



INSTITUT FRANÇAIS DE BIOINFORMATIQUE

The IFB Core Cluster Infrastructure

FAIR Bioinfo 2023

Gildas Le Corguillé & Julien Seiler
IFB Core Cluster taskforce



DOI [10.5281/zenodo.6628340](https://doi.org/10.5281/zenodo.6628340)

High Performance Computer

Votre ordinateur peut-il faire de la bioinformatique ?



Un ou deux microprocesseurs

Un microprocesseur est chargé de l'exécution des instructions élémentaires demandées par le logiciel

8 à 16 Go de mémoire vive (RAM)

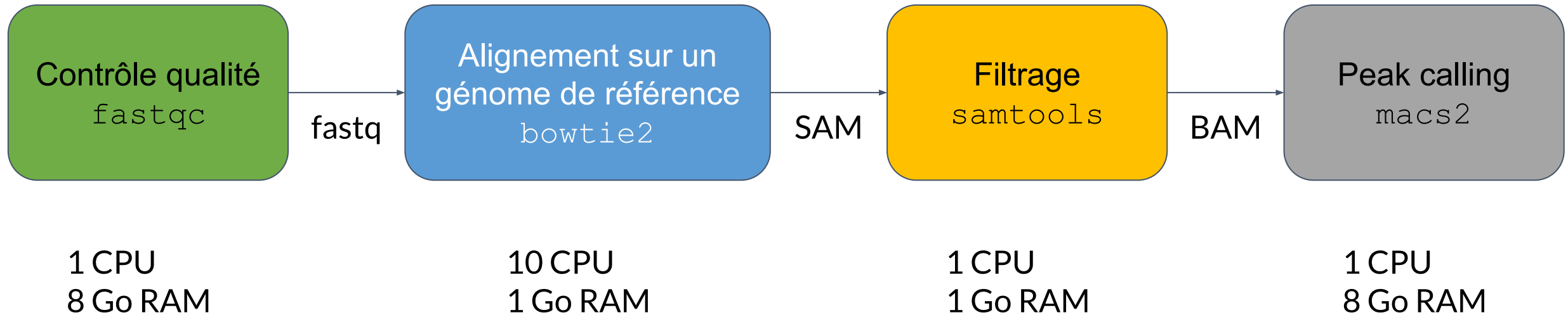
La mémoire vive est utilisée par le microprocesseur pour traiter les données

≈ 1 To d'espace de stockage

L'espace de stockage est utilisé pour conserver de grandes quantités de données de manière plus permanente



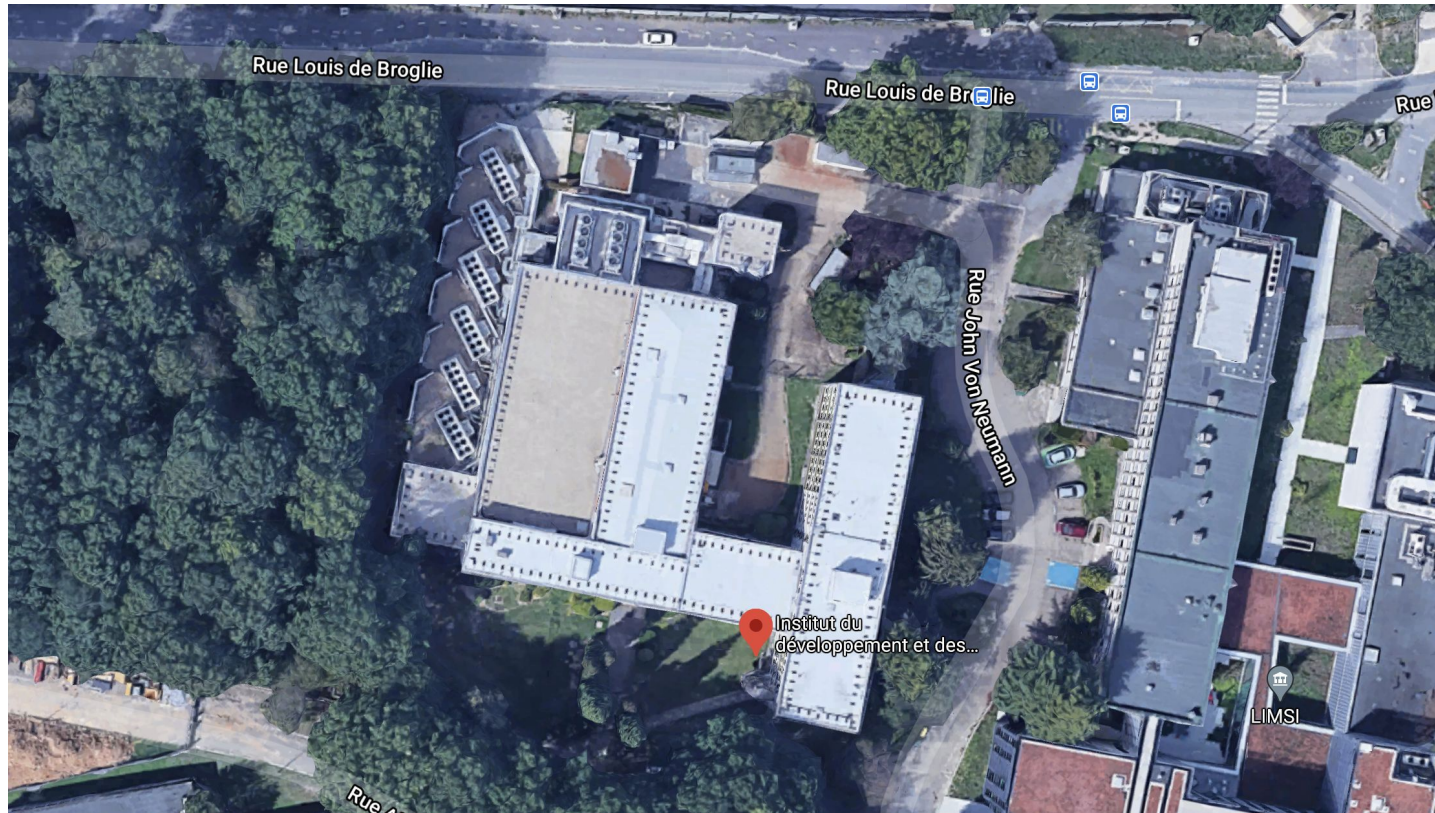
Votre ordinateur peut-il faire de la bioinformatique ?



L'exécution de ce workflow nécessite au minimum toutes les ressources d'un ordinateur de bureau pendant plusieurs heures et ceci seulement pour 1 seul fichier fastq.

Pour faire ce type d'analyse nous avons besoin d'ordinateurs plus puissants !

Du data center au coeur



Le Data Center de l'IDRIS
Un bâtiment conçu pour accueillir des infrastructures informatiques

Du data center au coeur

Groupes froid
Pour refroidir les
équipements



Du data center au coeur

Groupe électrogène
Pour garantir l'alimentation
électrique



Du data center au coeur



Les armoires de l'IFB

Chaque armoire peut contenir
80 super-ordinateurs

Du data center au coeur

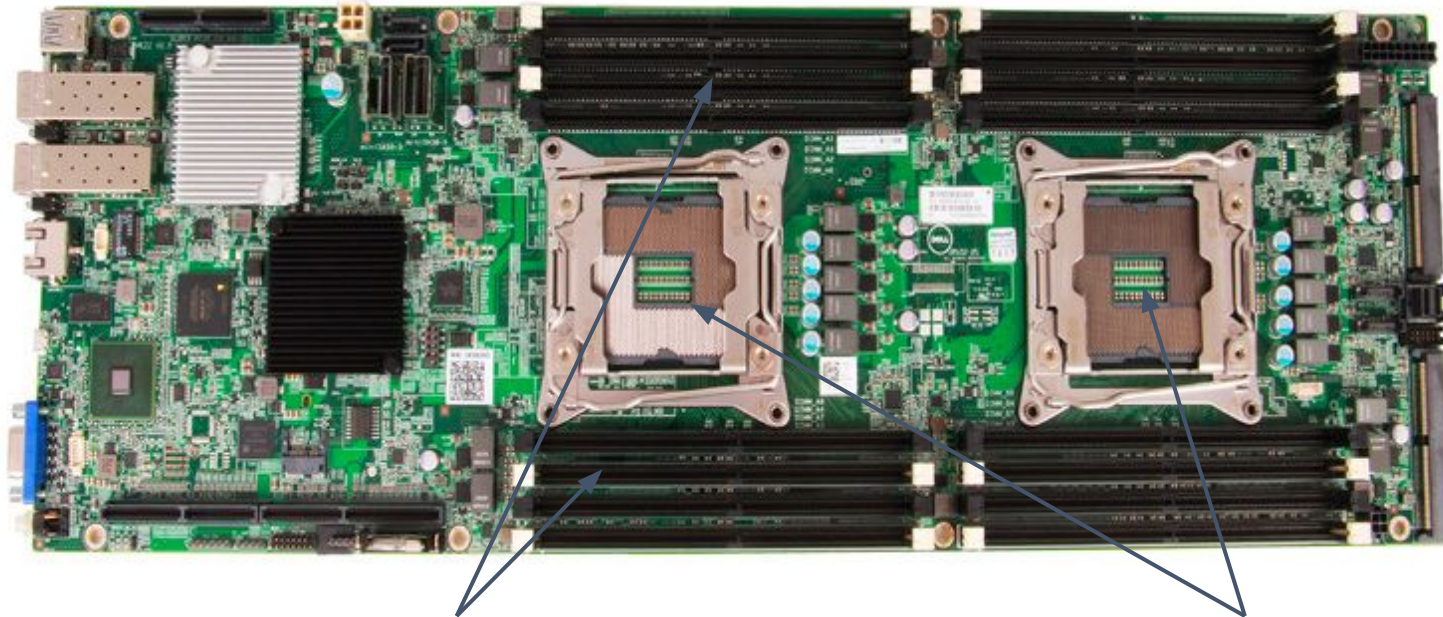


ordinateurs de calcul

Baies de stockage

Du data center au coeur

Un ordinateur ou **noeud** de calcul



Mémoire vive

Supports processeurs

Du data center au coeur

Un microprocesseur

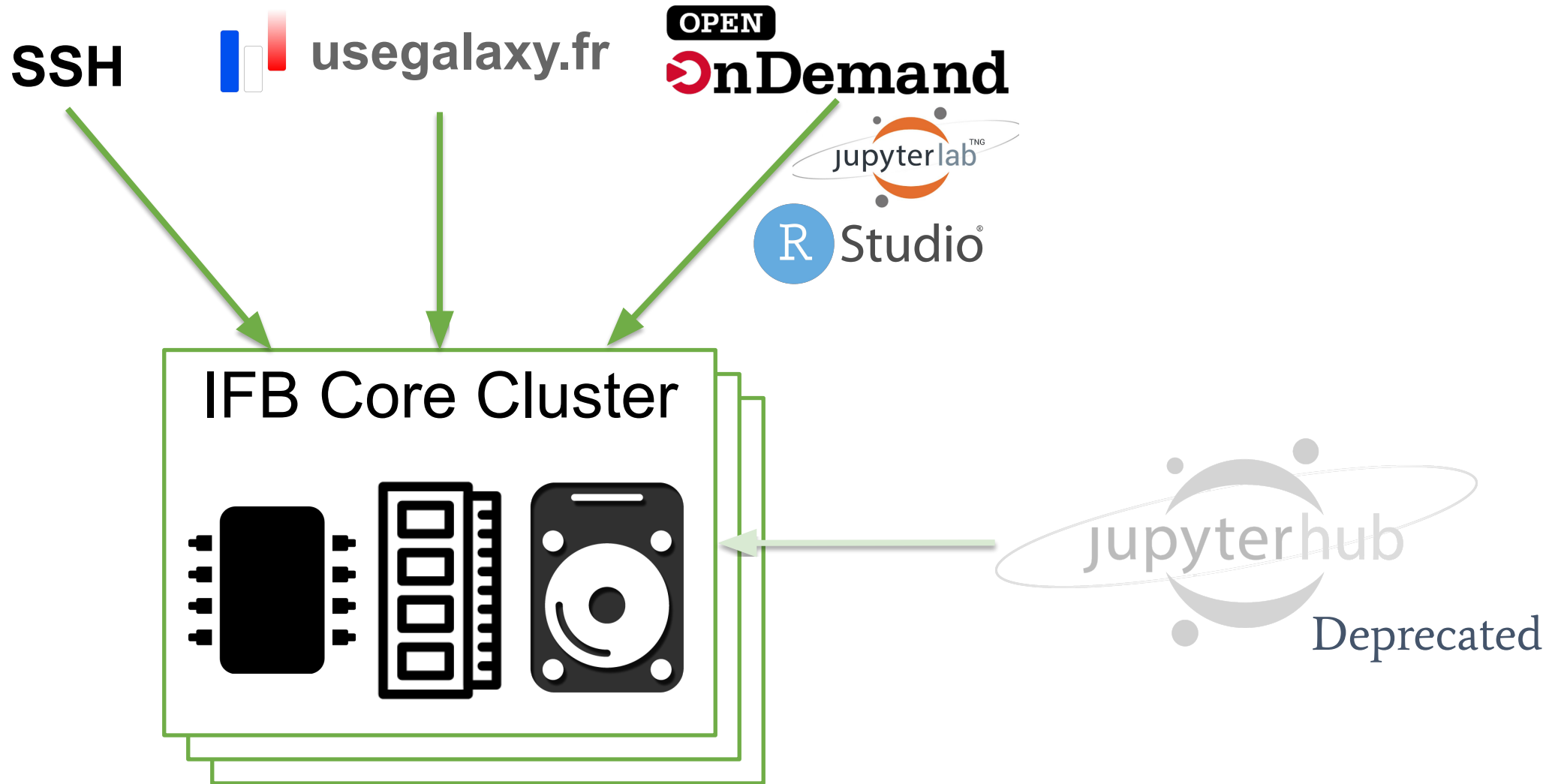


Un microprocesseur contient plusieurs **coeurs**
Chaque coeur se comporte comme un microprocesseur unique.

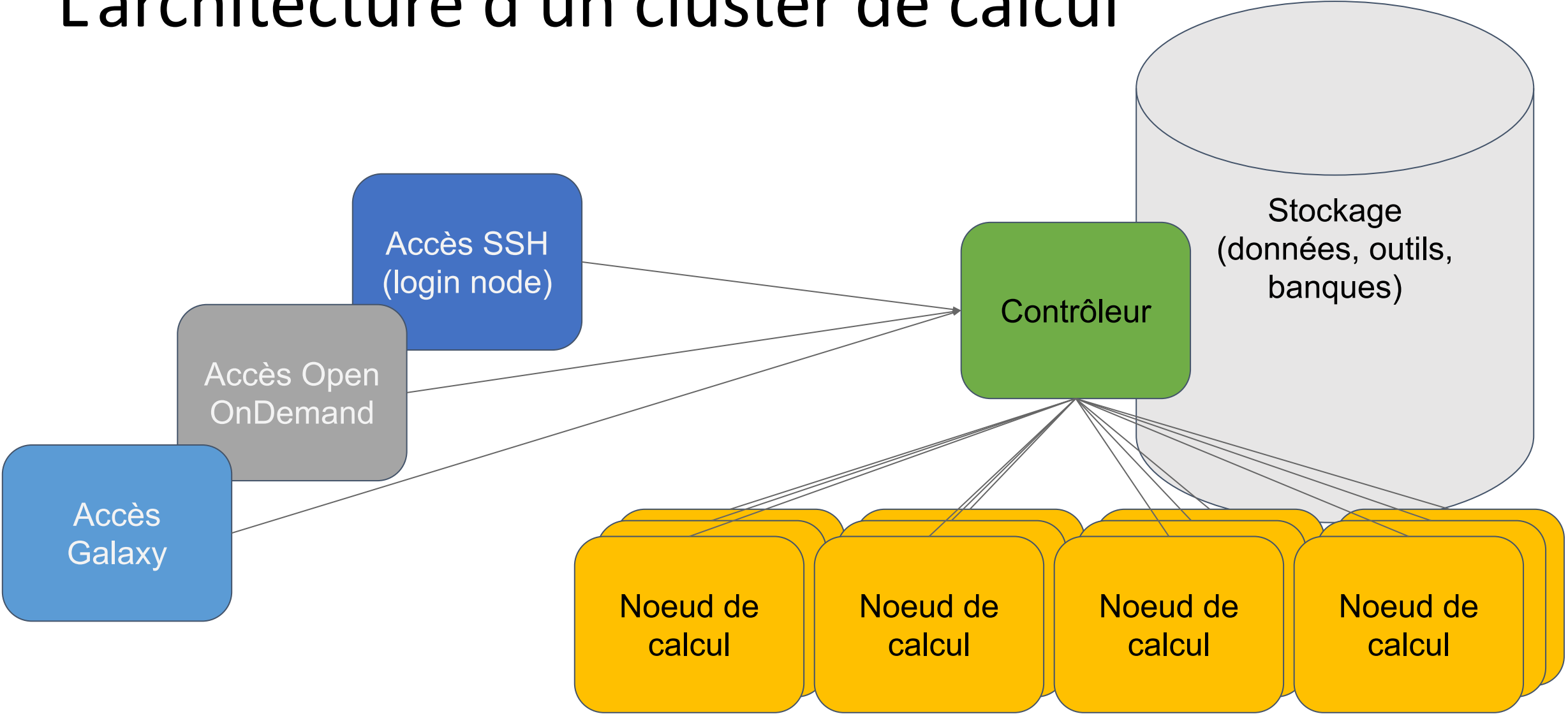
La fédération de cluster de l'IFB (NNCR)

Cluster	Localisation du Data center	Coeurs	RAM (Go)	Stockage (To)
IFB Core	IDRIS - Orsay	5 042	26 542	2 000
Genotoul	Toulouse	6 128	34 304	3 000
ABiMS	Roscoff	2 608	10 600	2 500
GenOuest	Rennes	1 824	7 500	2 300
Migale	Jouy en Josas	1 084	7 000	350
BiRD	Nantes	560	4 000	500

L'infrastructure Core Cluster de l'IFB



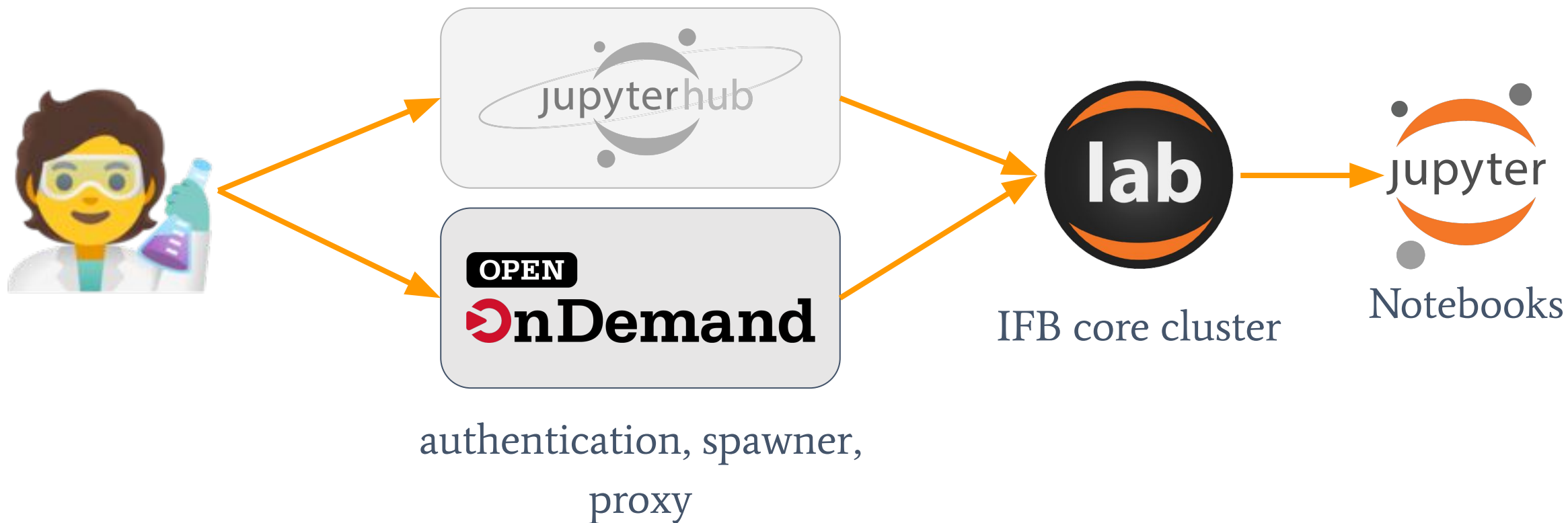
L'architecture d'un cluster de calcul



Jupyter Notebook on a HPC infrastructure

Jupyter Notebook on a HPC infrastructure

JupyterHub and **Open on Demand** are web applications that let you spawn JupyterLab servers on a cluster or cloud infrastructure.



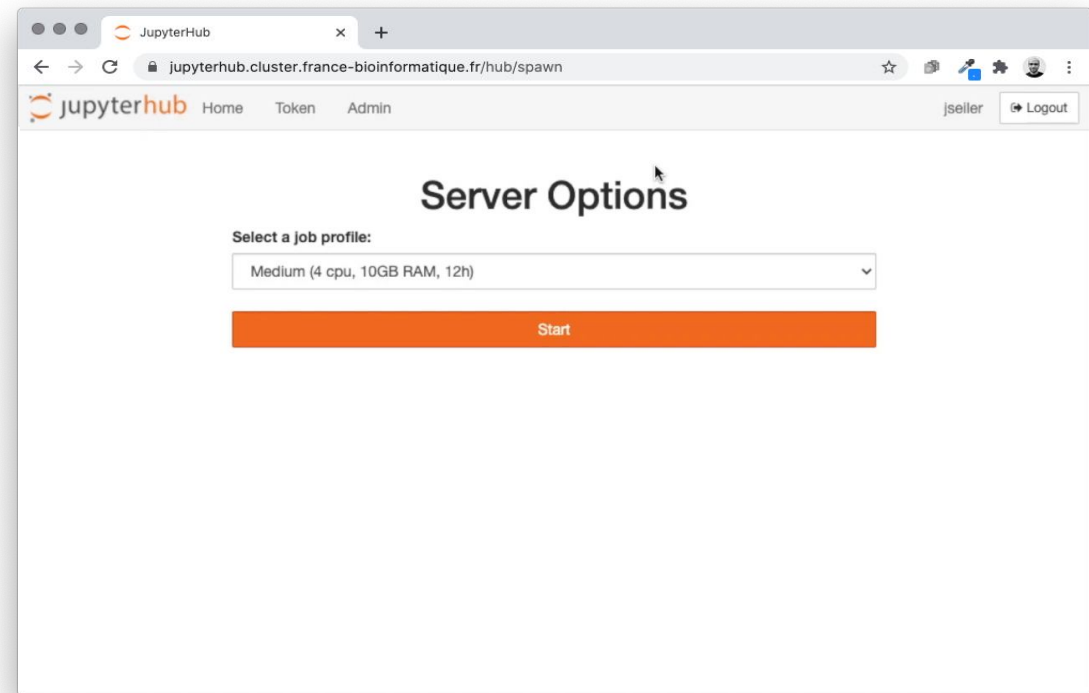
JupyterHub @ IFB

<https://jupyterhub.cluster.france-bioinformatique.fr>

Use your **IFB cluster account**
to log in

Spawn JupyterLab server
in **SLURM jobs**

Work on the **same storage**
as the cluster (ssh)



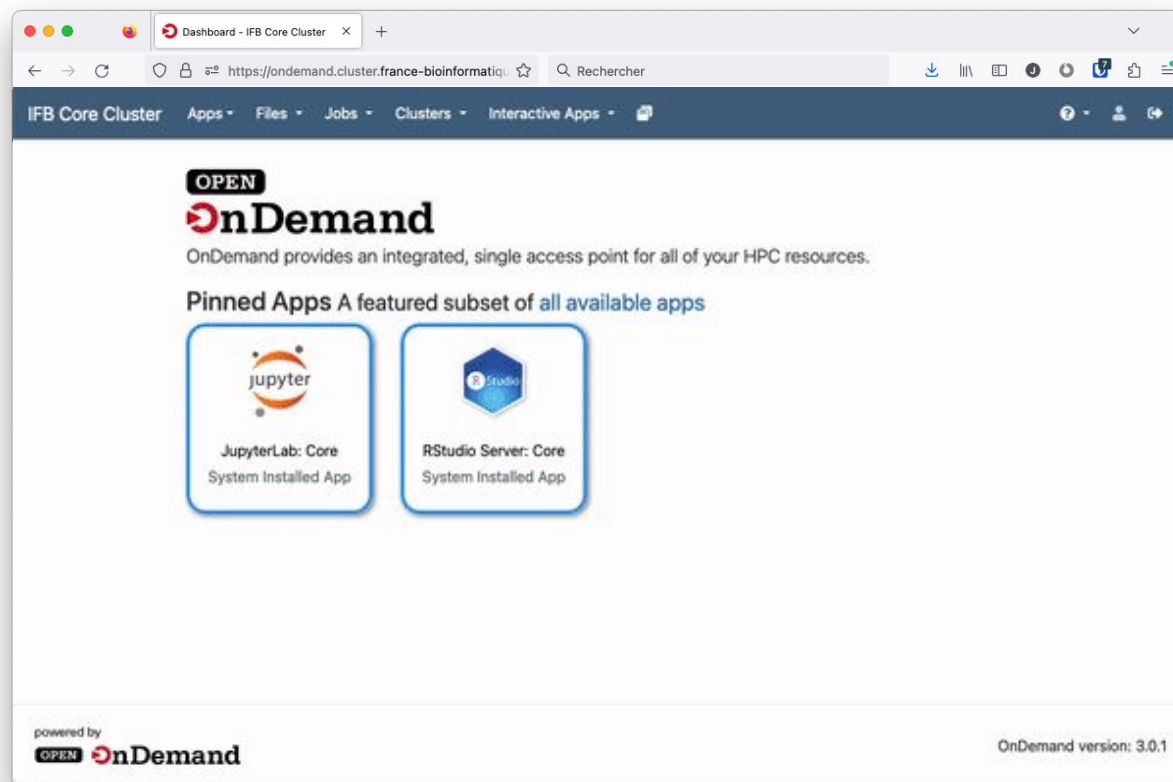
Open OnDemand @ IFB

<https://ondemand.cluster.france-bioinformatique.fr>

Use your **IFB cluster account**
to log in

Spawn Interactive web apps
in **SLURM jobs**

Work on the **same storage**
as the cluster





Demo of notebooks and JupyterLab

Présentation et démonstration ->

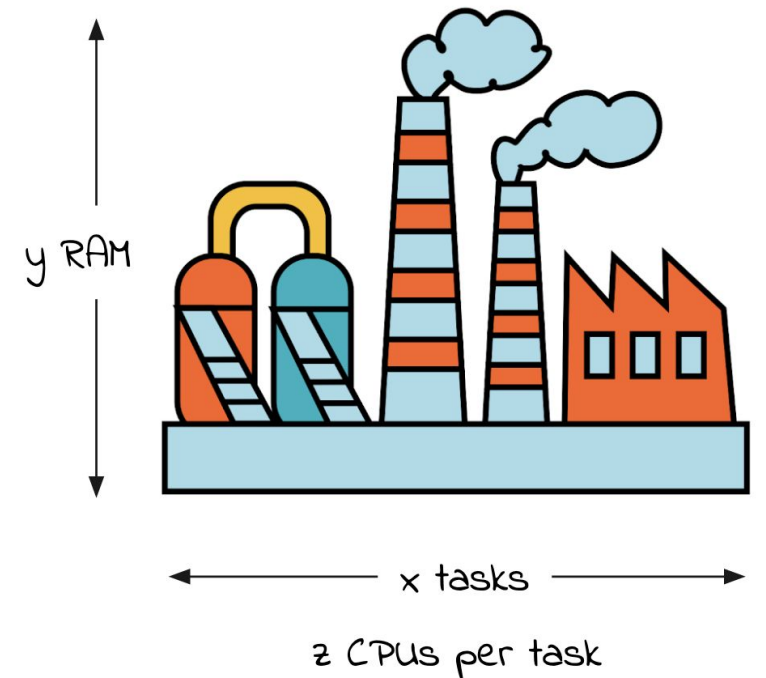
Lean SLURM in a notebook

JupyterLab supports a Bash kernel that let you write notebooks using Bash commands.

The IFB is proposing a SLURM tutorial based on a notebook :

https://gitlab.com/ifb-elixirfr/cluster/tutoriel_slurm

Let's view some best practices to use SLURM the FAIR way.



FAIR Jupyter notebook best practices

- Use Git to follow history of your notebooks (see [JupyterLab Git extension](#))
- Automatically download data from a repository
- Make sure to identify the version of the libraries you are using in your notebooks :
 - Python : [watermark](#) or [session_info](#)
 - R : [sessionInfo](#)

For more tips read :

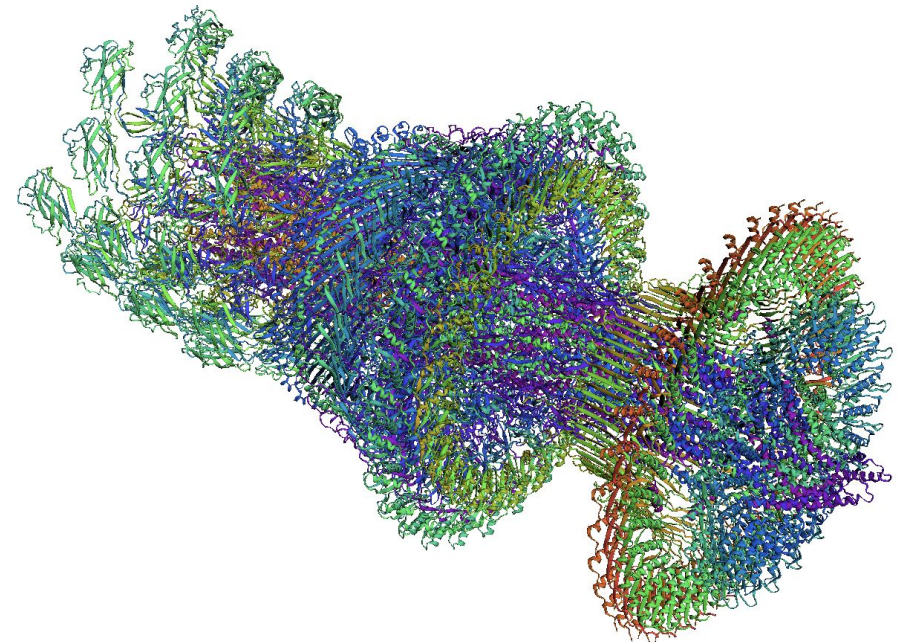
<https://github.com/jupyter-guide/ten-rules-jupyter>

More interactive analysis with notebooks

- Render (dynamic) charts and visualize 3d models
- Train a network with Tensorflow and visualize training logs with Tensorboard

Demo notebooks can be downloaded from

<https://gitlab.com/ifb-elixirfr/notebooks/fairbioinfo-demo>





Conda - usage

Installation of miniconda (only once)

```
$ wget https://repo.anaconda.com/miniconda/Miniconda3-py39_4.9.2-Linux-x86_64.sh
$ bash Miniconda3-py39_4.9.2-Linux-x86_64.sh -b -p ~/miniconda3
$ conda config --add channels bioconda; conda config --add channels conda-forge
```

Search for a package

```
$ conda search fastqc==0.11.9    or https://anaconda.org/search?q=fastqc
```

Create an environment for a tool (recommended)

```
$ conda create -n fastqc-0.11.9 fastqc==0.11.9
```

Load a conda environment and use

```
$ conda activate fastqc-0.11.9
$ fastqc --version
FastQC v0.11.9
```



Conda packages are provided by a central repository hosted by a company called Anaconda.org

Conda - building



Conda packaging consists of 2 files

```
1  package:
2  name: fastqc
3  version: 0.11.9
4
5  source:
6  url: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/fastqc_v0.11.9.zip
7  sha256: 15510a176ef798e40325b717cac556509fb218268cfdb9a35ea6776498321369
8  patches:
9    - java_xms.patch
10
11 build:
12 noarch: generic
13 number: 1
14
15 requirements:
16 run:
17   - openjdk >=8.0.144
18   - perl
19   - fontconfig
20
21 test:
22 commands:
23   - fastqc -h
24   - fastqc --version
25
26 about:
27 home: 'http://www.bioinformatics.babraham.ac.uk/projects/fastqc/'
28 license: GPL >=3
29 summary: 'A quality control tool for high throughput sequence data.'
```

[recipes/fastqc/meta.yml](https://github.com/bioconda/recipes/blob/master/recipes/fastqc/meta.yaml)

```
1  #!/bin/bash
2
3  fastqc=$PREFIX/opt/$PKG_NAME-$PKG_VERSION
4  mkdir -p $fastqc
5  cp -r ./ * $fastqc
6  sed -i.bak '1 s|^.*$|#!/usr/bin/env perl|g' $fastqc/fastqc
7  rm -f $fastqc/fastqc.bak
8  chmod +x $fastqc/fastqc
9  mkdir -p $PREFIX/bin
10 ln -s $fastqc/fastqc $PREFIX/bin/fastqc
11
```

[recipes/fastqc/build.sh](https://github.com/bioconda/recipes/blob/master/recipes/fastqc/build.sh)

```
$ # To build and test locally
$ conda build .
```



Consider to contribute to the
Bioconda community/channel
<https://bioconda.github.io/>

Docker - usage



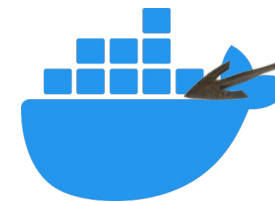
Search for a Docker image or <https://hub.docker.com/r/biocontainers/fastqc>

```
$ docker search fastqc
NAME                DESCRIPTION          STARS     OFFICIAL   AUTOMATED
biocontainers/fastqc fastqc               3         OFFICIAL   [OK]
```

Pull and Run

```
$ docker run biocontainers/fastqc:v0.11.9_cv8 fastqc --version
[...]
$ FastQC v0.11.9
```

i Docker isn't reliable in the context of an HPC infrastructure because of the need of the Docker daemon



Docker - building



```
1 FROM ubuntu:19.04
2
3 RUN apt-get update && apt-get install -y software-properties-common
4
5 RUN apt-get update && \
6     apt-get install -y openjdk-8-jre && \
7     rm -rf /var/lib/apt/lists/*
8
9 ENV JAVA_HOME /usr/lib/jvm/java-8-openjdk-amd64/
10
11 RUN apt-get -qq update && apt-get -y upgrade && \
12     apt install -y wget libfindbinlibs-perl software-properties-common unzip
13
14 RUN wget https://www.bioinformatics.babraham.ac.uk/projects/fastqc/fastqc_v0.11.9.zip -O /tmp/fastqc.zip && \
15     unzip /tmp/fastqc.zip -d /opt/ && \
16     rm /tmp/fastqc.zip && \
17     chmod 777 /opt/FastQC/fastqc
18
19 ENV PATH="/opt/FastQC/:${PATH}"
20
21 ENTRYPOINT ["fastqc"]
22
```

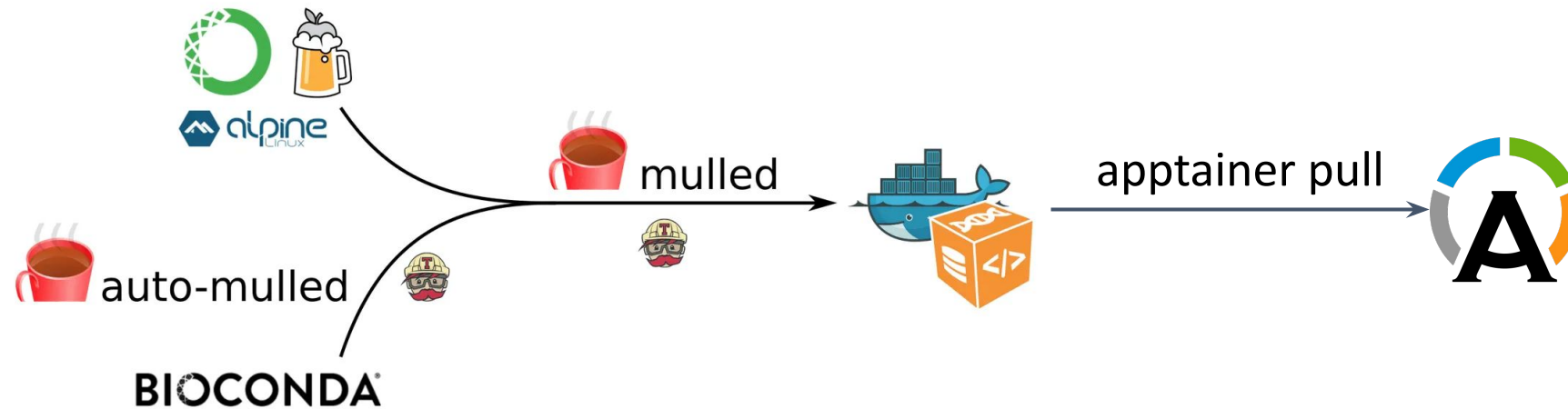
[Dockerfile](#)

```
$ # To build and test locally
$ docker build -t fastqc-0.11.9
```

```
$ # Use
$ docker run fastqc-0.11.9 --version
$ FastQC v0.11.9
```

Conda 2 Docker 2 Apptainer

BioContainer



Process this full example:

<https://ifb-elixirfr.gitlab.io/cluster/doc/singularity/#a-full-example>

Apptainer - usage



Search for a Docker a image

<https://hub.docker.com/r/biocontainers/fastqc>

Pull an image

```
$ apptainer pull docker://biocontainers/fastqc:v0.11.9_cv8
$ ls -l fastqc_v0.11.9_cv8.sif
-rwxr-xr-x 1 foo bar 297582592 Jun 22 18:11 fastqc_v0.11.9_cv8.sif
```

Use

```
$ ./fastqc_v0.11.9_cv8.sif fastqc --version
$ FastQC v0.11.9
```

Apptainer - build






```
1  BootStrap: docker
2  From: biocontainers/fastqc:v0.11.9_cv8
3
4  %labels
5      Author IFB
6      Version 0.11.9
7
8  %environment
9      export PATH=/usr/local/bin:$PATH
10
11 %runscript
12     exec "$@"
13
14 %test
15     export PATH=/usr/local/bin:$PATH
16     fastqc --version | grep "0.11.9"
```

OR

```
1  BootStrap: docker
2  From: ubuntu:19.04
3
4  %labels
5      Author IFB
6      Version 0.11.9
7
8  %post
9      apt-get update && apt-get install -y software-properties-common
10     apt-get update && \
11         apt-get install -y openjdk-8-jre && \
12         rm -rf /var/lib/apt/lists/*
13     JAVA_HOME /usr/lib/jvm/java-8-openjdk-amd64/
14     apt-get -qq update && apt-get -y upgrade && \
15     apt install -y wget libfindbin-libs-perl software-properties-common unzip
16
17     wget https://www.bioinformatics.babraham.ac.uk/projects/fastqc/fastqc_v0.11.9.zip -O /opt/fastqc.zip && \
18     unzip /opt/fastqc.zip -d /opt/ && \
19     rm /opt/fastqc.zip && \
20     chmod 777 /opt/FastQC/fastqc
21
22 %environment
23     export PATH=/usr/local/bin:$PATH
24
25 %runscript
26     exec "$@"
27
28 %test
29     export PATH=/usr/local/bin:$PATH
30     fastqc --version | grep "0.11.9"
```

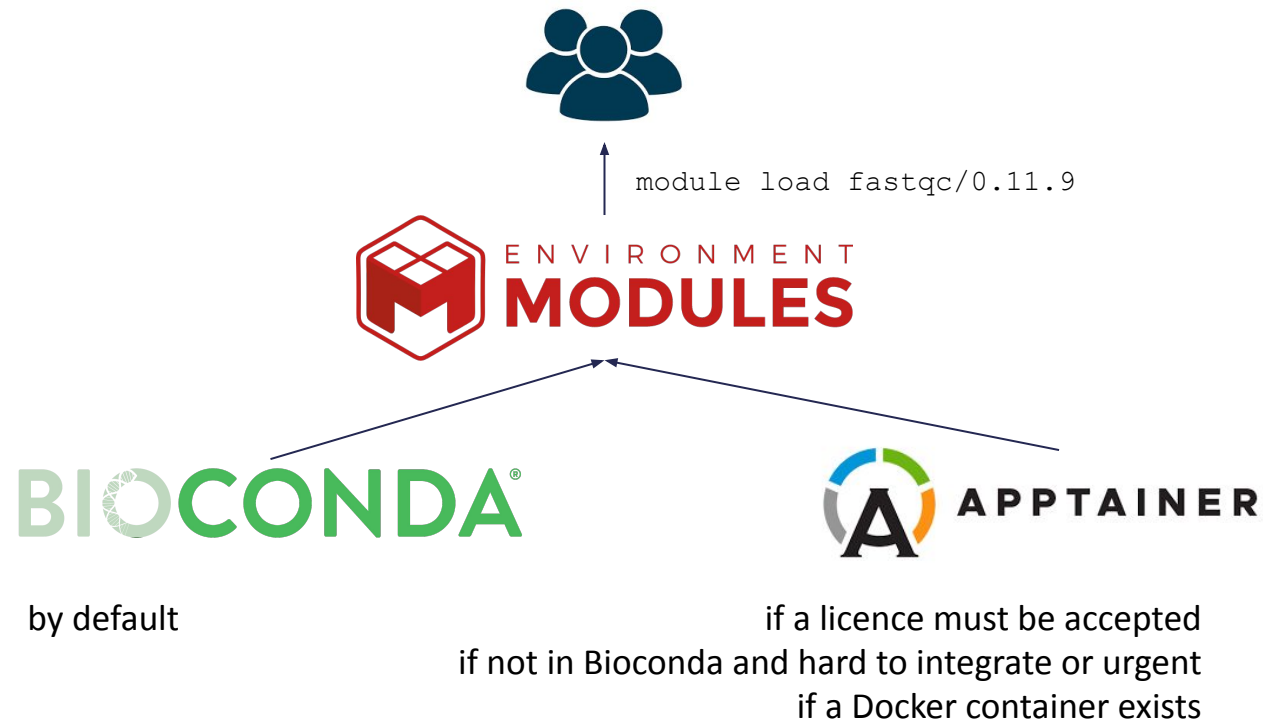
image.def

Conda or Docker or Apptainer ?

	PROS	CONS
	<ul style="list-style-type: none">• Light during the installation• Not need to be root• Sharing repository• The Alpha exported in Docker	<ul style="list-style-type: none">• They are issue to install “old” packages 2 or 3 years after their creation (dependencies have changed)• A lot of tiny files• No isolation - security issue• Can be slow to revolve multi-packages env
	<ul style="list-style-type: none">• Portable• Sharing repository• Can be ate by Singularity• Come with the OS	<ul style="list-style-type: none">• Not compatible with the HPC infrastructure• Rather heavy to install, need root grants<ul style="list-style-type: none">◦ Need a centralized daemon• Some security issues/concerns
	<ul style="list-style-type: none">• Compatible with HPC since it's execute as a binary• Compatible with Docker image format• Come with the OS	<ul style="list-style-type: none">• Don't provide the same layer system as Docker<ul style="list-style-type: none">◦ So heavier on the filesystem• No stable shared repository (yet)• It's a deadlock that can't be exported• Not well integrated on MacOSX

Module - usage at IFB

2 technologies - 1 user interface



Module - usage

Why do we need to "load" tools ?

- Each tools need its environment (binaries, libraries, documentation, special variables)
- Each tools has its own dependencies.
- It is not possible to coexist all tools in the same environment.
- Reproducibility does matter: some user might need different versions of the same tool
- At the IFB, the cluster community is installing all tools required by the users.

All tool deployment are based on Conda packages or Singularity images :



To get access to a tool, you need to load it into your environment using a special tool called **module**.

Module - usage

Loading, listing, switching, unloading

```
module avail          # List the modules available (477 in June 2021)
module avail fastqc  # List the versions available for a tool

module load fastqc   # Bad: Load latest version available on the cluster
module load fastqc/0.11.9 multiqc/1.10.1 # Load software
module list          # List tools currently loaded in your environment

module switch fastqc/0.11.7 # Replace current version

module unload blast  # Unload blast from your environment
module purge         # Unload all tools
```

Module - build at IFB

Institut Français de Bioinformatique > Cluster > tools

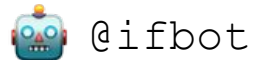


tools

Project ID: 15693267

```
1 channels:  
2 - conda-forge  
3 - bioconda  
4 - defaults  
5 dependencies:  
6 - bioconda::fastqc=0.11.9  
7 name: fastqc-0.11.9
```

[tools/fastqc/0.11.9/meta.yml](#)



```
1 deployment: conda  
2  
3 about:  
4   description: "A quality control tool for high throughput sequence data."  
5   url: http://api.anaconda.org/packages/bioconda/fastqc
```

[tools/fastqc/0.11.9/meta.yml](#)



ENVIRONMENT
MODULES

.pre

Changes de...

Test

IFB dev Con...

IFB dev Sing...

IFB preprod ...

IFB preprod Si...

Production

ABIMS Sing...

ABiMS Conda

BiRD Conda

BiRD Singul...

CCUS Conda

CCUS Singu...

IFB Conda

IFB Singulari...

IGBMC Conda

IGBMC Sing...

MCIA Conda

MCIA Singul...

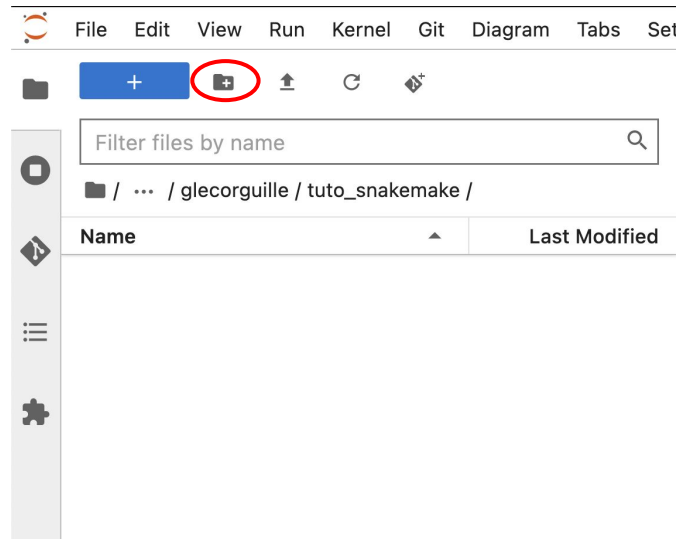


TP - Snakemake over SLURM

TP - Snakemake over SLURM

Exercice 1: connect to the cluster through JupyterHub

- Go to <https://ondemand.cluster.france-bioinformatique.fr>
- Start a small JupyterLab server with 1 CPU and 4 GB of RAM
- Create an empty folder in your project directory



TP - Snakemake over SLURM

Exercice 1: connect to the cluster through JupyterHub

- Go to <https://ondemand.cluster.france-bioinformatique.fr>
- Start a small JupyterLab server with 1 CPU and 4 GB of RAM
- Create an empty folder in your project directory
- Clone the tutorial git repository

`https://gitlab.com/ifb-elixirfr/notebooks/tutoriel_slurm.git`

TP - Snakemake over SLURM

Exercice 1

- Go to [h](#)
- Start a
- Create
- Clone t

[http](http://)

The screenshot displays the JupyterLab interface. The main window shows a file browser for the path `shared/projects/taskforce/glecorguille/tuto_snakemake`. A message indicates that the current directory is not a Git repository. Three blue buttons are visible: "Open the FileBrowser", "Initialize a Repository", and "Clone a Repository". A "Clone a repo" dialog box is open in the foreground, containing the following elements:

- Title: "Clone a repo"
- Instruction: "Enter the URI of the remote Git repository"
- Input field: `https://gitlab.com/ifb-elixirfr/notebooks/tutoriel_slurm.git`
- Checkbox: "Include submodules"
- Checkbox: "Download the repository"
- Buttons: "Cancel" and "Clone"

TP - Snakemake over SLURM

Exercise 2: Get your environment ready

- Download the workflow
- Download your input data
- Load the snakemake module and all required tools

TP - Snakemake over SLURM

Exercice 2: Get your environment ready

- Download the workflow
- Download your input data

The data used for the snakemake tutorial are available on Zenodo :

DOI [10.5281/zenodo.3997237](https://doi.org/10.5281/zenodo.3997237)

TP - Snakemake over SLURM

Exercise 2: Get your environment ready

- Download the workflow
- Download your input data

Download the snakemake workflow and data archive :

```
$ git clone https://github.com/clairetn/FAIR_smk.git  
$ cd FAIR_smk
```

```
$ module load zenodo_get/1.3.2  
$ zenodo_get 10.5281/zenodo.3997237  
$ tar -xvzf FAIR_Bioinfo_data.tar.gz
```

TP - Snakemake over SLURM

Exercise 3: Run snakemake

- Run your workflow using `--cluster mode`
- Run your workflow using `--drmaa mode`

TP - Snakemake over SLURM

Exercice 3: Run snakemake

- Run your workflow using `--cluster` mode

```
module load snakemake
```

```
snakemake -c 1 -s ex1_o8.smk --delete-all-output; rm -rf multiqc_*
```

```
snakemake --cluster "sbatch" --jobs=3 --cores=3 --use-conda -s ex1_o8.smk
```

Drawbacks : no control on workflow execution (you can't stop it)

TP - Snakemake over SLURM

Exercise 3: Run snakemake

- Run your workflow using `--cluster` mode
- Run your workflow using `--drmaa` mode

Distributed Resource Management Application API



TP - Snakemake over SLURM

Exercise 3: Run snakemake

- Run your workflow using `--cluster mode`
- Run your workflow using `--drmaa mode`

```
module load snakemake
```

```
snakemake --drmaa --use-conda --jobs=3 -s ex1_o8.smk
```

TP - Snakemake over SLURM --use-conda



 CONDA

```
rule fastqc:  
[...]  
  conda:  
    "envs/fastqc-0.11.9.yml"  
  container:  
    "docker://biocontainers/fastqc:v0.11.9_cv8"  
  envmodules:  
    "fastqc/0.11.9"  
  shell: "fastqc --outdir FastQC/ {input} 1>{log.std} 2>{log.err}"
```

```
module purge; module load snakemake conda
```

```
snakemake -c 1 -s ex1_o8.smk --delete-all-output; rm -rf multiqc_*
```

```
time snakemake --drmaa --jobs=3 -s ex1_o8.smk --use-conda
```

TP - Snakemake over SLURM --use-singularity



```
rule fastqc:
[...]
```

```
conda:
    "envs/fastqc-0.11.9.yml"
```

```
container:
    "docker://biocontainers/fastqc:v0.11.9_cv8"
```

```
envmodules:
    "fastqc/0.11.9"
```

```
shell: "fastqc --outdir FastQC/ {input} 1>{log.std} 2>{log.err}"
```

```
module purge; module load snakemake singularity
```

```
snakemake -c 1 -s ex1_o8.smk --delete-all-output; rm -rf multiqc_*
```

```
time snakemake --drmaa --jobs=3 -s ex1_o8.smk --use-singularity
```


TP - Snakemake over SLURM --use-envmodule



```
rule fastqc:
[...]
```

```
  conda:
    "envs/fastqc-0.11.9.yml"
```

```
  container:
    "docker://biocontainers/fastqc:v0.11.9_cv8"
```

```
  envmodules:
    "fastqc/0.11.9"
```

```
  shell: "fastqc --outdir FastQC/ {input} 1>{log.std} 2>{log.err}"
```

```
module purge; module load snakemake
```

```
snakemake -c 1 -s ex1_o8.smk --delete-all-output; rm -rf multiqc_*
```

```
time snakemake --drmaa --jobs=3 -s ex1_o8.smk --use-envmodule
```

Useful links

Request an account:

<https://my.cluster.france-bioinformatique.fr>

Community support:

<https://community.france-bioinformatique.fr/>

Learn SLURM in 5 minutes:

<https://asciinema.org/a/275233>

IFB Core Cluster Documentation

<https://ifb-elixirfr.gitlab.io/cluster/doc/>

BONUS

The IFB Core Cluster Infrastructure

- Infrastructure administration is automated using Continuous Integration technologies :



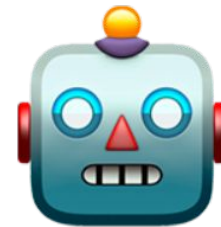
ANSIBLE



git



GitLab



IFBot

- Most IFB Core Cluster repositories are **open to contribution**
 - Help us manage the cluster infrastructure
 - Deploy bioinformatics software (conda, singularity, etc.)
 - Deploy new services



What's ~~was~~ new on the IFB NNCR Cluster(s) ?

David BENABEN ^{1,2}, Nicole CHARRIÈRE ³, David CHRISTIANY ³, François GERBES ^{3,6}, Jean-Christophe HAESSIG ⁴,
Didier LABORIE ⁵, Gildas LE CORGUILLÉ ^{6*}, Olivier SALLOU ⁷, Julien SEILER ^{4*} and Guillaume SEITH ⁴



- ¹ CBiB, Université de Bordeaux, 142 rue Léo Saignat, 33076 Bordeaux, France
² INRAE, UMR 1332, Biologie du Fruit et Pathologie, CS20032 Villenave d'Ornon, France
³ IFB/Institut Français de Bioinformatique, CNRS UMS 3601, IFB-Core, Génoscope, 91057, Évry, France
⁴ CNRS, INSERM, IGBMC, 1 rue Laurent Fries, 67404, Illkirch, France
⁵ GenoToul-Bioinfo, INRAE, 24 chemin de Borde-Rouge, Auzeville, 31326 Castenet-Tolosan, France
⁶ Sorbonne Université/CNRS, FR2424, ABiMS, Station Biologique, 29680, Roscoff, France
⁷ IRISA/Université Rennes 1, 263 Avenue Général Leclerc, 35000 Rennes, France

* Corresponding Authors: lecorguille@sb-roscoff.fr, julien.seiler@igbmc.fr

1

[Link](#)