

UiO : **Department of Biosciences**
University of Oslo

Quality control

Produce contigs

Demultiplexing

Dereplication

OTU construction

Chimera checking

Taxonomic annotation

Removal of non-target organisms

Cleaning of tag bleeding

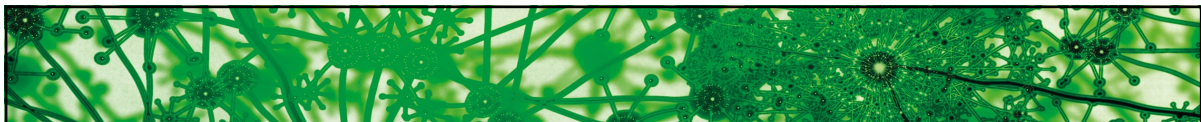
OTU modifications

Positive negatives

Singleton removal

Data transformation (e.g.
rarification)

2



UiO : **Department of Biosciences**
University of Oslo

Introduction to Unix and Linux

Anders K. Krabberød
a.k.krabberod@ibv.uio.no



3

Why learn about UNIX and Linux?

- Bioinformatics tools and algorithms are very often developed for **UNIX/Linux** computers.
- Easy to set up **pipelines** and automate workflows.
- Powerful for handling **text** files, large files, and large number of files – very common in modern biology.
- Able to work on High Performance Computing platforms (Supercomputers) e.g. Saga

4

HPC (High Performance Computing)

- In Norway Sigma2 runs several HPC (High Performance Computing) clusters as well as other e-infrastructure facilities. <https://www.sigma2.no/services-overview>
- Current clusters
 - **Saga**
 - Fram
 - Vilje
 - Stallo
 - Betzy

5

Saga

- Online as of fall 2019
- A computing cluster with 200 standard compute nodes, with 40 cores and 192 GiB memory each,
- Total 9824 cores
- Access to resources is managed with a queue system: **SLURM**
- Runs linux: CentOS

Details	Saga
System	Hewlett Packard Enterprise - Apollo 2000/6500 Gen10
Number of Cores	9824
Number of nodes	244
Number of GPUs	32
CPU type	Intel Xeon-Gold 6138 2.0 GHz (normal) Intel Xeon-Gold 6130 2.1 GHz (bigmem) Intel Xeon-Gold 6126 2.6 GHz (accel)
GPU type	NVIDIA P100, 16 GiB RAM (accel)
Total max floating point performance, double	645 Teraflop/s (CPUs) + 150 Teraflop/s (GPUs)
Total memory	75 TiB
Total NVMe+SSD local disc	89 TiB + 60 TiB
Total parallel filesystem capacity	1 PB

6

The UNIX philosophy

- The success of the UNIX framework is because of the philosophy behind the development
 - Write programs that do **one** thing and **do it well**.
 - Write programs that will **work together**.
 - Write programs to handle **text** streams, because that is a universal interface.

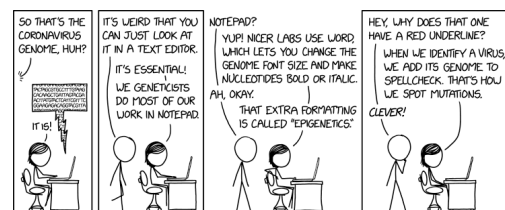
7

- Advantages with Linux
 - Stability
 - Security
 - Good control of users and processes
 - Can handle large computing load
 - Open Source (you can see and modify the code)
 - Several free options (Ubuntu, CentOS, Debian, etc.)
 - Large community of developers
 - Large amount of free software (specially for bioinfo)
- Disadvantages
 - Some hardware may not be supported
 - Can be less user friendly than other options
 - (For CLI) It's not point-and-click

8

Demystifying files

- Many files used in bioinformatics are just **text** files.
- Extension given to the file reflects the format, or how the "internal layout" of that file is.
- Often the format is formulated as set of rules that tells the user what the file should contain and how it should be organised



```
>Seq1
ATGGTATGACCATGT
>Seq2
ATGGTGGCTTGCTG
```

9

Demystifying files - examples

- **Fasta**
 - Lines starting with > is the sequence identifier
 - Lines that follow *without* > is the sequence (DNA, RNA or Amino Acids)
- **Fastq** -
 - Sequence with quality score
 - 4 lines per sequencecne
 - Line 1 begins with a '@' followed by header
 - Line 2 is the sequence
 - Line 3 begins with a '+' and is *optionally* followed by the sequence identifier and description
 - Line 4 encodes the quality values for the sequence in Line 2, and must contain the same number of symbols as letters in the sequence.

```
>First_sequence_id
ATGGTATGACCATGT
>Next_sequence_id
ATGGTGGCTTGCTGGGCGGGGT
```

```
@First_sequence_id
ATGGTATGACCATGT
+
! '*' ((( (***+)) %1
@Next_sequence_id
ATGGTGGCTTGCTGGGCGGGGT
+
%%) .1***-+*'') **55) KM
```

10

The Shell / Terminal

```
bash-3.25 $ ls -lh
total 877496
drwxr-xr-x 6 anderkr staff 192B Apr 28 15:07 Applications
-rw-r--r-- 1 anderkr staff 365K Apr 2 11:11 Arabidopsis.tgz
drwxr-xr-x 4 anderkr staff 128B Aug 28 2019 Boostnote
drwxr-xr-x 53 anderkr staff 1.7K Feb 21 12:21 CytoscapeConfiguration
-rw-r--r-- 1 anderkr staff 53K Feb 19 19:41 dendroscope_macos_3_7_2.dmg
drwxr-xr-x 14 anderkr staff 448B May 2 14:02 Desktop
drwxr-xr-x 23 anderkr staff 736B May 2 14:01 Documents
drwxr-xr-x 27 anderkr staff 864B May 1 20:06 Downloads
drwxr-xr-x 39 anderkr staff 1.2K May 1 09:24 Dropbox
drwxr-xr-x 33 anderkr staff 1.0K Apr 29 16:19 Genetious 2019.0 Data
drwxr-xr-x 80 anderkr staff 2.5K Mar 6 11:58 Library
drwxr-xr-x 5 anderkr staff 160B Apr 27 2019 Movies
drwxr-xr-x 11 anderkr staff 352B Oct 1 2019 Music
drwxr-xr-x 5 anderkr staff 160B Apr 27 13:56 Pictures
-rw-r--r-- 1 anderkr staff 11M Apr 23 2019 Pr
drwxr-xr-x 5 anderkr staff 160B Apr 25 23:44 Public
-rw-r--r-- 1 anderkr staff 736B Sep 29 2019 Untitled.ipynb
drwxr-xr-x 16 anderkr staff 512B Aug 29 2019 Zotero
drwxr-xr-x 2 anderkr staff 64B Mar 17 14:44 alyrata_pet_assembly1
-rw-r--r-- 1 anderkr staff 733B Dec 10 2018 derby.log
drwxr-xr-x 13 anderkr staff 416B Jan 18 2017 gganimate
drwxr-xr-x 5 anderkr staff 160B Nov 2 21:11 github
drwxr-xr-x 22 anderkr staff 704B Jan 5 18:38 iCloud Drive (Archive)
drwxr-xr-x 3 anderkr staff 96B Feb 3 09:36 iCloud Drive (Archive) - 1
drwxr-xr-x 6 anderkr staff 192B Jan 3 17:55 ivg
drwxr-xr-x 3 anderkr staff 96B May 20 2018 isus
drwxr-xr-x 2 anderkr staff 64B Oct 16 2019 notes
drwxr-xr-x 5 anderkr staff 160B Feb 2 2017 test
-rw-r--r-- 1 anderkr staff 988B Apr 2 11:11 wget-log
bash-3.25 $
```

- The user writes commands on the command line usually in a terminal window
- Line-by-line
- Step-by-step
- Logical work flow

Linux is case-sensitive!

11

The Shell

- Getting a shell
- **Windows:** Windows Terminal
 - <https://github.com/microsoft/terminal>
- **Mac and Linux:** A shell is already installed (launch the Terminal application).

12

- If you've never used the terminal, you can start with simple commands and a good amount of patience 😊
- There's plenty of resources and documentation online
 - Google is your friend!
- A set of instructions and exercises made for the course 3010 will help you:
- <https://github.com/krabberod/bios3010-unix-linux>

13

Login into Saga

- Open terminal
- Type `ssh <username>@saga.sigma2.no`
- Type password

```
(base) [anderkkr~]$ ssh anderkkr@saga.sigma2.no
anderkkr@saga.sigma2.no's password:
Last login: Fri May 1 19:13:14 2020 from 84.211.16.85

Welcome to saga.sigma2.no!

Documentation: https://documentation.sigma2.no/
Support email: support@metacenter.no
Request resources: https://www.sigma2.no/content/apply-e-infrastructure-resources/

Latest news from: https://opslog.sigma2.no/
o 2020-04-29: Stallo problems / urgent maintenance
o 2020-04-27: Stallo RAM upgrade
o 2020-04-24: Downtime 20th 24th of April is over. Services are back in prod
o 2020-04-16: Stallo slurm problem
o 2020-04-08: tos-project3 on NIRD is read only

WARNING: Please save all your important files.
Backup is taken "only" for $HOME which has quota in place.
For more info see: https://documentation.sigma2.no/storage/backup.html

-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[anderkkr@login-1.SAGA ~]$
```

14

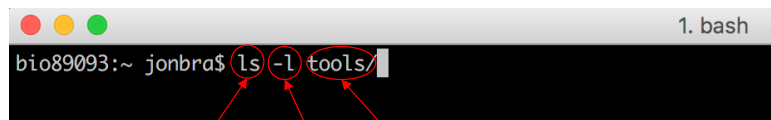
General comments on commands

- The commands operates on **text**. Either files containing text or input from the keyboard (called *standard input*).
- Commands in Linux usually takes the form
`command [-options] inputfile`
- The philosophy of Linux (and UNIX) programs is that they are supposed to be simple and efficient with a standardised way of taking inputs and giving outputs. Because of the standardisation, the output of one command can (often) be used as the input of another command. The advantage of this will be clear later.
- The full manual for a command can be accessed with `man command`

15

Unix command syntax

command [-option(s)] [argument(s)]



```
bio89093:~ jonbra$ ls -l tools/
```

command

option

argument

There must be spaces
between commands,
options and arguments

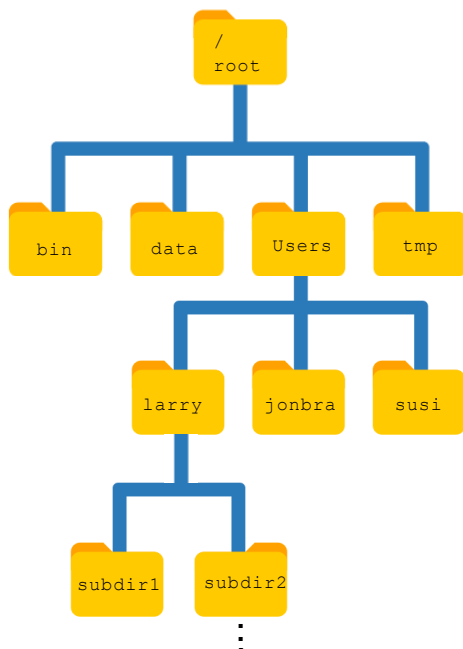
16

Navigating the filesystem

- `cd` changes directory
- `ls` lists the files and directories in the current directory
- `pwd` prints the path of the current directory
- `~/` or `$HOME` is your home directory (`cd` will take you there)
- `.` is your current directory
- `..` is the directory directly above the current

<http://swcarpentry.github.io/shell-novice>

17



- To make a new directory: `mkdir`
 - `mkdir larry`
- To make subdirectories
 - `mkdir larry/subdir1`
 - `mkdir larry/subdir2`
 - `mkdir larry/subdir2/subdir1`
- To navigate use `cd` (change directory)
 - `cd larry`
 - `cd ..` (takes you one level up)
 - `cd .` (you stay where you are)
 - `cd` (shortcut directly to your home directory)
- To find out where you are, use `pwd` (print working directory)
- To list the files and directories in the current directory use `ls`

18

Some of the most useful commands which will be covered in the exercises

- | | |
|--|--|
| <ul style="list-style-type: none"> • <code>ssh</code> - Secure log on • <code>ls</code> - List content • <code>pwd</code> - Print the working directory • <code>cd</code> - change directory • <code>mkdir</code> - make directory • <code>cp</code> - Copy files and directories • <code>scp</code> - Secure copying of files and directories • <code>rsync</code> - Synchronise and transfer files and directories • <code>rm</code> - Remove files and directories • <code>less</code> - Print content of file to screen • <code>more</code> - Print content of file to screen • <code>head</code> - Print the head of file(s) • <code>tail</code> - Print the end (tail) of file(s) | <ul style="list-style-type: none"> • <code>cat</code> - Print and concatenate several files • <code>grep</code> - Search through text files • <code>sort</code> - Sort the content of files • <code>uniq</code> - Print unique (lines, or words) <p>Special operators:</p> <ul style="list-style-type: none"> • <code>></code> (IO redirection) • <code>>></code> (IO append) • <code> </code> (the vertical line is called <i>pipe</i>) <p>Useful cheat sheet:
 https://cheatography.com/davechild/cheat-sheets/linux-command-line/ </p> |
|--|--|

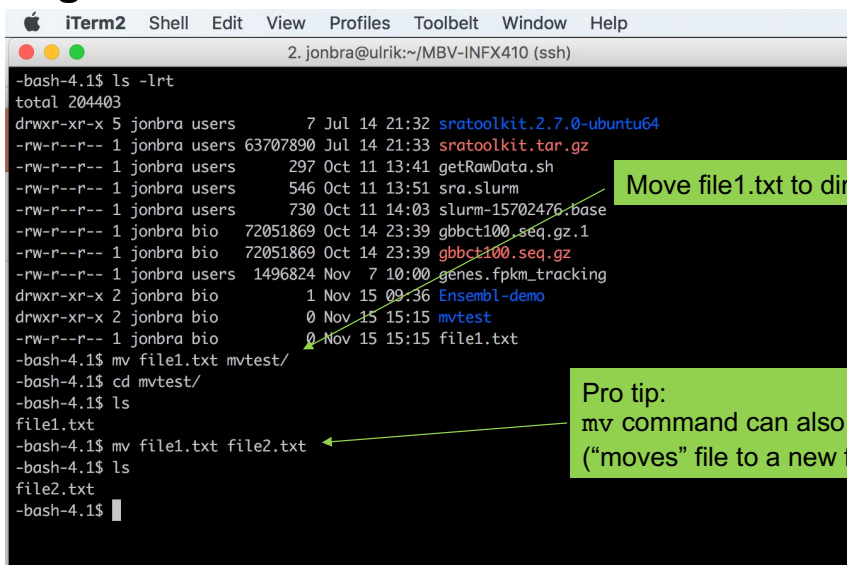
19

Remember

- Linux commands are case sensitive!
 - `ssh`
 - is **NOT** the same as
 - `Ssh`
- Filenames and paths are also case sensitive!
 - `huge_file.fasta`
 - is **NOT** the same as:
 - `Huge_file.fasta`
- Protip: Use the tab-key to autocomplete

20

Moving files with the `mv` command



```

-iTerm2  Shell  Edit  View  Profiles  Toolbelt  Window  Help
2. jonbra@ulrik:~/MBV-INF410 (ssh)

-bash-4.1$ ls -lrt
total 204403
drwxr-xr-x 5 jonbra users      7 Jul 14 21:32 sratoolkit.2.7.0-ubuntu64
-rw-r--r-- 1 jonbra users 63707890 Jul 14 21:33 sratoolkit.tar.gz
-rw-r--r-- 1 jonbra users   297 Oct 11 13:41 getRawData.sh
-rw-r--r-- 1 jonbra users    546 Oct 11 13:51 sra.slurm
-rw-r--r-- 1 jonbra users    730 Oct 11 14:03 slurm-15702476-base
-rw-r--r-- 1 jonbra bio 72051869 Oct 14 23:39 gbbct100.seq.gz.1
-rw-r--r-- 1 jonbra bio 72051869 Oct 14 23:39 gbbct100.seq.gz
-rw-r--r-- 1 jonbra users 1496824 Nov  7 10:00 genes.fpk_tracking
drwxr-xr-x 2 jonbra bio      1 Nov 15 09:36 Ensembl-demo
drwxr-xr-x 2 jonbra bio      0 Nov 15 15:15 mvtest
-rw-r--r-- 1 jonbra bio      0 Nov 15 15:15 file1.txt

-bash-4.1$ mv file1.txt mvtest/
-bash-4.1$ cd mvtest/
-bash-4.1$ ls
file1.txt
-bash-4.1$ mv file1.txt file2.txt
-bash-4.1$ ls
file2.txt
-bash-4.1$

```

Move file1.txt to directory mvtest

Pro tip:
mv command can also rename files!
("moves" file to a new filename)

21

Delete files with the **rm** command

```
iTerm2 Shell Edit View Profiles Too
-bash-4.1$ ls
file1.txt file2.txt file3.txt
-bash-4.1$ rm file1.txt
-bash-4.1$ ls
file2.txt file3.txt
-bash-4.1$ rm file*
-bash-4.1$ ls
-bash-4.1$
```

NB! There's no trash can on Unix!
Your files are immediately deleted!

Delete empty directories with **rm -r**

Delete directories AND
files within with **rm -rf** Use with
caution!! (**rm -i**)

```
iTerm2 Shell Edit View Profiles Toolbelt Window Help
2. jonbra@u
-bash-4.1$ ls -lrt
total 63677
drwxr-xr-x 5 jonbra users 7 Jul 14 21:32 sratoolkit.2.7.0-ubuntu64
-rw-r--r-- 1 jonbra users 63707890 Jul 14 21:33 sratoolkit.tar.gz
-rw-r--r-- 1 jonbra users 297 Oct 11 13:41 getRawData.sh
-rw-r--r-- 1 jonbra users 546 Oct 11 13:51 sra.slurm
-rw-r--r-- 1 jonbra users 730 Oct 11 14:03 slurm-15702476.base
-rw-r--r-- 1 jonbra users 1496824 Nov 7 10:00 genes.fpkms_tracking
drwxr-xr-x 2 jonbra bio 1 Nov 15 09:36 Ensembl-demo
drwxr-xr-x 2 jonbra bio 0 Nov 15 14:44 delete
-bash-4.1$ rm -r delete/
```

22

Viewing files

Pro tip:
Use the different
options to tweak the output

- **cat** displays the entire content on the screen
- **less** displays the content on the screen one file/line at a time (press "q" to exit).
- **head/tail** displays the first or last lines of a file

```
iTerm2 Shell Edit View Profiles Toolbelt W
-bash-4.1$ head -4 gbct100.seq
GBCT100.SEQ Genetic Sequence Data Bank
October 15 2016
NCBI-GenBank Flat File Release 216.0
-bash-4.1$
```

```
iTerm2 Shell Edit View Profiles Toolbelt Window Help
2. jonbra@ulrik:~/M
-bash-4.1$ tail -9 gbct100.seq
6542881 ctttgcgc cgtaccac gatccccg ccagccagg ttcttggt atggcgaa
6542941 gattcgca tcagcggtt gaggcatt cgtcgacc cgagcttg gttctctt
6543001 cgtaccac ggcagcgg ggggtgat agaccgtt cgcgcacc cagcagac
6543061 tcttgccg ggccttgc gatcgagg tcgaagact cgtgattg ccggctgc
6543121 agtagact ttctcggt gaagtcga ctaccacc gtaccgga atcagcgg
6543181 ttagacgtt acgctctt gctgcgac ggcagaga ctgacgac gttctggt
6543241 ccatagcg ccggaacc tgggcgac cgcgttgg ggtctggg tgggaagt
6543301 gttctgat tgcgtact ggttcgac ttgagtcg agtgaccg
//
-bash-4.1$
```

23

Transfer files with scp

- `scp` is used to copy files from one machine to a remote machine over a secure SSH connection.
- General structure:

`scp <from> <to>`

- Example a files sent to saga:

`scp file.fasta username@saga.sigma2.no:/cluster/home/username/`

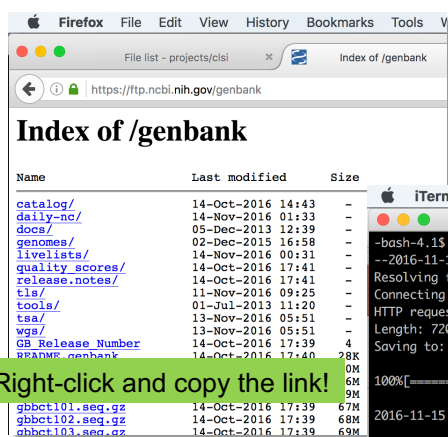
command

file to be sent

where to send it
username@location:path

24

Downloading files from the internet



The `wget` command make it possible to download files from the web



25

Redirect and pipe

- Since Linux (and UNIX) use streams of text it is easy to
- write the output of a command to a file.
 - Two versions:
 - The **greater than symbol** (>) is used to write a new files (will overwrite)


```
ls -lh > list_of_files.txt
```
 - Two symbols (>>) will append the content to a file


```
ls -lh >> list_of_files.txt
```

26

Redirect and pipe

- Since Linux (and UNIX) use streams of text it is easy to
- Use the output of one command as input in another with the horizontal line called pipe |

```
wc -l *fasta | sort -n
```

First command

Pipe

Second command

27

```
$ wc -l *.pdb
```

```
wc -l *.pdb
```

OUT



```
$ wc -l *.pdb > lengths
```

```
wc -l *.pdb
```

OUT



```
$ wc -l *.pdb | sort -n | head -1
```

```
wc -l *.pdb
```

OUT

IN

```
sort -n
```

OUT

IN

```
head -1
```

OUT



28

- Many of the commands takes extra arguments
- `*` = a wildcard!
- `ls -lh start_of_filena*`
- `cd (home)` ; `cd dir` (go to dir); `cd ..` (one level up, to the parent directory); `cd ../../` (two levels up)
- `mv old_name new_name`
- `mv /old/location/for/my/file /new/location/for/my/file`
- `cp /path/to/file/to/copy/file /new/location/of/file`
- `cp -r /dir/to/copy/dir /location/for/dir/copy`
- `rm file`
- `rm -rf /remove/directory/and/all/contents` (use carefully!)
- `cat file1 file2 > newcatfile`
- `paste file1 file2 > newpastefile`
- `chmod a+x executablefile`
- `chmod 755 file` (owner:group:world permissions)

29

- A few other very useful commands:
- `tail -n 300 file`
 - (see last 300 lines file)
- `head -n 300 file`
 - (see the first 300 lines of a file)
- `grep pattern file`
 - (search pattern in file)
- `grep -c pattern file`
 - (count pattern in file)
- `grep -f file_with_pattern file`
 - (look up patterns from one file in another)
- `ssh user@machine`
 - (remote connection)
- Execute a program locally
 - `./executable`

30

- It is possible to combine commands with a “pipe” |
 - output | input
 - `ls | grep -c pattern` (count patterns in filenames)
 - `ls | grep pattern > file`
- Send output to file with “>”
 - `grep pattern file > file2`
- Cut characters from a file
 - File1 (tab separated):
 - This is a test 1
 - This is a test 2
 - This is a test 3
 - `Cut -f 1 file1`
 - This
 - This
 - This

31

- Loops
 - Assuming a directory with:
 - file1, file2,..., fileN
 - We want to run a function over all files, then
 - `for i in $(ls file*); do cut -f 2 $i > $i.out; done`
 - (cuts the field 2 of each file and sends it to an output)
- Execute functions in bashscripts
 - Open a new file and edit it in vim, nano or emacs. Call it bashscript.sh
 - `#!/bin/bash`
 - `for i in $(ls file*); do cut -f 2 $i > $i.out; done`
 - `for i in $(ls *out); do wc $i > $i.wc: done`
 - Make it executable:
 - `chmod a+x bashscript.sh`
 - `./bashscript.sh` (execute)

32

Piping commands together

Several commands in a row:

```
wc -l * | sort -n | head -n 3 > results.txt
```

More on this in Exercise 3.

<https://github.com/krabberod/bios3010-unix-linux>

33

Some online resources

- Linux Cheat Sheet
 - <https://www.cheatography.com/davechild/cheat-sheets/linux-command-line/>
 - <https://files.fooswire.com/2007/08/fwunixref.pdf>
- Vim Cheat Sheet
 - <https://www.cheatography.com/nuggles/cheat-sheets/vim/>
- Nano Cheat Sheet
 - <https://www.cheatography.com/bipinthe/cheat-sheets/nano-editor/>

34

Preinstalled programs on Saga

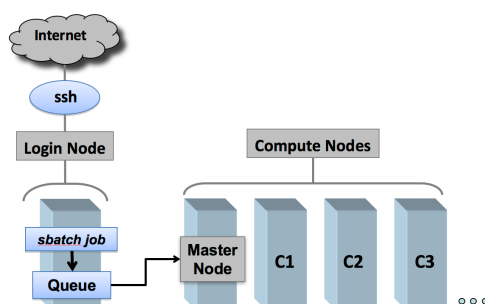
- Saga has already many preinstalled programs.
- Check available modules with
 - `module avail`
- These can be loaded by the command
 - `module load <program>`

```
lander@login1 ~ % module avail
----- /cluster/modulefiles/all
ABYSS/2.0.2-gompi-2019a
AdapterRemoval/2.3.1-foss-2018b
AdapterRemoval/2.3.1-GCC-8.2.0-2.31.1
ADP/2019.103-StaticMKL
AmberTools/5.1-GCC-7.3.0-2.30
AMXTURE/1.3.0
Amber/18-foss-2018b-AmberTools-18-patchlevel-18-8
Amber/18-foss-cuda-2018b-AmberTools-18-patchlevel-18-8
AMOS/3.1.0-foss-2018b
Anaconda2/2019.03
Anaconda3/2019.03
Ancestry_HMM/200105-foss-2019a
angsd-wraper/150321-GCC-7.3.0-2.30
angsd/0.925-foss-2018b
angsd/0.926-GCC-7.3.0-2.30
angsd/0.931-GCC-8.2.0-2.31.1
ARKS/1.0.4-gompi-2019a
Arm-PerfReports/19.1
Armadillo/9.700.2-foss-2019a
arpack-ng/3.7.0-foss-2019a
asf/200115-GCC-8.3.0
ASE/3.17.0-foss-2018b-Python-3.6.6
ASE/3.17.0-intel-2018b-Python-3.6.6
ASE/3.18.0-foss-2019a-Python-3.7.2
ASE/3.18.0-intel-2019a-Python-3.7.2
ASE/3.19.0-foss-2019b-Python-3.7.4
Aspera-CLI/3.9.0.1126.6985b21
AUGUSTUS/3.3.2-intel-2018b-Python-2.7.15
BAMM/2.5.0-foss-2018b
BamTools/2.5.1-foss-2018b
BamTools/2.5.1-GCC-8.2.0-2.31.1
BamTools/2.5.1-iccfort-2019.1.144-GCC-8.2.0-2.31.1
BamTools/2.5.1-intel-2018b
BayesScan/2.1-foss-2018b
BayesScan/2.1-intel-2018b
BayesScan/1.1-foss-2018b
Bazam/1.0.1-java-1.8
BBMap/38.50b-GCC-8.2.0-2.31.1
BCTools/1.9-foss-2018b
BCFTools/1.9-intel-2018b
beagle-1.10.3.0-foss-2018b
beagle-1.10.3.1.2-GCC-8.2.0-2.31.1
Beagle/4.2-22Jan18-Java-1.8
Beagle/5.1-24Mar20-Java-11
Beagle/5.1-25Nov19-Java-11
BeadX/2.5.2-GCC-8.2.0-2.31.1
BEDOPS/2.4.35-foss-2018b
BEDTools/2.27.1-foss-2018b
BEDTools/2.27.1-intel-2018b
BEDTools/2.28.0-GCC-8.2.0-2.31.1
BEDTools/2.28.0-iccfort-2019.1.144-GCC-8.2.0-2.31.1
```

35

Queue system - Slurm

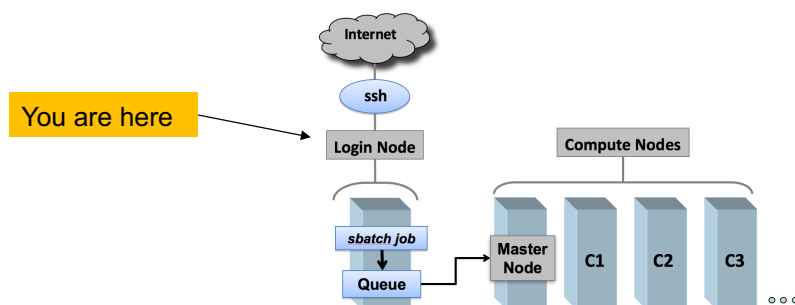
- Since Saga is used by hundreds of people it needs a queueing system to divide the resources
- The queue system is called Slurm.



36

Queue system - SLURM

- Since Saga is used by hundreds of people it needs a queueing system to divide the resources
- The queue system is called Slurm.



37

Typical SLURM script

```
#!/bin/sh
#SBATCH --job-name=job
#SBATCH --account=nn9338k
#SBATCH --output=slurm-%j.base
#SBATCH --cpus-per-task=16
#SBATCH --mem-per-cpu=6G
#SBATCH --time=48:00:00
```

```
module purge
module load SeqKit/0.11.0
<start the program>
```

- A slurm-script **must** start with the line `#!/bin/sh`
- You need to specify the account with computing hours nn9338k for OMG-peeps
- You need to specify the amount of resources you need:
 - CPUs per task is the number of cores
 - Mem-per-CPU is the amount of memory *per CPU* in the previous line (i.e. 16*6Gb = 96 Gb)
- You also need to indicate the time the job will take. Your job will be killed when the time runs out!
- The total amount of CPU hours used is the number of cpus * the actual running time.
- Then do the commands:
 - Example: load a module. Start a program.

38

SLURM script

- Finally send the script to the queue with:
 - `sbatch your.script.slurm`
- Check the queue
 - `squeue -u <username>`
- Check a specific job:
 - `scontrol show jobid <jobid_nr>`

39

Interactive terminal

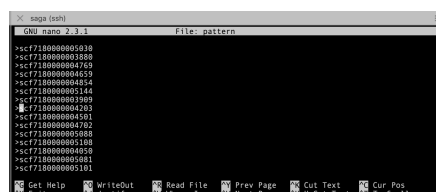
- On the login node you can do small tasks (zipping, coping moving files). Larger tasks will be killed.
- If you need an interactive termial (i.e. resources without sending a script) you can ask for it with `salloc` together with the wanted amount of resources:

```
salloc --account=nn9338k --mem-per-cpu=10G --time=10:00:00 --cpus-per-task=8
```

40

Text editors in Linux

- Unix has many built-in ways editing text files.
- For instance *nano* (or *vim*)
- CTRL-x – Exits the editor. If you are in the middle of editing a file the exit process will ask you if you want to save your work.
- CTRL-o – Save file name and continue working.
- CTRL-w – Search your text.
- CTRL-g – Get help with Nano.



41

- Overwhelmed?
- Keep calm and practice the simple stuff.
 - Test on real cases
- Do the exercises:
- <https://github.com/krabberod/bios3010-unix-linux>