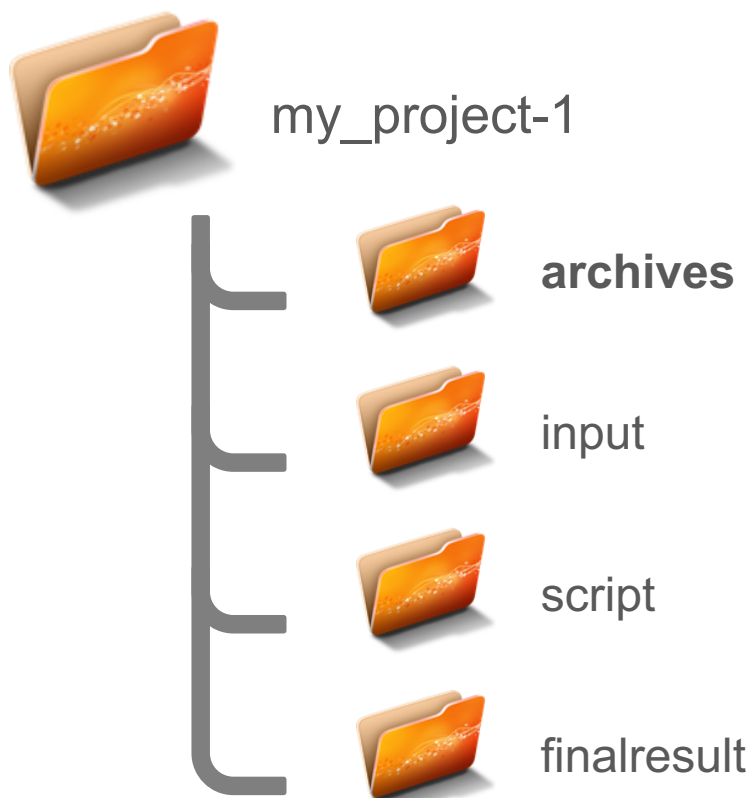
Backup system: **by inclusion.**

Only these folder are saved:

- **finalresult**
- **script**
- **archives**

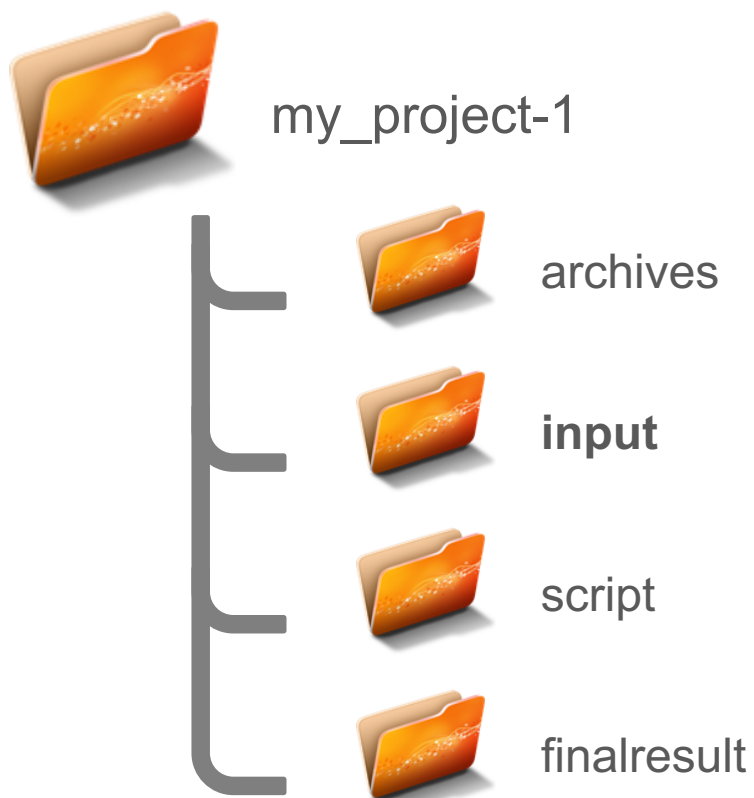
Pay attention to typo! Case sensitive

Working directory: projet – structure



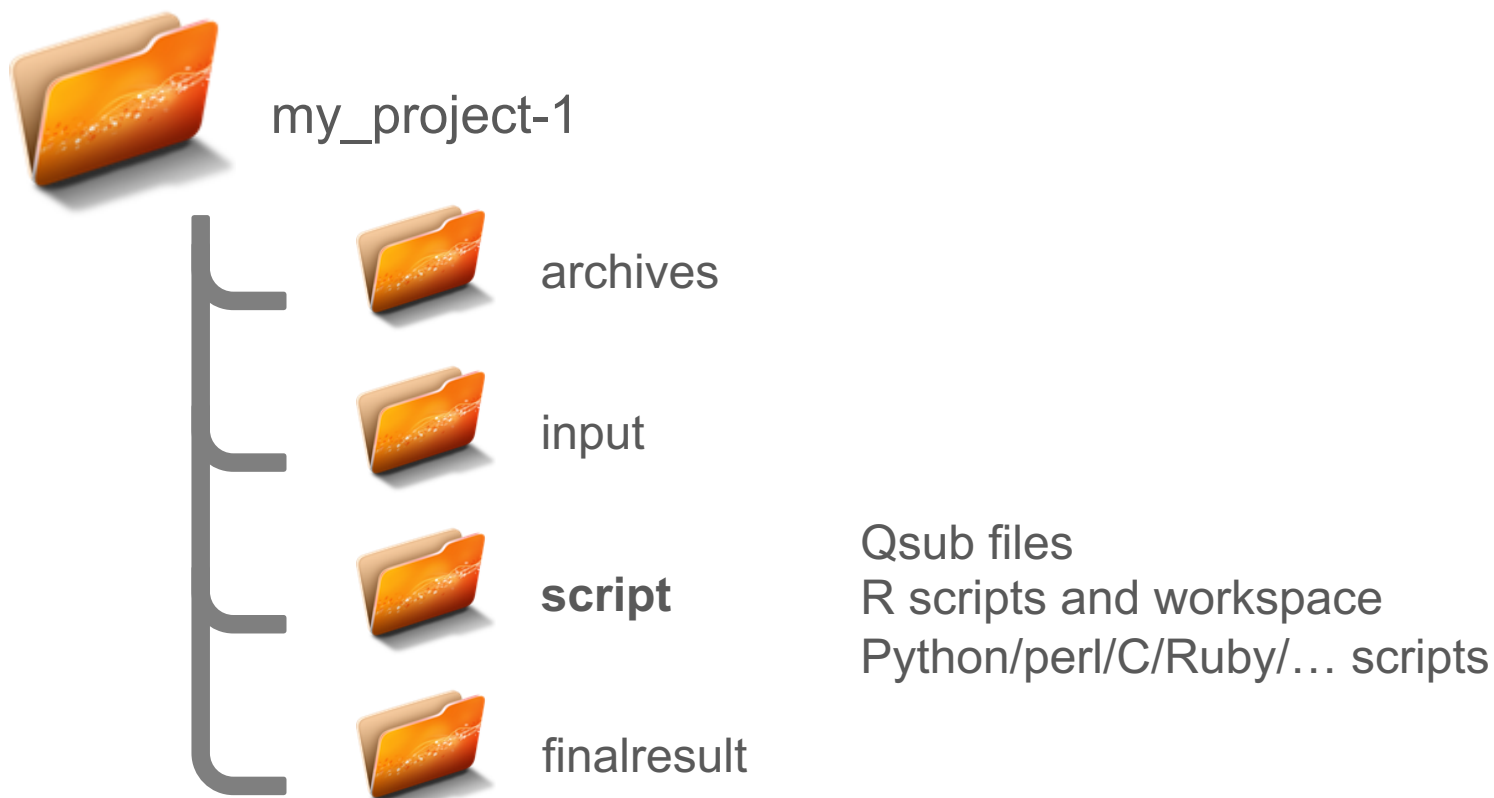
Original data sources.
Rarely used, only for archiving.
E.g. Raw data from sequencing (Sanger, DNA-seq, RNA-seq, etc)

Working directory: projet – structure

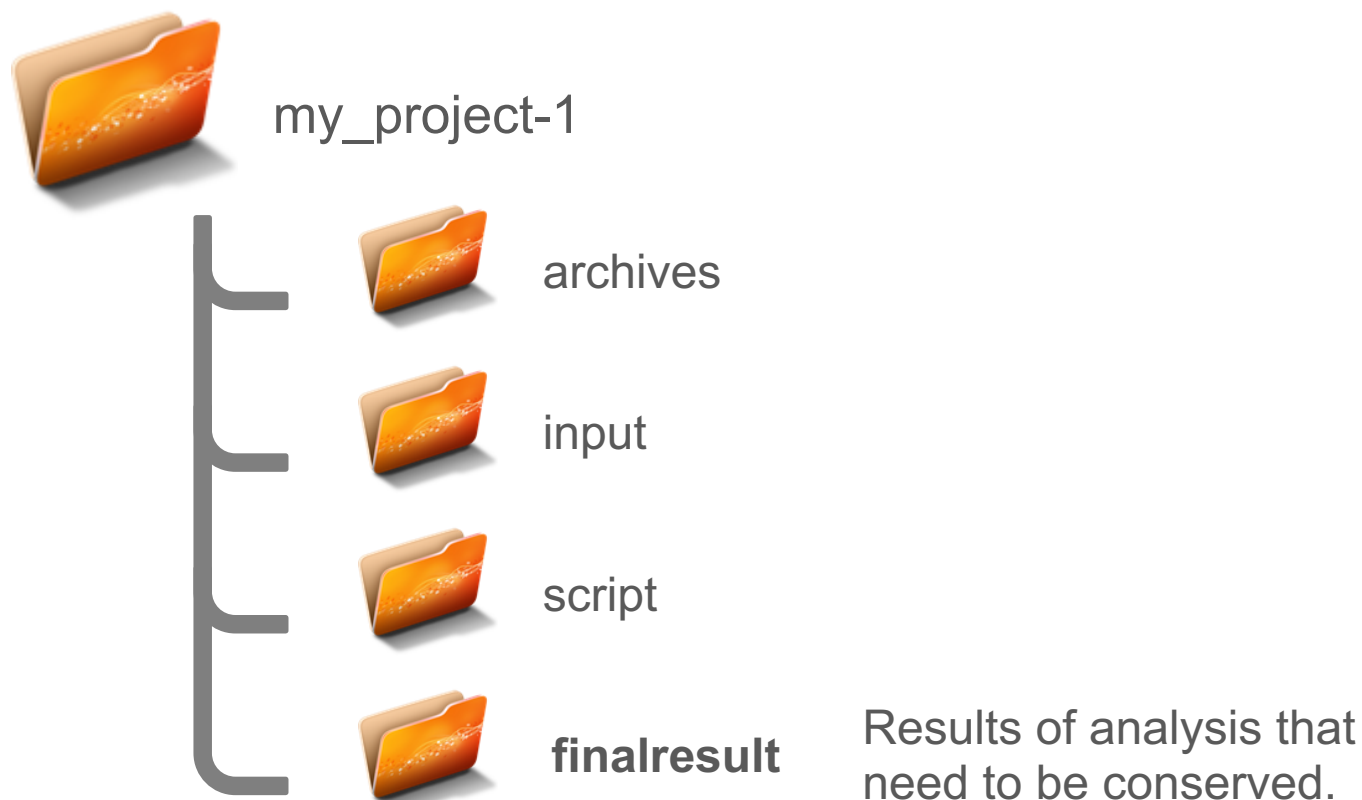


Data used as input file for analysis
E.g. Cleanded data from sequencing,
fasta files, etc

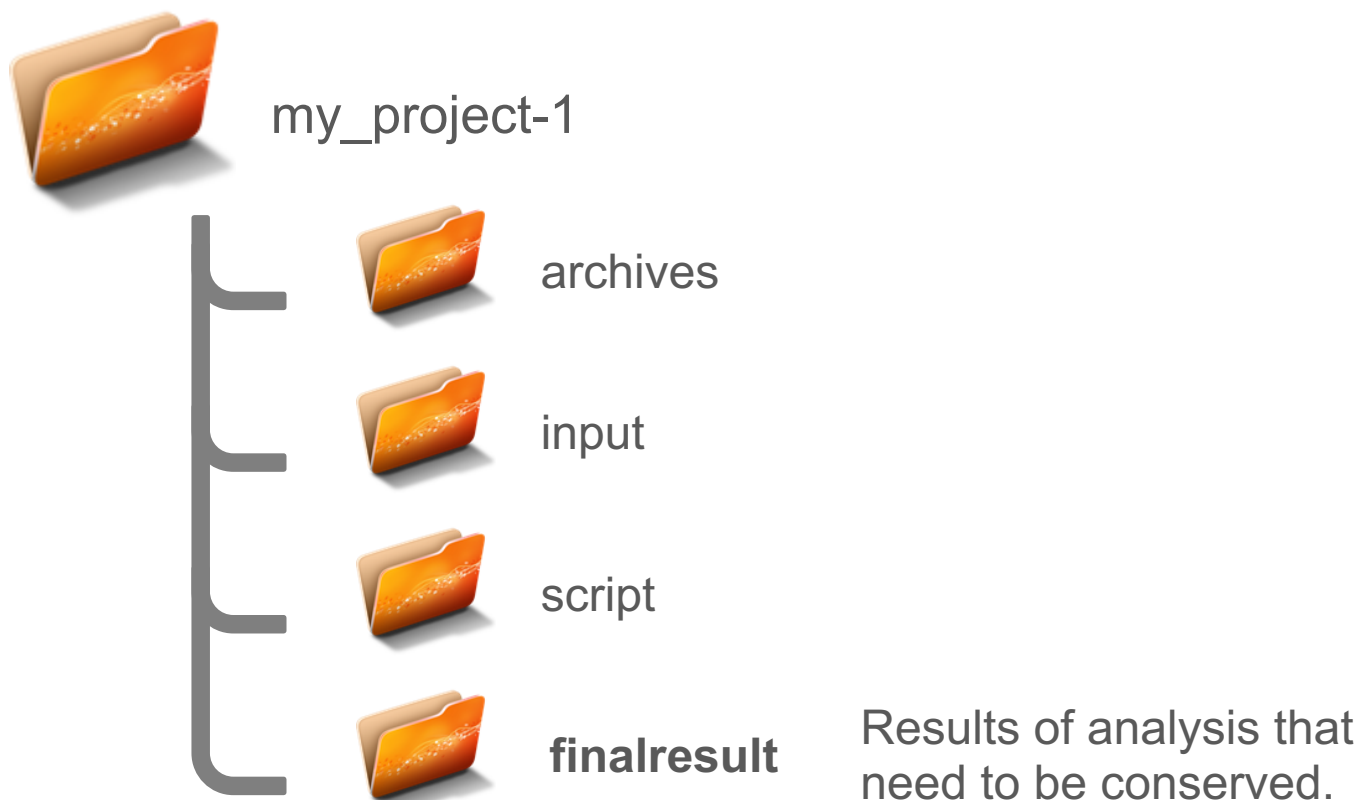
Working directory: projet – structure



Working directory: projet – structure



Working directory: projet – structure



~/.bashrc

```
alias mkp='mkdir archive input script tmp finalresult finalresult/document; \  
ln -s finalresult output; ln -s finalresult/document'
```



Regularly, check the volume of my project to prevent saturation. The storage is not by user, but by team...

```
$ df -h . #report filesystem disk space usage
Sys. de fich.      Tail. Occ. Disp. %Occ. Monté sur
cfs1:/projet/fr4242/sib  1,4T  651G  658G  50% /projet/fr4242/sib
```

```
$ cdscratch
```

```
$ du -sh * #size of each file/folder -> who is the biggest?
```

```
68G      assembly
341G     pagit
3,8G     remapping
12K      cache_tmp
17M      chr_similarity
1008M    galaxy_dataset
669M     metrics
2.1M     Tes
```

```
$ du -sh assembly/*
```

```
11G      assembly/transcriptome_V1
9.8G     assembly/transcriptome_V2
48G      assembly/trinity
```



Compress your data!

```

$ ll -h

-rw-rw-r--+ 1 foobar      sib  26G mars   8 08:16 140220_SND393_B_L006_GPO-12_R1.fastq
-rw-rw-r--+ 1 foobar      sib  26G mars   8 08:16 140220_SND393_B_L006_GPO-12_R2.fastq

$ gzip 140220_SND393_B_L006_GPO-12_R1.fastq
$ gzip 140220_SND393_B_L006_GPO-12_R2.fastq

$ ll -h

-rw-rw-r--+ 1 foobar      sib  7,7G mars   7 12:25 140220_SND393_B_L006_GPO-12_R1.fastq.gz
-rw-rw-r--+ 1 foobar      sib  7,9G mars   7 12:29 140220_SND393_B_L006_GPO-12_R2.fastq.gz
    
```

Some softwares are capable to use directly compressed data
(TopHat2, Trimmomatic,...)

RESSOURCES



- Applications (x700)
 - /usr/local/genome2/
 - Localisation of all software available on the cluster
 - Soon: a list of all tools
 - /usr/local/genome2/script/
 - Scripts developed by people of the SBR
 - Just send an email to support.abims@sb-roscoff.fr if you want to share your scripts
 - Conda (See next slide)
- The software components are shared:
 - One installation in a unique place
 - Request to add or update: support.abims@sb-roscoff.fr

- The issues
 - Rather old server distributions (CentOS 6.x) so difficulties to compile and install the tools
 - Reproducibility / Portability

BIOCONDA[®]

- <http://abims.sb-roscoff.fr/resources/tools/howto#conda>
- <https://bioconda.github.io/>

Ex: <https://www.google.com/search?q=conda+prokka>

```
$ # How an admin install a tool  
$ conda create -n stacks-2.3b stacks=2.3b
```

```
$ # How activate an environment  
$ $CONDA3/conda env list  
  
$ source $CONDA3/activate stacks-2.3  
(stacks-2.3) $ ustacks --version  
ustacks 2.3  
(stacks-2.3) $ source $CONDA3/deactivate  
  
$ source $CONDA3/activate stacks-2.3b  
(stacks-2.3b) $ ustacks --version  
ustacks 2.3b  
(stacks-2.3b) $ source $CONDA3/deactivate
```



Singularity

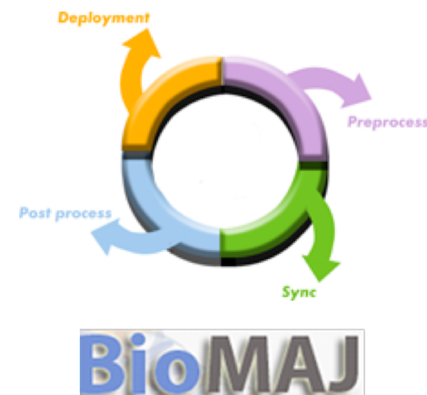
https://www.sylabs.io/guides/3.0/user-guide/quick_start.html

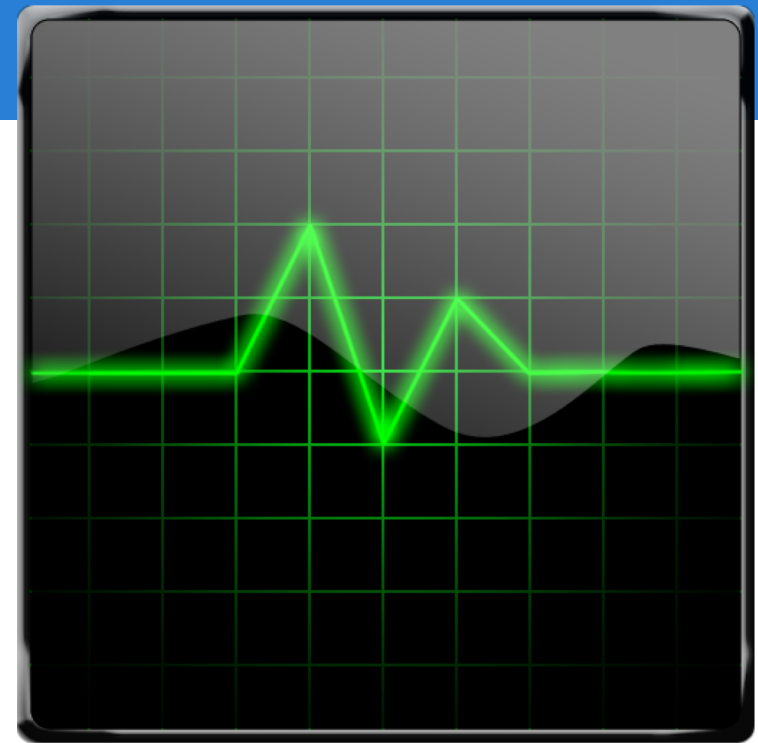
Not available on n221, please use a qlogin

```
$ # if there is only one binary
$ singalp-5.0.sif --help

$ # if there are many, you will have to mention the binary name
$ singularity exec /usr/local/genome2/singularity/singalp-5.0.sif singalp --help
```

- /db/
 - Public databank:
 - NCBI
 - GenBank
 - UniProt
 - InterPro
 - Etc
 - SBR databank
 - Start with the prefix “sbr_”
 - Description of these databank is currently in progress





How to use the cluster?

SUN GRID ENGINE - SGE

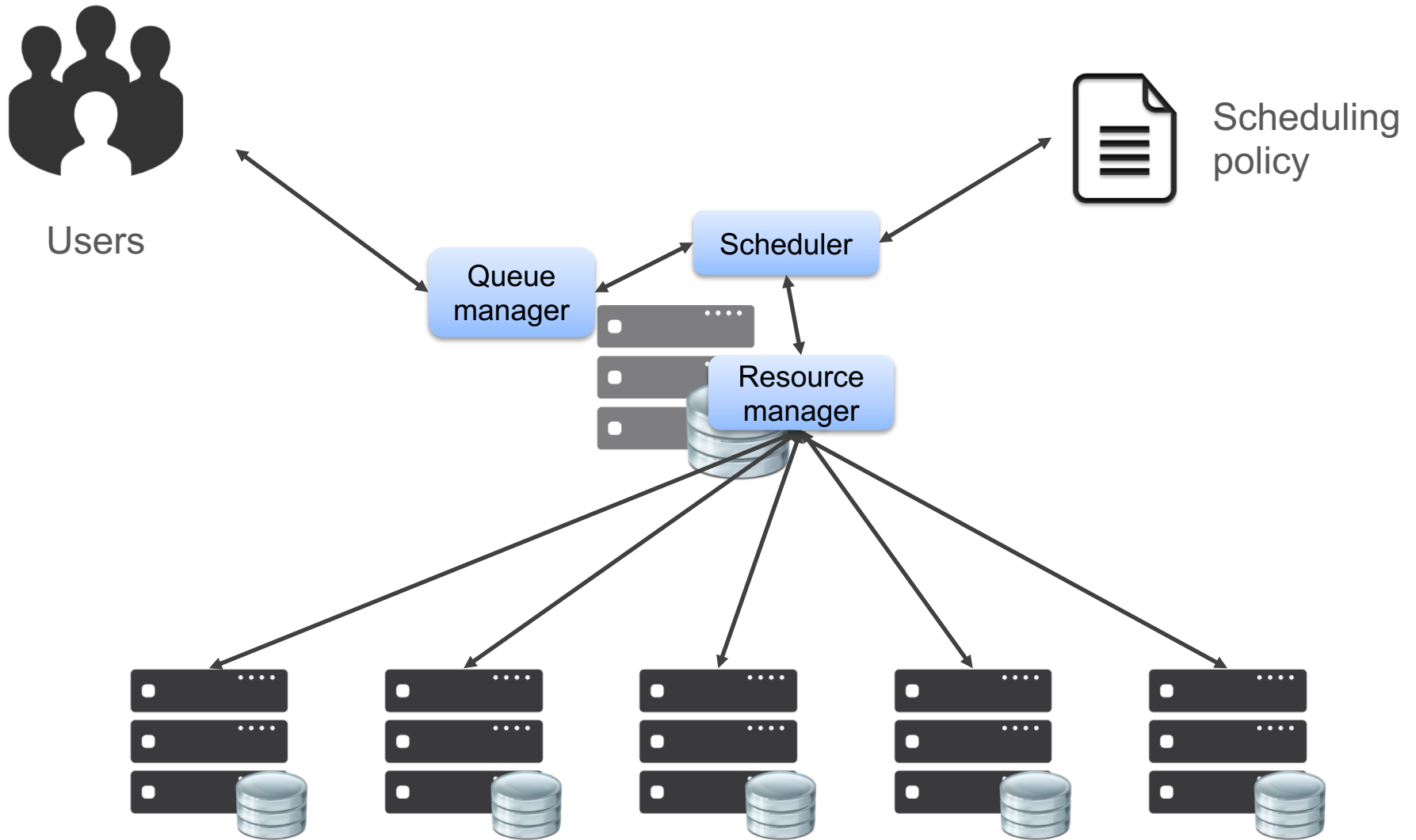
Job management system

- Sun Grid Engine (SGE)
 - Scheduler in charge of the jobs management
 - User interface for submitting and controlling jobs
- Task scheduling
 - Resources allocation
 - Nodes load
 - Priority
- Management policy and resource sharing
 - CPU / Memory
 - Execution time
- Reporting and errors
 - History
 - Usage statistics



- Job
 - Task unit
 - There are several types of jobs:
 - Batch (script)
 - Interactive
 - Serial vs parallel
 - Serial: only need 1 processor
 - Parallel: require more than 1 processor
- Slots
 - Number of jobs allowed on one node
- Queue
 - Type of resources (node groupe, execution time...)
- Priority
 - **Fair Share** : calculated on 1 week → sliding window

Job management system



	Time out	Available resources	Purpose
short.q	12 hours	High priority - 50 % of all CPU (- n99)	Regular jobs
long.q	10 days	Standard priority - 50 % of all CPU (- n99)	Long jobs
infinite.q	no limit	Low priority - 25 % of all CPU (- n99)	Really really long jobs
bigmem.q			For traitements requiring a lot of RAM
qlogin.q	2 days		For connections in qlogin mode
clc.q		n76	For CLC use (please contact support.abims@sb-roscoff.fr).

Max load: 1,25

→ By default, no queue!

Nodes groups

DIFFERENTS TYPES OF NODE

Group	Node	Type	Function	Processor	RAM
@@bigcpu	n40-n55	Dell blade M630	Multithreaded traitements	Intel 32 x 2.4 GHz	128 Go
	n60-n75	Dell blade M605	Reserved for some applications	AMD 8 x 2.4 GHz	32 Go
@@bignode	n76-n79	Dell R815	Multithreaded traitements Memory usage	AMD 48 x 2.2 GHz	256 Go
	n56-n59		Multithreaded traitements Memory usage	AMD 48 x 2.2 GHz	256 Go
@@bigcpu	n80-n95	Dell C6220	Multithreaded traitements	Intel 32 x 2.2 GHz	128 Go
@@bigmem	n99	Dell R910	Memory usage	Intel 40 x 2.0 GHz	1 To

\$ `qstat -g c` #displays the available queues

CLUSTER QUEUE	CQLOAD	USED	RES	AVAIL	TOTAL	aoACDS	cdsuE
short.q	0.32	140	0	184	324	0	0
long.q	0.32	93	0	231	324	0	0
infinite.q	0.32	0	0	66	66	0	0
bigmem.q	0.60	20	0	20	40	0	0
qlogin.q	0.30	7	0	23	30	0	0
clc.q	0.28	10	0	38	48	0	0
formation.q	0.61	3	0	57	60	0	0
galaxy.q	0.72	0	0	140	140	0	0
galaxy1.q	0.33	0	0	72	72	0	0

The queues freely available:

- short.q
- long.q
- infinite.q
- qlogin.q → for interactive jobs

On request:

- bigmem.q : for jobs that require a lot of RAM
- clc.q : for CLC Assembly Cell

Monitoring

\$ **ghost** #liste of all nodes

HOSTNAME	ARCH	NCPU	LOAD	MEMTOT	MEMUSE	SWAPTO	SWAPUS
n0	1x24-amd64	8	0.10	7.8G	841.7M	4.0G	81.6M
n60	1x24-amd64	8	0.03	31.5G	2.3G	1.0G	656.0K
n61	1x24-amd64	8	0.03	31.5G	350.3M	1.0G	180.0K
n62	1x24-amd64	8	1.32	31.5G	208.5M	1.0G	80.5M
n63	1x24-amd64	8	0.03	31.5G	1.8G	1.0G	72.2M
n64	1x24-amd64	8	1.00	31.5G	335.0M	1.0G	82.4M
n76	1x24-amd64	48	13.59	252.0G	22.4G	2.0G	28.5M
n77	1x24-amd64	48	11.12	252.0G	21.3G	2.0G	240.0K
n78	1x24-amd64	48	5.02	252.0G	22.4G	2.0G	58.8M
n79	1x24-amd64	48	37.07	252.0G	24.6G	2.0G	0.0
n80	1x24-amd64	32	22.14	126.0G	2.9G	1024.0M	11.0M
n81	1x24-amd64	32	32.02	126.0G	2.8G	1024.0M	0.0
n82	1x24-amd64	32	32.03	126.0G	2.8G	1024.0M	0.0
n83	1x24-amd64	32	32.02	126.0G	2.8G	1024.0M	0.0
n84	1x24-amd64	32	32.02	126.0G	3.1G	1024.0M	0.0
n99	1x24-amd64	40	24.00	1009.7G	238.6G	4.0G	107.6M

Allows to check load level on each node:

- Load level on the cluster

\$ **qstat** #shows all jobs

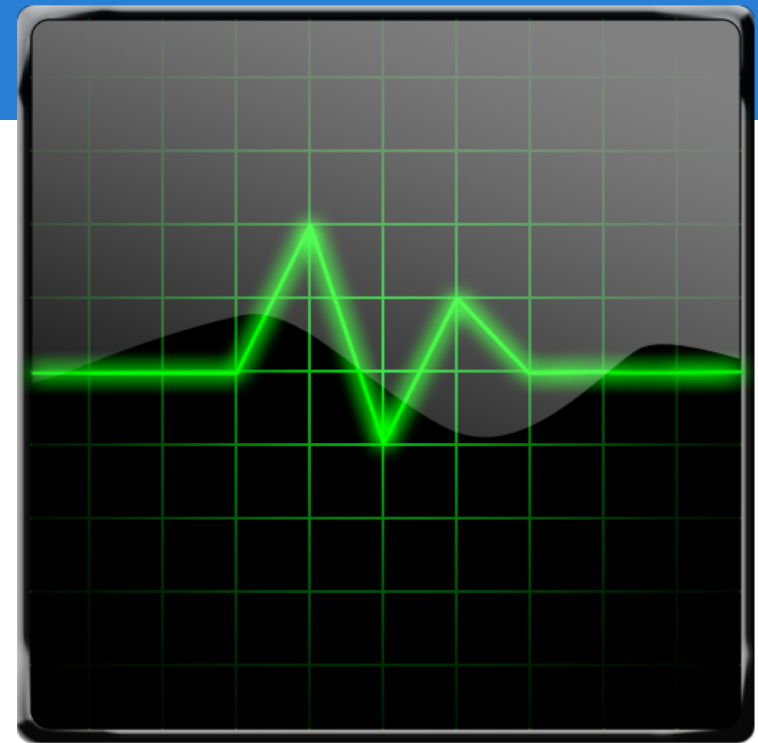
job-ID	prior	name	user	state	submit/start at	queue	slots	ja-task-ID
1236477	0.50000	QLOGIN	wcarre	r	04/30/2012 15:10:26	long.q@n77	1	
1236479	0.50000	QLOGIN	wcarre	r	04/30/2012 15:11:12	long.q@n77	1	
1268627	1.00000	QLOGIN	balzano	r	05/03/2012 09:39:24	long.q@n74	1	
1302170	0.06667	alpha0.sh	mrescan	r	05/06/2012 12:05:48	long.q@n64	1	
1302173	0.06667	alpha3.sh	mrescan	r	05/06/2012 12:06:18	long.q@n72	1	
1302174	0.06667	alpha4.sh	mrescan	r	05/06/2012 12:06:33	long.q@n70	1	
1302175	0.06667	alpha5.sh	mrescan	r	05/06/2012 12:06:43	long.q@n63	1	
1302261	0.06667	alpha6.sh	mrescan	r	05/06/2012 12:07:03	infinite.q@n60	1	
1314908	0.21765	evol.sh	ablanckaert	qw	05/07/2012 10:04:39		20	
2216309	0.00045	sge_blastn	gfarrant	qw	05/23/2013 14:43:03		1	69-2379:1

- Prior: priority level
 - Higher when close to 1.0000
- State
 - r: running
 - qw: pending
 - Eqw: in error
- Slots: cores used
- Ja-task-ID: job array

Jobs type

- Interactive mode: qlogin
 - Short job and/or development
 - Prerequisite: none
 - Note: **disconnect you at the end of the session**
- Batch mode: qsub
 - Heavy jobs
 - Prerequisite: text editor
 - One script per job

n221/bioinfo (master node) never should be used for computing!



qlogin

SUN GRID ENGINE - SGE

qlogin: Interactive mode

- Connexion on:
 - A queue: `qlogin -q qlogin.q`
 - A node : `qlogin -q qlogin.q@n72`
 - A group : `qlogin -q qlogin.q@@bignode`

```
$ qlogin -q qlogin.q
```

```
Your job 2217414 ("QLOGIN") has been submitted  
waiting for interactive job to be scheduled ...  
Your interactive job 2217414 has been successfully scheduled.  
Establishing /opt/sge/qlogin.sh session to host n78 ...  
Last login: Mon Apr 15 10:22:01 2013 from n0.sb-roscoff.fr
```

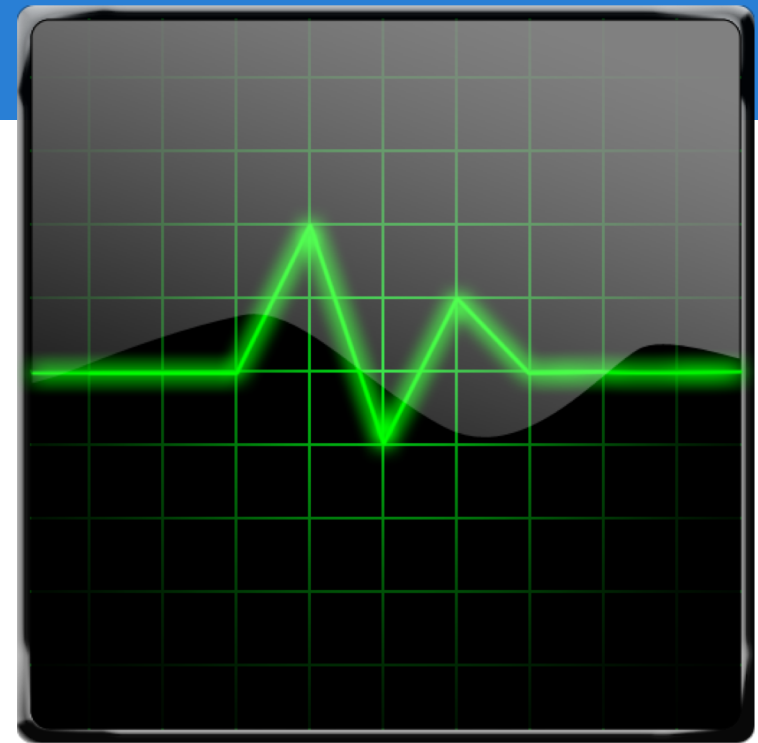
```
@n78$ cdprojet
```

```
.  
<my test>
```

```
.  
@n78$ exit
```

```
Connection to n78 closed.  
/opt/sge/qlogin.sh exited with exit code 0
```

```
$
```



qsub

SUN GRID ENGINE - SGE

qsub: Batch mode

- Progress:
 - Script edition
 - Choose the right queue
 - Submitting → Execution → Results
- Edition
 - In command line: vi, vim, nano...
 - In graphic mode: gedit, kate...

How to?

1. Prepare script of executable commands
2. Submit to batch system
3. Use the job ID for job control (query status, cancel, ...)
4. Check the job status (no execution error)

1. Prepare script of executable commands

The minimum

	<code>script.qsub</code>
Header	<code>{ #!/bin/bash #\$ -S /bin/bash #\$ -v #\$ -cwd</code>
cmd lines	<code>{ echo "Hello world!" > output.txt</code>

Essential for qsub:

- The header:
 - Shell path
 - -S : path to shell (for SGE)
 - -V : load the environment variables (~/.bashrc → /etc/bashrc)
 - -cwd : work in the current working directory
- The command line(s)

1. Prepare script of executable commands

Other practical settings

	script.qsub
Header	{ #!/bin/bash #\$ -S /bin/bash #\$ -V #\$ -cwd #\$ -o blastn.out #\$ -e blastn.err }
cmd lines	{ blastp -db nr -query query_1.fa -out blastout_1.txt blastp -db nr -query query_2.fa -out blastout_1.txt }

- The header:
 - -o stdout filename
 - -e stderr filename

1. Prepare script of executable commands

Other practical settings

	script.qsub
Header	<pre>#!/bin/bash #\$ -S /bin/bash #\$ -V #\$ -cwd #\$ -M foo.bar@sb-roscoff.fr #\$ -m bea</pre>
cmd lines	<pre>blastp -db nr -query query_1.fa -out blastout_1.txt blastp -db nr -query query_2.fa -out blastout_1.txt</pre>

- The header:
 - -m b|e|a|s|n|...: send mail at beginning|end|...of the job
 - -M: E-mail address for notification

1. Prepare script of executable commands

Other practical settings

	script.qsub
Header	{ #!/bin/bash #\$ -S /bin/bash #\$ -V #\$ -cwd #\$ -l mem_free=50G }
cmd lines	{ blastp -db nr -query query_1.fa -out blastout_1.txt blastp -db nr -query query_2.fa -out blastout_1.txt }

- The header:
 - -l mem_free=XXG

2. Submit to batch system

```
$ qsub -q short.q script.qsub
```

Your job 2217418 ("script.qsub") has been submitted

-q : select a queue

script.qsub

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -q short.q
```

```
blastp -db nr -query query_1.fa -out blastout_1.txt
blastp -db nr -query query_2.fa -out blastout_1.txt
```

```
$ qsub script.qsub
```

Your job 2217418 ("blast.qsub") has been submitted

2. Submit to batch system

```
$ qsub script.qsub
```

Your job 2217418 ("script.qsub") has been submitted

script.qsub

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -q short.q
```

```
blastp -db nr -query query_1.fa -out blastout_1.txt
blastp -db nr -query query_2.fa -out blastout_1.txt
```

- You can launch a "classic" script but ...

script.sh

```
#!/bin/bash
```

```
blastp -db nr -query query_1.fa -out blastout_1.txt
blastp -db nr -query query_2.fa -out blastout_1.txt
```

```
$ qsub -S /bin/bash -V -cwd -q short.q script.sh
```

Your job 2217418 ("script.sh") has been submitted

2. Submit to batch system

- The choice of the queue is subject to several criteria ...
 - Job duration:
 - < 12 hours → short.q
 - < 10 days → long.q
 - > 10 days → infinite.q
 - RAM
 - < 4 Go / CPU → @@bigcpu
 - > 4 Go / CPU → @@bignode
 - > 25 Go / CPU → bigmem.q
 - Tools:
 - CLC Assembly Cell → clc.q

```

$ qsub -q short.q script.qsub
$ qsub -q short.q@@bigcpu script.qsub
$ qsub -q long.q@@bignode script.qsub
$ qsub -q bigmem.q script.qsub
$ qsub -q clc.q script.qsub
    
```

OR

script.qsub

```

#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -q short.q@@bigcpu
    
```

```

blastall -p blastp -d nr -i query_1.fa ...
blastall -p blastp -d nr -i query_2.fa ...
    
```

2. Submit to batch system

- Multithreading

script.qsub

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -q short.q@@bigcpu
#$ -pe thread 8

bowtie2 -1 r1.fastq -2 r2.fastq -x ref -S align.sam --threads 8
```

2. Submit to batch system

- Multithreading

script.qsub

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -q short.q@@bigcpu
#$ -pe thread 8

bowtie2 -1 r1.fastq -2 r2.fastq -x ref -S align.sam --threads 8
```

BEWARE: You must set the same value in both "-pe thread" and software settings

In this case: you have only book 1 cpu but you will eventually use 8 CPU
-> Overload! It's bad!

2. Submit to batch system

- Multithreading

script.qsub

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -q short.q@@bigcpu
#$ -pe thread 8

bowtie2 -1 r1.fastq -2 r2.fastq -x ref -S align.sam --threads 8
```

BEWARE: You must set the same value in both "-pe thread" and software settings

In this case: you have book 8 slots but you only use 1 CPU
-> Useless

2. Submit to batch system

- Multithreading

script.qsub

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -q short.q@@bigcpu
#$ -pe thread 8

bowtie2 -1 r1.fastq -2 r2.fastq -x ref -S align.sam --threads $NSLOTS
```

\$NSLOTS will take the value you setted for `-pe thread`

2. Submit to batch system

- Multithreading
 - TopHat: -p / --num-threads
 - Bowtie2: -p / --threads
 - Trinity: --CPU
 - CLC Assembly Cell: --cpus

Multithreading is not possible with all software

RTFM

2. Submit to batch system



Job 1236477 (tophat.sh) Started
User = acormier
Queue = long.q
Host = n77
Start Time = 05/25/2013 13:30:39

2. Submit to batch system

- ... But also rules, because you are not alone on the cluster!
 - **Work in the scratch directory**
 - Choose the more adapted queue (by default, used long.q)
 - Disconnect you from your qlogin

3. Use the job ID for job control (status,...)

\$ **qstat** #shows all jobs

job-ID	prior	name	user	state	submit/start at	queue	slots	ja-task-ID
1236477	0.50000	QLOGIN	wcarre	r	04/30/2012 15:10:26	long.q@n77	1	
1236479	0.50000	QLOGIN	wcarre	r	04/30/2012 15:11:12	long.q@n77	1	
1268627	1.00000	QLOGIN	balzano	r	05/03/2012 09:39:24	long.q@n74	1	
1302174	0.06667	alpha4.sh	mrescan	r	05/06/2012 12:06:33	long.q@n70	1	
1302175	0.06667	alpha5.sh	mrescan	r	05/06/2012 12:06:43	long.q@n63	1	
1302261	0.06667	alpha6.sh	mrescan	r	05/06/2012 12:07:03	infinite.q@n60	1	
1314908	0.21765	evol.sh	ablanckaert	qw	05/07/2012 10:04:39		20	
2216309	0.00045	sgc_blastn	gfarrant	qw	05/23/2013 14:43:03		1	69-2379:1

- Prior: priority level
- State
 - r: running
 - qw: pending
 - Eqw: in error
- Slots: cores used

\$ **\qstat** #shows my jobs

job-ID	prior	name	user	state	submit/start at	queue	slots	ja-task-ID
1236477	1.00000	tophat.sh	acormier	r	05/25/2013 15:10:26	long.q@n77	8	

3. Use the job ID for job control (status,...)

```
$ qdel 2217417 #deletion by the job-ID
```

acormier has registered the job 2217417 for deletion

```
$ qdel -f 2217418
```

acormier has registered the job 2217418 for deletion

```
$ qdel -u acormier #deletion by user name
```

acormier has registered the job 2217419 for deletion

acormier has registered the job 2217420 for deletion

acormier has registered the job 2217421 for deletion

acormier has registered the job 2217422 for deletion

Job 2217424 (clc_mapping_info.sh)
 was killed by [acormier@n0.sb-roscoff.fr](#)

Job 2217424 (clc_mapping_info.sh) Aborted
 Exit Status = 137
 Signal = KILL
 User = acormier
 Queue = [clc.q@n76](#)
 Host = n76.sb-roscoff.fr
 Start Time = 05/30/2013 21:24:06
 End Time = 05/30/2013 21:24:08
 CPU = 00:00:01
 Max vmem = 75.023M
 failed assumedly after job because:
 job 2217424.1 died through signal KILL (9)

4. Check the job status (no execution error)

- Logs



```

Job 1236477 (tophat.qsub) Complete
User                = acormier
Queue               = long.q@n77
Host                = n77
Start Time          = 05/25/2013 13:30:39
End Time            = 05/25/2013 17:39:00
User Time           = 04:01:31
System Time         = 00:06:02
Wallclock Time      = 04:08:21
CPU                 = 04:07:33
Max vmem            = 10.976G
Exit Status         = 0
    
```

<my_script>.**e**<job-ID> : error file and/or progress bar

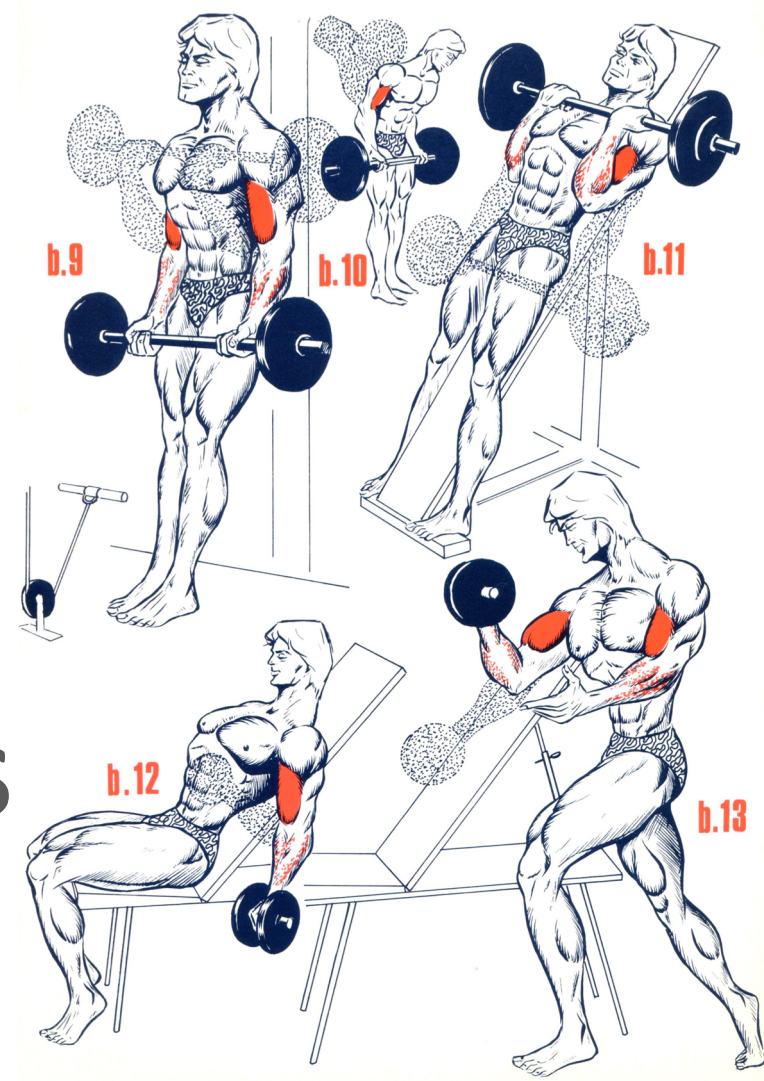
<my_script>.**o**<job-ID> : results, except if the program provides an option to output file.

And in mutlithreading mode:

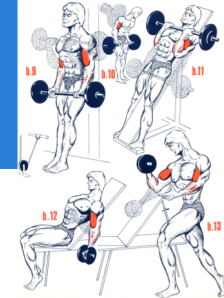
<my_script>.**pe**<job-ID>

<my_script>.**po**<job-ID>

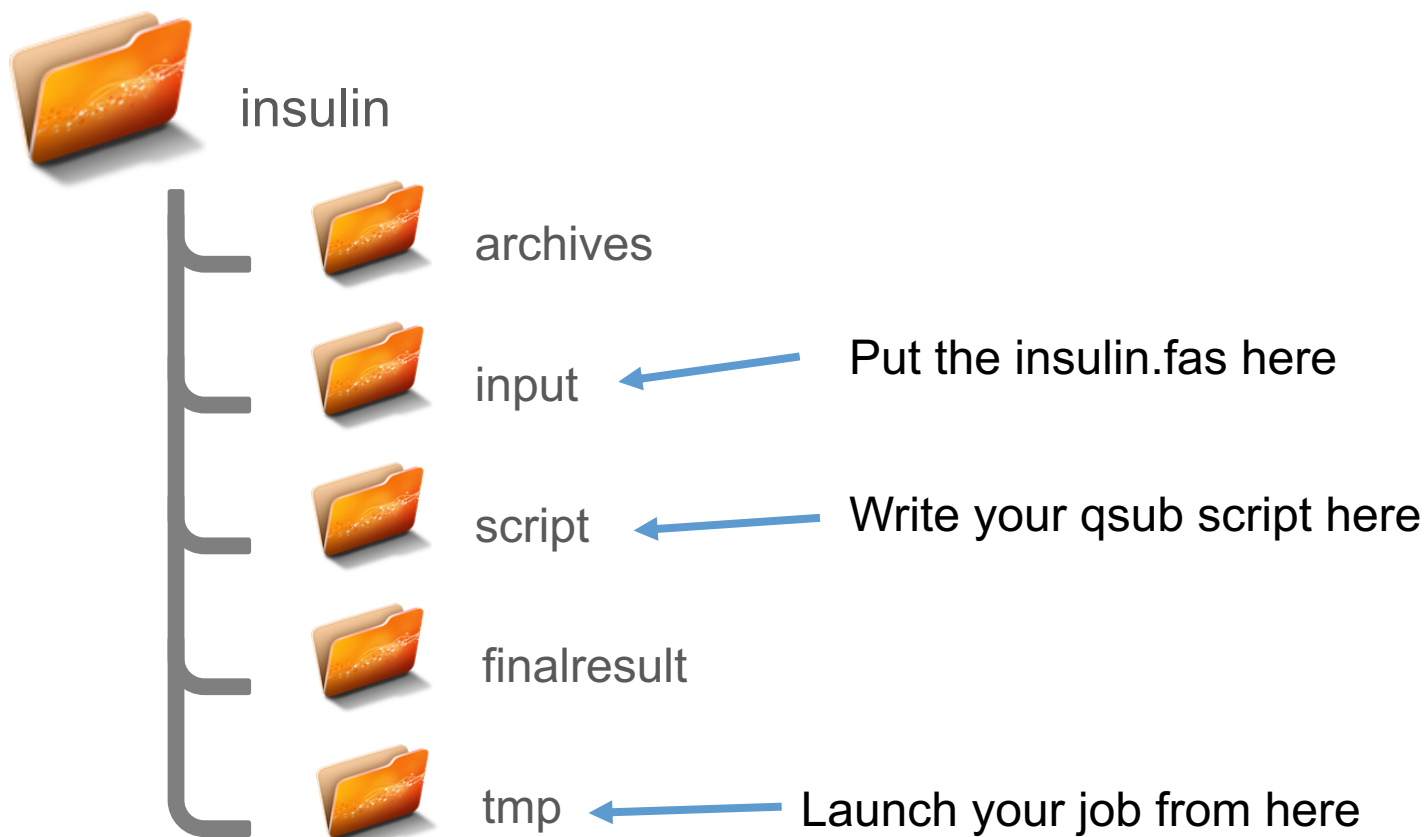
EXERCICE / EXAMPLES



Exercises/Examples



Search for sequence similarities using *blastn* on the fasta file *insulin.fas* again the database *nt*

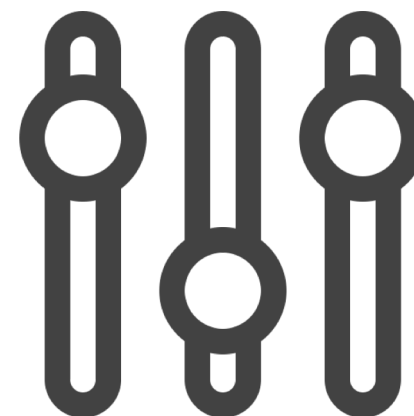




Search for sequence similarities using *blastn* on the fasta file *insulin.fas* against the database *nt*

Parameters:

- query insulin.fas
- outfmt 6
- evaluate 1e-6
- max_target_seqs 5
- db /db/blast/all/nt
- out insulin_nt.blastn.tab





Using:

1. qlogin
2. qsub: Simple script
3. qsub: Multithread script
4. qsub: Job-array



1. qlogin



1. Open a connection using `qlogin` on a node

```
$ qlogin -q qlogin.q
```

```
Your job 2217414 ("QLOGIN") has been submitted  
waiting for interactive job to be scheduled ...  
Your interactive job 2217414 has been successfully scheduled.  
Establishing /opt/sge/qlogin.sh session to host n78 ...  
Last login: Mon Apr 15 10:22:01 2013 from n0.sb-roscoff.fr
```

```
@n78$ cdprojet  
@n78$ cd tp-cluster/tmp  
@n78$ . $CONDA3/activate blast-2.7.1  
@n78$ blastn -help  
@n78$ blastn -query ...
```

2. qsub: simple script



1. Edit a text file using gedit or vim

```
$ gedit script/blastn.qsub
```

```
$ vim script/blastn.qsub
```

2. Add settings for SGE

```
#!/bin/bash  
#$ -S /bin/bash  
#$ -V  
#$ -cwd  
#$ -M foo.bar@sb-roscoff.fr  
#$ -m bea
```

2. qsub: simple script



3. Append with the command line

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -M foo.bar@sb-roscoff.fr
#$ -m bea
#$ -q short.q

INPUT="../input/insulin.fas"
OUTPUT="insulin.blast"
DATABASE="/db/blast/all/nt"

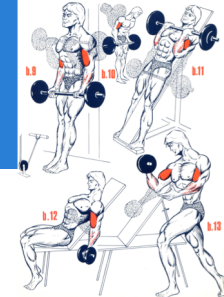
blastn -query $INPUT -db $DATABASE -out $OUTPUT -outfmt 6 -evaluate 1e-6 -max_target_seqs 5
```

4. Launch a qsub request in the terminal

```
$ qsub ../script/blastn.qsub
```

```
Your job 744348 ("blastn.qsub") has been submitted
```

2. qsub: simple script



5. Monitor your job (quickly)

- Results in a other terminal
- Cluster state
- Jobs running

```
$ \qstat
```

```
744348 1.00000 blastn.qsu fbar      qw    05/13/2016 08:58:06      1
```

```
$ \qstat
```

```
744348 1.00000 blastn.qsu fbar      r     05/13/2016 08:58:06 short.q@n78.sb-roscoff.fr 1
```

```
$ ll
```

```

-rw-r--r-- 1 fbar grp      474 mai 31 09:53 blastn.qsub
-rw-r--r-- 1 fbar grp        0 mai 31 09:53 blastn.qsub.e744348
-rw-r--r-- 1 fbar grp    1538 mai 31 09:53 blastn.qsub.o744348

```

3. qsub: Multithread script



2. Add settings for SGE

3. for blastn

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -M foo.bar@sb-roscoff.fr
#$ -m bea
#$ -pe thread 2
#$ -q short.q
```

```
INPUT=" ../input/insulin.fas"
OUTPUT="insulin.blast"
DATABASE="/db/blast/all/nt"
```

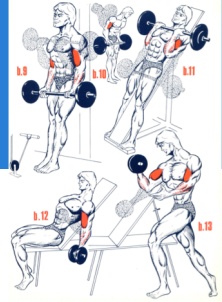
```
blastn -query $INPUT -db $DATABASE -out $OUTPUT -outfmt 6 -evaluate 1e-6 -max_target_seqs 5
-num_threads $NSLOTS
```

4. Launch

```
$ qsub blastn.qsub
```

```
Your job 744349 ("blastn.qsub") has been submitted
```

2. qsub: simple script



5. Monitor your job (quickly)

```

$ \qstat
744349 1.00000 blastn.qsu fbar      qw    05/13/2016 08:58:06                2
$ \qstat
744349 1.00000 blastn.qsu fbar      r     05/13/2016 08:58:06 short.q@n78.sb-roscoff.fr 2
    
```

- Job-array: the idea is to split a job in a large number of sub-job
 - Very high load on the cluster
 - Problem with slot reservation
 - To prevent this problem:

Job-array should be run only on short.q!

4. qsub: Job-array

Problem: a large number of jobs to run and they are largely identical in terms of the command to run.

For example, you may have 1000 data sets, and you want to run a single program on each of them.

Naive solution: generate 1000 shell scripts, and submit them to the cluster.

Best solution: on SGE systems – array jobs. The advantages are:

→ You only have to write one shell script

*One Script to rule them all, One Script to find them,
One Script to bring them all and in the darkness bind them*

4. qsub: Job-array

```
$ ls *.fq
```

488_albifrons_bret.fq	493_albifrons_bret.fq	577_praehirsuta_bret.fq
584_praehirsuta_bret.fq	594_hybride_norm.fq	703_praehirsuta_norm.fq
714_albifrons_norm.fq	724_praehirsuta_norm.fq	490_albifrons_bret.fq
570_praehirsuta_bret.fq	580_praehirsuta_bret.fq	587_hybride_norm.fq
595_hybride_norm.fq	707_praehirsuta_norm.fq	719_albifrons_norm.fq

1. Create the structure

```
#!/bin/bash  
#$ -S /bin/bash  
#$ -V  
#$ -cwd  
#$ -short.q
```

```
ustacks -f $INPUT -o . -m 2
```

4. qsub: Job-array

```
$ ls *.fq
```

488_albifrons_bret.fq	493_albifrons_bret.fq	577_praehirsuta_bret.fq
584_praehirsuta_bret.fq	594_hybride_norm.fq	703_praehirsuta_norm.fq
714_albifrons_norm.fq	724_praehirsuta_norm.fq	490_albifrons_bret.fq
570_praehirsuta_bret.fq	580_praehirsuta_bret.fq	587_hybride_norm.fq
595_hybride_norm.fq	707_praehirsuta_norm.fq	719_albifrons_norm.fq

2. Get the n^{th} **INPUT** / the **SGE_TASK_ID**th

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -short.q
```

```
INPUT=$(ls *.fq | awk "NR==$SGE_TASK_ID")
```

```
ustacks -f $INPUT -o . -m 2
```

- **NR** = Number of Row
- **\$ ()** means launch the command and get back the result

4. qsub: Job-array

```
$ ls *.fq
```

488_albifrons_bret.fq	493_albifrons_bret.fq	577_praehirsuta_bret.fq
584_praehirsuta_bret.fq	594_hybride_norm.fq	703_praehirsuta_norm.fq
714_albifrons_norm.fq	724_praehirsuta_norm.fq	490_albifrons_bret.fq
570_praehirsuta_bret.fq	580_praehirsuta_bret.fq	587_hybride_norm.fq
595_hybride_norm.fq	707_praehirsuta_norm.fq	719_albifrons_norm.fq

3. Set the `SGE_TASK_ID` range

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -short.q
#$ -t 1-42
```

```
INPUT=$(ls *.fq | awk "NR==$SGE_TASK_ID")
```

```
ustacks -f $INPUT -o . -m 2
```

- -t 1-1000
- -t 250-500

4. qsub: Job-array

```
$ \qstat
```

job-ID	prior	name	user	state	submit/start at	queue	slots	ja-task-ID
2216309	0.00045	blastn.qsu	foobar	r	05/23/2013 14:19:56	short.q@n78	1	64
2216309	0.00045	blastn.qsu	foobar	r	05/23/2013 14:22:12	short.q@n43	1	65
2216309	0.00045	blastn.qsu	foobar	r	05/23/2013 14:25:03	short.q@n41	1	66
2216309	0.00045	blastn.qsu	foobar	r	05/23/2013 14:35:05	short.q@n78	1	67
2216309	0.00045	blastn.qsu	foobar	r	05/23/2013 14:40:02	short.q@n77	1	68
2216309	0.00045	blastn.qsu	foobar	qw	05/23/2013 14:43:03		1	69-2379:1

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -short.q
#$ -t 1-42
```

```
INPUT=$(ls *.fq | awk "NR==$SGE_TASK_ID")
```

```
ustacks -f $INPUT -o . -m 2
```

- -t 1-1000
- -t 250-500

4. qsub: Job-array

```
$ ls *.fq
```

488_albifrons_bret.fq	493_albifrons_bret.fq	577_praehirsuta_bret.fq
584_praehirsuta_bret.fq	594_hybride_norm.fq	703_praehirsuta_norm.fq
714_albifrons_norm.fq	724_praehirsuta_norm.fq	490_albifrons_bret.fq
570_praehirsuta_bret.fq	580_praehirsuta_bret.fq	587_hybride_norm.fq
595_hybride_norm.fq	707_praehirsuta_norm.fq	719_albifrons_norm.fq

4. [optional] manage the stdout and stderr

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -short.q
#$ -t 1-42
```

```
INPUT=$(ls *.fq | awk "NR==$SGE_TASK_ID")
```

```
ustacks -f $INPUT -o . -m 2 > $INPUT.ustacks.stdout 2> $INPUT.ustacks.stderr
```

4. qsub: Job-array

```
$ ls *.fq
```

488_albifrons_bret.fq	493_albifrons_bret.fq	577_praehirsuta_bret.fq
584_praehirsuta_bret.fq	594_hybride_norm.fq	703_praehirsuta_norm.fq
714_albifrons_norm.fq	724_praehirsuta_norm.fq	490_albifrons_bret.fq
570_praehirsuta_bret.fq	580_praehirsuta_bret.fq	587_hybride_norm.fq
595_hybride_norm.fq	707_praehirsuta_norm.fq	719_albifrons_norm.fq

5. [optional] save a link table

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -short.q
#$ -t 1-42
```

```
INPUT=$(ls *.fq | awk "NR==$SGE_TASK_ID")
```

```
echo -e $(date '+%y%m%d-%H:%M') "\t"$SGE_TASK_ID"\t"$INPUT >> qsub_array_files.tab
```

```
ustacks -f $INPUT -o . -m 2 > $INPUT.ustacks.stdout 2> $INPUT.ustacks.stderr
```

4. qsub: Job-array

```
$ ls *.fq
```

488_albifrons_bret.fq	493_albifrons_bret.fq	577_praehirsuta_bret.fq
584_praehirsuta_bret.fq	594_hybride_norm.fq	703_praehirsuta_norm.fq
714_albifrons_norm.fq	724_praehirsuta_norm.fq	490_albifrons_bret.fq
570_praehirsuta_bret.fq	580_praehirsuta_bret.fq	587_hybride_norm.fq
595_hybride_norm.fq	707_praehirsuta_norm.fq	719_albifrons_norm.fq

5. [optional] limit the number of running jobs

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -short.q
#$ -t 1-42
#$ -tc 5
```

```
INPUT=$(ls *.fq | awk "NR==$SGE_TASK_ID")
```

```
echo -e $(date '+%y%m%d-%H:%M') "\t"$SGE_TASK_ID"\t"$INPUT >> qsub_array_files.tab
```

```
ustacks -f $INPUT -o . -m 2 > $INPUT.ustacks.stdout 2> $INPUT.ustacks.stderr
```


4. qsub: Job-array

```
$ ls *.fq
```

488_albifrons_bret.fq	493_albifrons_bret.fq	577_praehirsuta_bret.fq
584_praehirsuta_bret.fq	594_hybride_norm.fq	703_praehirsuta_norm.fq
714_albifrons_norm.fq	724_praehirsuta_norm.fq	490_albifrons_bret.fq
570_praehirsuta_bret.fq	580_praehirsuta_bret.fq	587_hybride_norm.fq
595_hybride_norm.fq	707_praehirsuta_norm.fq	719_albifrons_norm.fq

5. [optional] Super Saiyan transformation



```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -short.q
#$ -tc 5
```

```
INPUT=$(ls *.fq | awk "NR==$SGE_TASK_ID")
```

```
echo -e $(date '+%y%m%d-%H:%M') "\t"$SGE_TASK_ID"\t"$INPUT >> qsub_array_files.tab
```

```
ustacks -f $INPUT -o . -m 2 > $INPUT.ustacks.stdout 2> $INPUT.ustacks.stderr
```

```
#qsub -t 1-$(ls *.fq | wc -l) ustacks_array.qsub
```

```
$ qsub -t 1-$(ls *.fq | wc -l) ustacks_array.qsub
```

4. qsub: Job-array: Atomicblastplus.py

Standard blast: weak performances with big dataset against huge databases (nr, nt,...)

Solution: splitting your set of sequences in order to create a job-array:

- A way to parallelized blast

```
$ atomicblastplus -p blastn -i input.fasta -o myproject/tmp/blast/input_vs_nr -d  
/db/blast/all/nt -e 1e-4 -n 100 --verbose
```

```
!!! This is Atomic Blast !!!
```

```
PROGRAM:  
blastn: 2.2.28+  
Package: blast 2.2.28, build Mar 12 2013 16:52:31
```

```
QUERY: insulin.fasta  
DB: /db/blast/all/nt  
OUTDIR: test
```

```
INFO: The query was splitted into 1 subfiles
```

```
INFO: SGE qsub script was written to test/qsub.insulin.atomic_blastn_vs_nt.sh
```

```
INFO: Running job-array on SGE...
```

```
CMD: qsub -q short.q -t 1-1 -tc 100 -sync yes -N at_blastn_insulin.atomic_blastn_vs_nt  
test/qsub.insulin.atomic_blastn_vs_nt.sh
```

```
$ atomicblastplus -p blastn -i input.fasta -o myproject/tmp/blast/input_vs_nr -d  
/db/blast/all/nt -e 1e-4 -n 10 --dont_wait --verbose
```

4. qsub: Job-array: Atomicblastplus.py

Standard blast: weak performances with big dataset against huge databases (nr, nt,...)

Solution: splitting your set of sequences in order to create a job-array:

- A way to parallelized blast

```
$ atomicblastplus -p blastn -i input.fasta -o myproject/tmp/blast/input_vs_nr -d
/db/blast/all/nt -e 1e-4 -n 100 --verbose
```

```
!!! This is Atomic Blast !!!
```

```
PROGRAM:
```

```
blastn: 2.2.28+
```

```
Package: blast 2.2.28, build Mar 12 2015
```

```
QUERY: insulin.fasta
```

```
DB: /db/blast/all/nt
```

```
OUTDIR: test
```

```
INFO: The query was splitted into 1 sequences
```

```
INFO: SGE qsub script was written to test/qsub.insulin.atomic_blastn_vs_nr
```

```
INFO: Running job-array on SGE...
```

```
CMD: qsub -q short.q -t 1-1 -tc 100 -o test/qsub.insulin.atomic_blastn_vs_nr
```

```
test/qsub.insulin.atomic_blastn_vs_nr
```

**Do not split your input
file in 10000 files
It's useless and
resource consuming**

```
$ atomicblastplus -p blastn -i input.fasta -o myproject/tmp/blast/input_vs_nr -d
/db/blast/all/nt -e 1e-4 -n 10 --dont_wait --verbose
```

Optional

ADVANCED PRACTICE

- Pass arguments to a qsub script

```
#!/bin/bash  
#$ -S /bin/bash  
#$ -V  
#$ -cwd  
#$ -short.q
```

```
ustacks -f $1 -o . -m $2 -M $3 > $1.ustacks.stdout 2> $1.ustacks.stderr
```

```
$ qsub ustacks.qsub 488_albifrons_bret.fq 10 5
```

- Pass arguments to a qsub script

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -short.q
```

```
ustacks -f $1 -o . -m $m -M $M > $1.ustacks.stdout 2> $1.ustacks.stderr
```

```
$ qsub -v m=10,M=5 ustacks.qsub 488_albifrons_bret.fq
```

- No loop

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -short.q
```

```
INPUT="488_albifrons_bret.fq"
```

```
ustacks -f $INPUT -o . -m $m -M $M > $INPUT.ustacks.stdout 2> $INPUT.ustacks.stderr
```

```
$ qsub -v m=10,M=5 ustacks.qsub
```

- For loop + loop

```
#!/bin/bash  
#$ -S /bin/bash  
#$ -V  
#$ -cwd  
#$ -short.q
```

```
INPUT="488_albifrons_bret.fq"
```

```
ustacks -f $INPUT -o . -m $m -M $M > $INPUT.ustacks.stdout 2> $INPUT.ustacks.stderr
```

```
$ for m in $(seq 1 10); do qsub -v m=$m,M=5 ustacks.qsub; done
```

– Number of job = 10

- For loop + loop

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -short.q
```

```
INPUT="488_albifrons_bret.fq"
```

```
ustacks -f $INPUT -o . -m $m -M $M > $INPUT.ustacks.stdout 2> $INPUT.ustacks.stderr
```

```
$ for m in $(seq 1 10); do for M in "1 3 5"; do qsub -v m=$m,M=$M ustacks.qsub; done ; done
```

– Number of job = 10 x 3 = 30

- For loop + loop + job_array

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -short.q
#$ -t 1-200
#$ -tc 20
INPUT=$(ls *.fq | awk "NR==$SGE_TASK_ID")
ustacks -f $INPUT -o . -m $m -M $M > $INPUT.ustacks.stdout 2> $INPUT.ustacks.stderr
```

```
$ for m in $(seq 1 10); do for M in "1 3 5"; do qsub -v m=$m,M=$M ustacks.qsub; done ; done
```

– Number of job = 10 x 3 x 200 = 6000

- For loop + job control

```
#!/bin/bash
#$ -S /bin/b
#$ -V
#$ -cwd
#$ -short.g
#$ -t 1-200
```

```
INPUT=$(ls
a ==$ SK
ustacks -f -c $m > /dev/null; INP ack err
```

```
$ for m in $(seq 1 200); do "1 3 5"; do $m, M=ack; done ; done
```

– Number of jobs = 200

A black and white photograph of a city skyline at night, likely New York City, with the text "The End" overlaid in a large, white, cursive font. The skyline features several prominent skyscrapers, including the Empire State Building, which is illuminated and stands out against the dark sky. The foreground is dark and out of focus, showing some lights and structures. The overall mood is dramatic and final.

The End