



BIOC | Bioinformatics Competence Center

Bioinformatics Analysis of RNA-sequencing

BIO-693

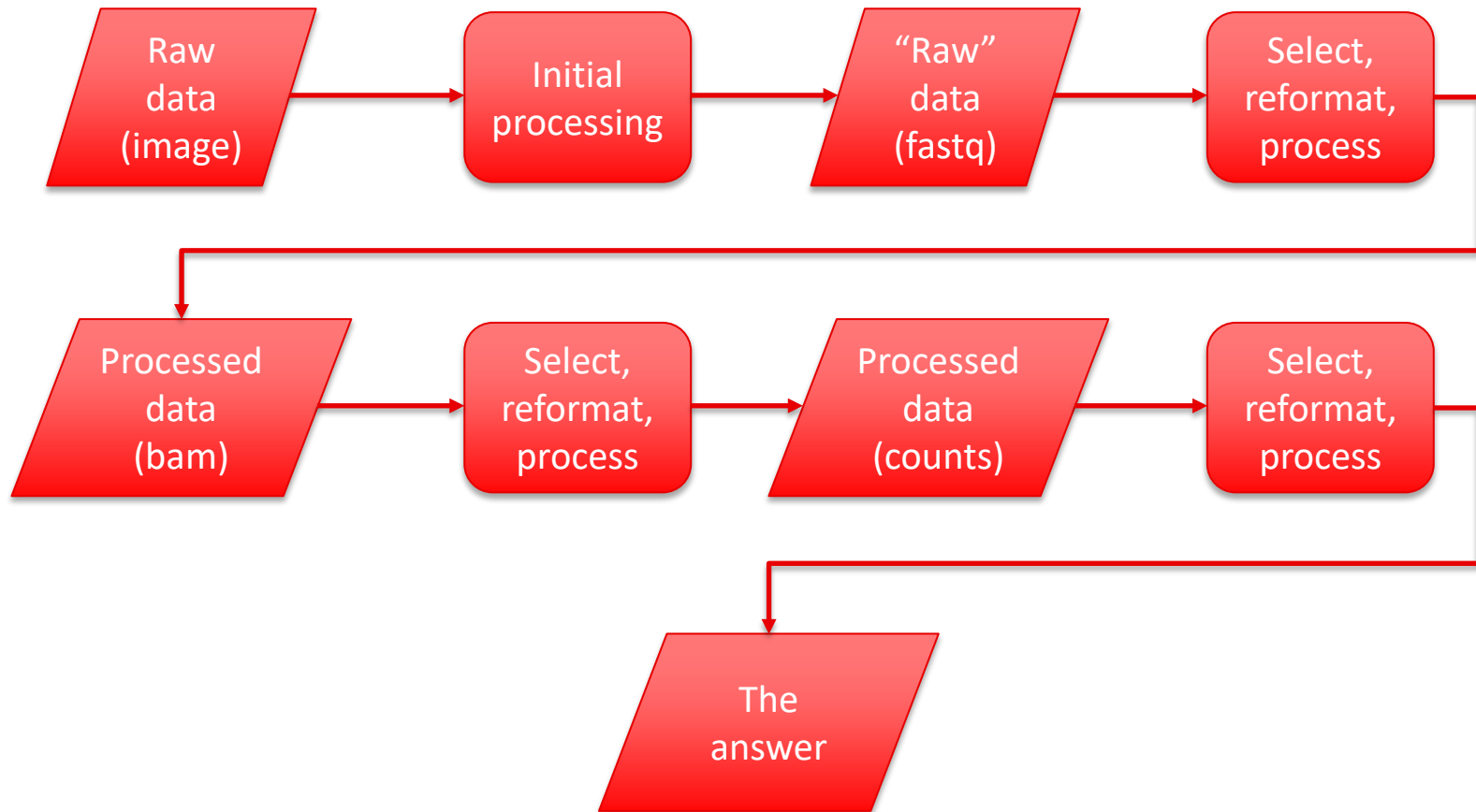
Unil **EPFL**

Allison Burns, Maxime Jan, Linda Mhalla,
Christian Iseli & Nicolas Guex
– summer 2023 –

Program of the course

- Day 1: Working with data at the command line
 - A refresher of bash and R, tips and tricks for data reproducibility
- Day 2: Preparing sequence alignments
 - Downloading and aligning raw RNA-seq data, sequences QC
- Day 3: Bulk RNA-sequencing workflows
 - Getting counts per gene, samples QC, DE, visualization, gene ontology
- Day 4: Single cell RNA-sequencing workflows
 - Filtering and normalizing, clustering and cell type assignments
 - Differential expression analysis and visualization
- Day 5: Prepare and present reports of independent student analysis performed in teams

Bioinformatics analysis



This morning

- Connect to the cluster
- Quick tour of Unix commands
- Containers
- Retrieve raw data to analyze

Create and use **ssh** keys

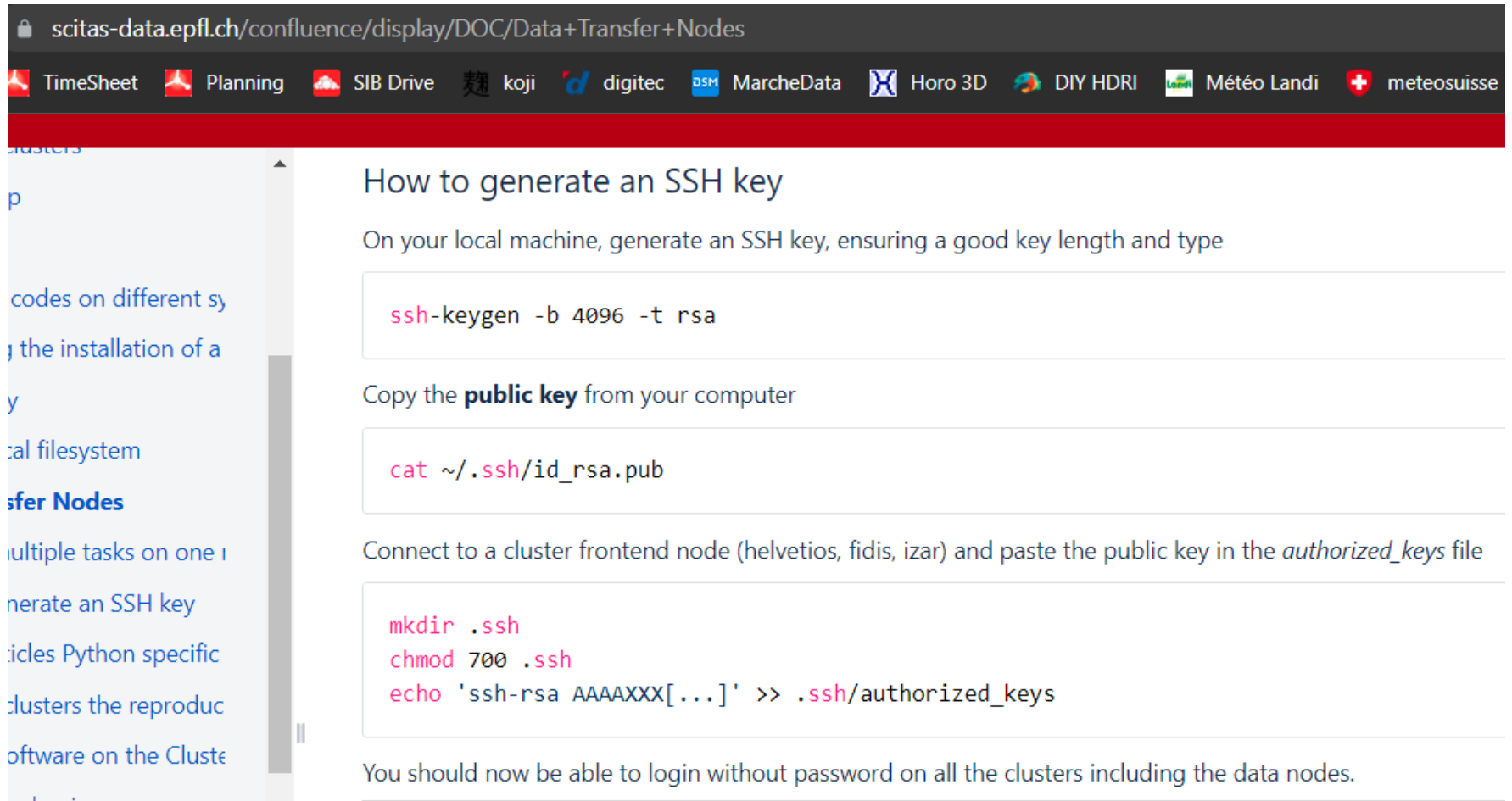


Private key



Public key

Create an ssh key pair



The screenshot shows a web browser window with the address bar displaying `scitas-data.epfl.ch/confluence/display/DOC/Data+Transfer+Nodes`. The browser's top bar contains several icons and labels: TimeSheet, Planning, SIB Drive, koji, digitec, MarcheData, Horo 3D, DIY HDRI, Météo Landi, and meteosuisse. A red navigation bar is visible below the browser bar. On the left side of the page, there is a sidebar with a list of links, including 'clusters', 'p', 'codes on different sy', 'y the installation of a', 'y', 'cal filesystem', 'sfer Nodes', 'multiple tasks on one i', 'nerate an SSH key', 'icles Python specific', 'clusters the reproduc', 'oftware on the Cluste', and 'cluster...'. The main content area of the page is titled 'How to generate an SSH key' and contains the following text and code blocks:

On your local machine, generate an SSH key, ensuring a good key length and type

```
ssh-keygen -b 4096 -t rsa
```

Copy the **public key** from your computer

```
cat ~/.ssh/id_rsa.pub
```

Connect to a cluster frontend node (helvetios, fidis, izar) and paste the public key in the *authorized_keys* file

```
mkdir .ssh
chmod 700 .ssh
echo 'ssh-rsa AAAAXXX[...]' >> .ssh/authorized_keys
```

You should now be able to login without password on all the clusters including the data nodes.

Note

- Part of commands in angular brackets need to be adapted
- E.g.
 - if your username is “iseli”
 - and the command is “ssh <username>@izar.epfl.ch”
 - you need to type the command as “ssh iseli@izar.epfl.ch”

Some ssh basics

- Generate a key pair – `ssh-keygen`
- Organize keys
 - `.ssh/`
 - `id_rsa`
 - `id_rsa.pub`
 - `authorized_keys`
- Agent and forwarding
- Connect to destination – `ssh <user>@<host>`

Example GEO GSM5615574

- URL
 - <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM5615574>
- Select GSE185453 (RNAseq data)
 - <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE185453>
- Go to SRA run selector (link near bottom of page)
 - <https://www.ncbi.nlm.nih.gov/Traces/study/?acc=PRJNA769123>
- Download metadata for all samples
 - Should obtain a text file named `SraRunTable.txt`

Many base file formats are text based

- Line based
 - Can be further split in columns by separator character, tsv, csv
- Group of lines
 - Some lines with special marking
- Mix of the above

Basic shell commands

- `ls, ls -l, pwd, cd, mkdir, cat, cp, mv, rm`
- `stdin, stdout, stderr, >, >>, <, |`
- Need to rudimentary learn 1 text editor (e.g., `vim`)
- `grep, less, wc, echo, cut, head, tail, sed, sort, uniq`
- Be aware of `$PATH`, `env` and `export`
- Be aware of `.bashrc`, `.bash_profile`, `.bash_history`
- Useful to learn variables and arithmetic `$((i + 1))`
- Some characters have a special meaning to the shell
 - `; & < > | ` $ () ' " \`
- Documentation: `man`, `info`, web search

Learn basics of regular expression and globbing

- All shell implement some kind of file globbing
- Many Unix commands make use of regular expressions
- Implementation details vary
- Playground can be useful :
 - <https://regex101.com/>
 - <https://regexr.com/>

Organize data

- Unix uses file paths, e.g., `/work/bix/course`
- `/` has a special meaning as a separator
- A full path always starts with a slash `/`
- A path is relative if it does not start with a slash
- `.` is the current directory, `..` is the parent
- Unix has a concept of symbolic link – `ln -s`
- Each cluster has its own set of conventions
 - `/tmp` – short term temporary files
 - `/scratch` – scratch space
 - `/work` – longer term data storage
 - `/home` – user scripts

Group and compress

- Group using **tar**
- Compress using **gzip**, **bzip2**, **xz**, **7za**
- Group and compress using **zip**

Retrieve data from the web

- `wget`
- `curl`
- SRA toolkit – <https://hpc.nih.gov/apps/sratoolkit.html>
- Cloud toolkits – e.g., `aws-cli`
- <https://scitas-data.epfl.ch/confluence/display/DOC/Mount+a+NAS+share>
- `dbus-run-session -- bash`
- `MYSHARE="smb://intranet;<user>@sv-nas1.rcp.epfl.ch/pteg-raw/Lecture"`
- `gio mount $MYSHARE`
- `gio { list | copy } $MYSHARE/<dir>/<file>`
- `gio mount -u $MYSHARE`

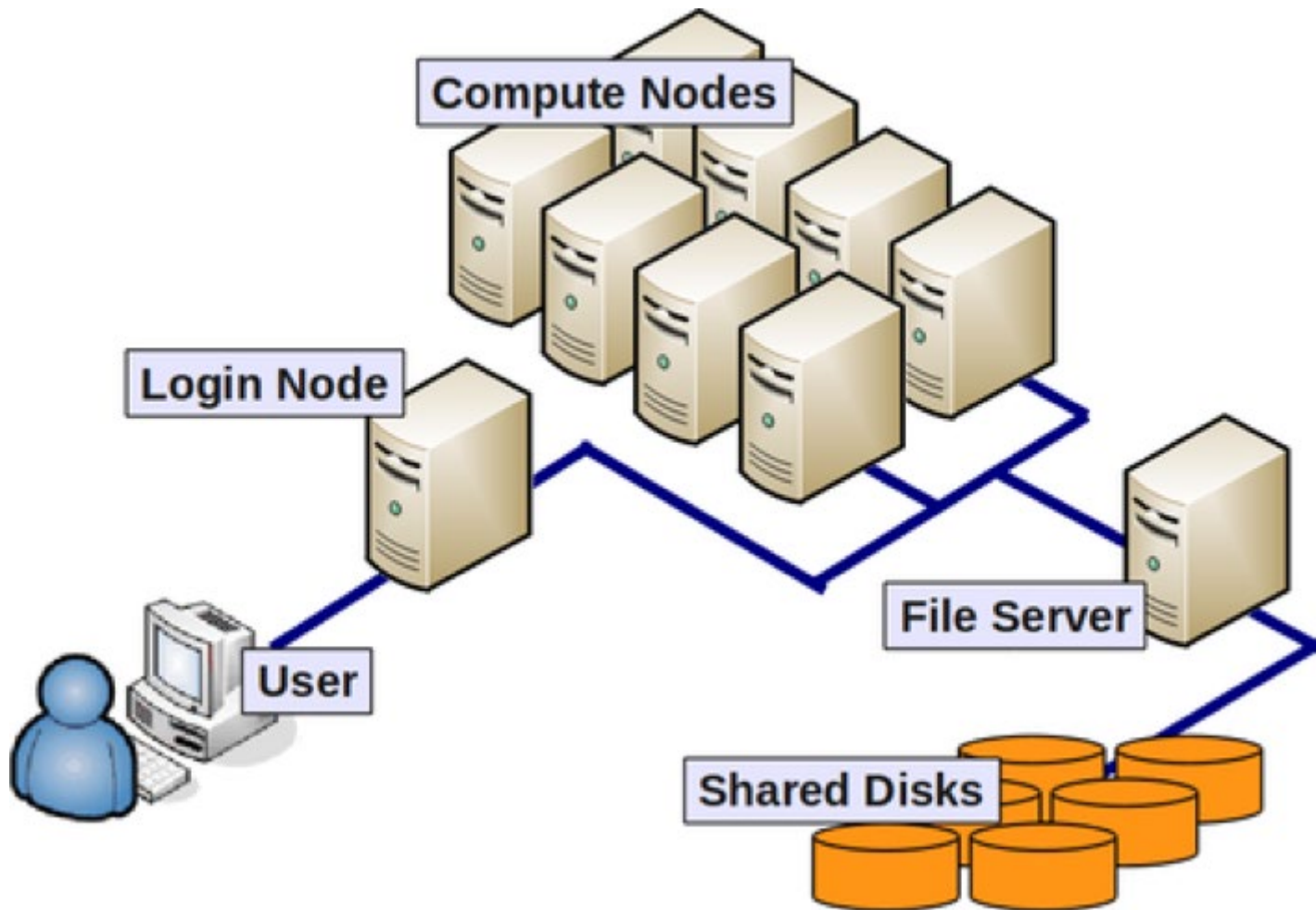
Simple bash scripting

- Process and subprocess – `ps`, `htop`, `jobs`, `history`
- `source` and `.` vs new process
- Executable files – hashbang
- Background, detach, `&`, `&&`, `|` `|`

Persistent sessions: `screen` and `tmux`

- By default, act as a transparent layer
- Can launch several, with different names
- Control character to enter a command
 - Ctrl-A in `screen`, Ctrl-B in `tmux` by default
- Configuration file in your home
 - `.screenrc` and `.tmux.conf`

HPC Structure

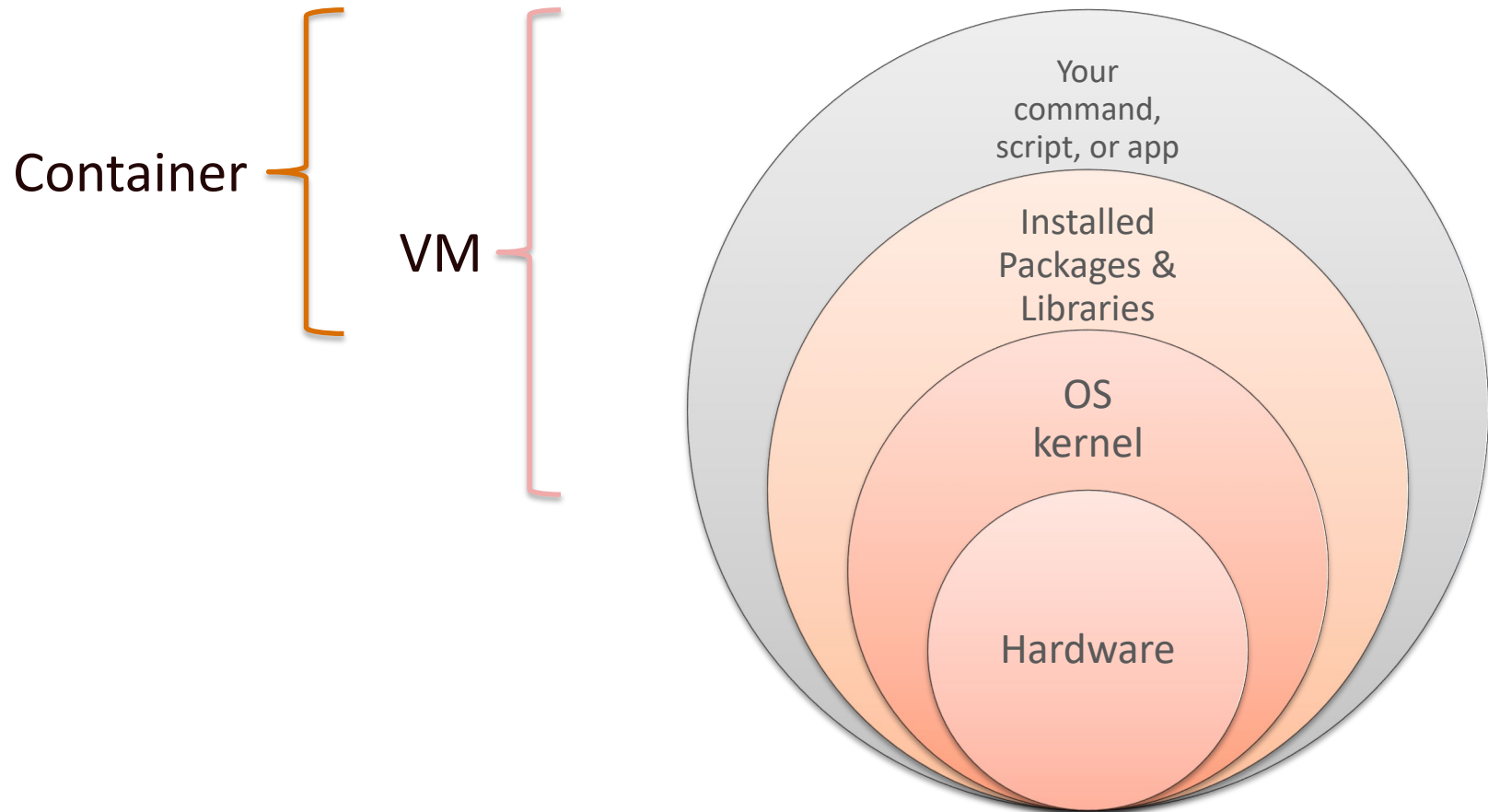


From https://hbctraining.github.io/Intro-to-shell-flipped/lessons/08_HPC_intro_and_terms.html

Cluster, frontal and nodes

- Connect on the frontal
 - Launch **tmux** or **screen**
 - Prepare files
 - Prepare scripts
 - No compute on the frontal
- Use nodes to perform computation
 - Use **Sinteract** for interactive work
 - Use **sbatch** to submit non-interactive jobs
 - Time and resources are limited

Computer layers



Demonstration of container creation

- This video on mediaspace explains container creation
 - [BIO-693 Bioinformatic Analysis of RNA-sequencing - EPFL - Ecole polytechnique fédérale de Lausanne](#)
- Singularity is now named apptainer

Run container

- Use `-B <PATH>` to make directories available in container
- `singularity shell \`
 - `-s /bin/bash \`
 - `-B /scratch/iseli \`
 - `-B /work/bix \`
 - `/work/bix/MicMap-2.4-f38.sif`

Example SRA – setup

- Initial setup
 - `vdb-config -interactive`
- Need a proxy on the cluster
 - `vdb-config --proxy '$HTTP_PROXY' --proxy-disable no`
- Check the resulting configuration
 - `cat ~/.ncbi/user-settings.mkfg`

Example SRA – retrieve

- Retrieve a sample
 - `fasterq-dump --split-3 SRR24947468`
- Retrieve stat info
 - `sra-stat --quick SRR24947468 >SRR24947468_stat.txt`
- Check retrieved data
 - `cat SRR24947468_stat.txt`
 - `wc -l SRR24947468_1.fastq`
 - `echo $((316088 / 4))`
- Compress the fastq files
 - `pigz --fast SRR24947468_?.fastq`