

Introduction to the iPOP-UP HPC cluster

Julien Rey Olivier Kirsh Magali Hennion

March 2025



Table of Contents

- 1 Introduction
- 2 The iPOP-UP@RPBS HPC cluster
- 3 In practice
- 4 Slurm usage
- 5 Modules
- 6 Conclusion

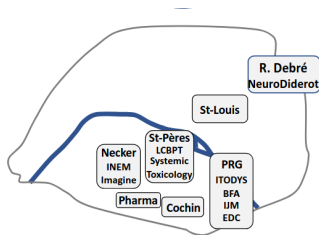


Who is this training for

- You work at Université Paris Cité
- You need (or might need) more computational power than you currently have
- You are familiar with Unix systems and Bash



The iPOP-UP project



- iPOP-UP: Integrative Platform for Omics Projects at Universit  de Paris
- Multisite Bioinformatics platform
- Ranging from multiple 'Omics' techniques to structural and chemo- *in silico* Bioinformatics
- Compute nodes

iPOP-UP : people involved

Scientific Manager

- Bertrand Cosson (EDC)

Executive committee

- Valérie Doye (IJM)
- Franck Letourneur (Cochin)
- Valérie Mezger (EDC)
- Pierre Tufféry (BFA)

Technical committee - 2025

- Nicolas Chevrollier (BFA)
- Yves Clément (IJM)
- Thomas Denecker (IFB)
- Magali Hennion (EDC)
- Olivier Kirsh (EDC)
- Nikos Konstantinides (IJM)
- Joël Marchand (IJM)
- Julien Rey (BFA)
- Benjamin Saintpierre (Cochin)
- Jacques Van Helden (IFB)



The RPBS platform

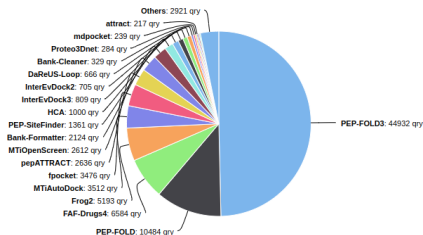
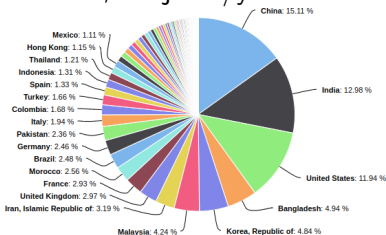
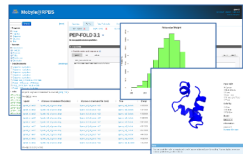


Missions:

- Development of Structural Bioinformatics methods
- Services deployment on the computing resource
- Expertise and training
- Hardware hosting

The RPBS Web portal

- <https://mobyline.rpbs.univ-paris-diderot.fr>: free access
- 30+ Structural Bioinformatics services
- 1 million CPU hours /year
- 80,000 jobs /year

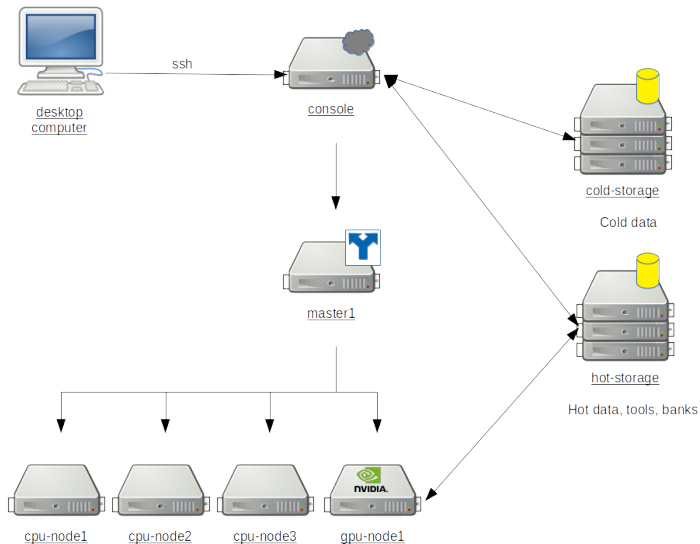


What is a HPC cluster for ?

- High hardware resources needs
- Long running analyses
- A lot of similar analyses
- Shared work between users
- Free your desktop from heavy tasks



What is a HPC cluster ?



Computational hardware

Partitions (groups of compute nodes):

- ipop-up: 16 nodes, 2048 CPUs, GPUs
- rpbs: 20 nodes, 832 CPUs, 9 GPUs (+ 9 nodes, 520 CPUs, 4 GPUs)
- cmpli: 6 nodes, 264 CPUs, 14 GPUs (+ 1 node, 64 CPUs, 2 GPUs)
- epigen: 4 nodes, 128 CPUs
- master-bi: 1 node, 32 CPUs, 3 GPUs

Storage:

- hot-storage: 125TB, very fast
- cold-storage: 240TB, slow + backup (soon)

Virtualization servers, etc...



Connexion

Go to your terminal and connect to the cluster using the following line, don't forget to replace username with your personal username.

```
ssh username@ipop-up.rpbs.univ-paris-diderot.fr
```

Type in your password and enter.

```
You@YourComputer:~/PathTo/RNAseqProject$ ssh username@ipop-up.rpbs.univ-paris-diderot.fr
#####
#
#                               #
#   iPOP-UP                   #
#                               #
#   Hosted by:                 #
#   Ressource Parisienne en Bioinformatique Structurale               #
#   -----                     #
#   All connections are monitored and recorded.                       #
#   Disconnect IMMEDIATELY if you are not an authorized user!        #
#                               #
#####
username@ipop-up.rpbs.univ-paris-diderot.fr's password:
Last login: Tue Jan 25 15:21:45 2022 from 172.28.18.162
Bienvenue sur le cluster iPOP-UP.

Pour toute question ou demande de support, rejoignez-nous sur le forum de RPBS : https://discourse.rpbs.univ-paris-diderot.fr

Pour changer le compte projet par défaut : sacctmgr update user $USER set defaultaccount=<project-name>
project1 [-----] 3 / 1024 GB
project2 [#-----] 913 / 10240 GB
Update: 2022-01-21 16:00 - default account in bold - More info: statusBars --help
[username@ipop-up ~]$
```



Note about ssh and security

Your IP will be banned after 5 failed authentication attempts. Each public key count as one authentication attempt. To disable public key authentication:

```
ssh -o PubkeyAuthentication=no rey@ipop-up.rpbs.univ-paris-diderot.fr
```

A good thing to do, change your password:

```
[rey@ipop-up ~]$ passwd
```



Connexion - JupyterHub

You can also access the cluster using the web interface JupyterHub. Open a web browser and go to

`https://jupyterhub.rpbs.univ-paris-diderot.fr`

The screenshot shows the 'Server Options' interface of JupyterHub. At the top, there is a navigation bar with 'jupyterhub Home Token' on the left and 'Home' and 'Logout' on the right. The main content area is titled 'Server Options' and contains several configuration fields:

- Project:** A dropdown menu with 'biodic' selected.
- Partition:** A dropdown menu with 'ipop-up' selected.
- CPUs:** A text input field containing the number '1'.
- Memory (in GB):** A text input field containing the number '2'.
- GPU:** A dropdown menu with '0' selected and 'A100 10.20GB' visible as an option.
- Time limit (in hours):** A text input field containing the number '6'.

At the bottom of the form, there is a prominent orange 'Start' button.

Type in your login and password and enter.

Select your project, the resources you need, and press start.



Connexion - JupyterHub

The screenshot shows the JupyterHub Launcher interface. On the left is a file browser for the directory `/formation_cluster/`. The main area is titled "Launcher" and contains three sections of launchable environments:

- Notebook:** Includes Python 3 (ipykernel), Bash, Colabfold 1.5.5, Python 3.9, R 4.2.1, and RStudio.
- Console:** Includes Python 3 (ipykernel), Bash, Colabfold 1.5.5, Python 3.9, and R 4.2.1.
- Other:** Includes Terminal, Text File, Markdown File, Python File, R File, and Show Contextual Help.

The file browser on the left shows the following files and their last modified dates:

Name	Last Modified
230612	9 months ago
corrections	9 months ago
slurm_output	last year
star_output	11 months ago
array_example.sh	last year
array_example2.sh	last year
flatter.sh	last year
star_4cpu.sh	last year
star.sh	last year
test_ok.sh	last year
test_ok2.sh	last year

Where you can go, write, or execute

User environments

`/shared/home/username`

Computations (hot data)

`/shared/projects/projectname`

Processed data (cold data)

`/cold-storage/username`

Data banks (read-only)

`/shared/banks/`

Note about quotas

We operate quotas to limit the amount of disk space a group or a project can use on the hot storage filesystem. To check available space for all your projects:

```
[rey@ipop-up ~]$ statusBars
alphafold [###-----]      847 /   5000 GB
training  [-----]         0 /   1024 GB
```

```
[rey@ipop-up ~]$ lfsquotas training
Disk quotas for grp 6011 (gid 6011):
      Filesystem  used   quota  limit  grace  files  quota  limit  grace
 /shared/projects/training 96.26M  1T     2T     -      4      0      0      -
gid 6011 is using default file quota setting
```

Jobs **must** be launched from a project directory.



About Slurm



Slurm is the job scheduling system.

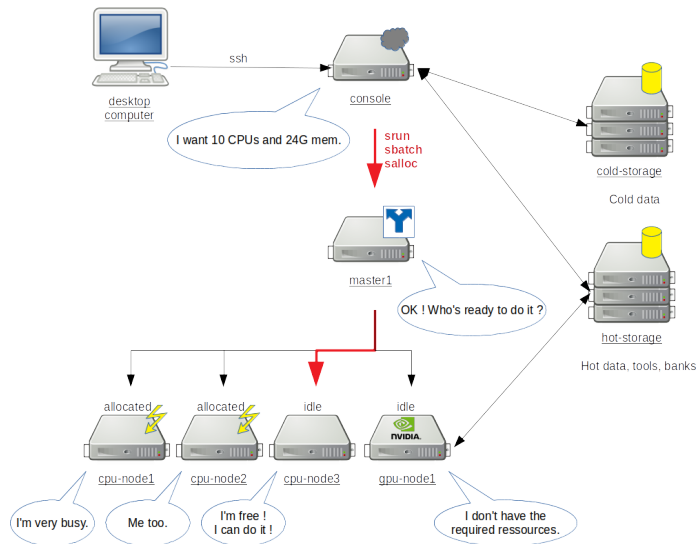
It is what will take your code and distribute it on the computing nodes, while ensuring they have the CPU(s) and RAM that you asked for.

It requires specific commands to run (srun, sbatch, salloc, etc...).

You **must** launch jobs with Slurm.



Flowsheet



srun

Launch a (simple) interactive job.

```
[rey@ipop-up ~]$ srun hostname
```

Some parameters can be added to the command line:

- partition**/–**p**: request a specific partition
- account**/–**A**: select the (project) account
- cpus-per-task**/–**c**: request that ncpus be allocated (default: 1 cpu)
- mem-per-cpu**: specify the required memory per cpu (default: 2GB)

Example:

```
[rey@ipop-up ~]$ srun -A training -p ipop-up -c 8 hostname
```



sbatch

Launch more complex jobs.

myscript.sbatch

```
#!/bin/bash
#SBATCH --partition=ipop-up
#SBATCH --account=training
#SBATCH --cpus-per-task=8
#SBATCH --mem-per-cpu=4GB
#SBATCH --output=resultat.log
hostname
```

Example:

```
[rey@ipop-up ~]$ sbatch myscript.sbatch
```

About associations

Cluster, account, partition and qos must match your user's associations.
To check your associations:

```
[rey@ipop-up ~]$ sacctmgr show user rey withassoc
```

User	Def Acct	Cluster	Account	Partition	QOS	Def QOS
rey	demo	production	training	ipop-up	normal	
rey	demo	production	demo	ipop-up	normal	
rey	demo	production	alphafold	cmpli	normal	
rey	demo	production	alphafold	rpbs	normal	



squeue

List submitted jobs on the cluster:

```
[rey@ipop-up ~]$ squeue
      JOBID PARTITION     NAME     USER  ST       TIME  NODES NODELIST(REASON)
    1008002      cmpli  argtoser domingue R    20:58:09      1 gpu-node14
    993325      cmpli  calcam-p  ghoula  R 6-20:23:17      1 gpu-node8
    993574      ipop-up amyloid_  meuret PD         0:00      1 (AssocGrpCPUMinutesLimit)
943019_[165-299]      ipop-up kappa_va badaoui PD         0:00      1 (JobArrayTaskLimit)
    943019_117      ipop-up kappa_va badaoui  R 7-17:55:27      1 cpu-node132
```

Some parameters can be added to filter jobs or show more infos:

- partition**/**-p**: specify the partition to view
- user**/**-u**: request jobs from a list of users
- format**/**-o**: specify the information to be displayed

Example:

```
[rey@ipop-up ~]$ squeue --me -p ipop-up
[rey@ipop-up ~]$ squeue -o "%10i %.9P %.8j %.12u %.15T %.15M %.15l %.6D %.20R %.6C %.10b %.10Q"
```



scancel

Kill a job

```
[rey@ipop-up ~]$ scancel job_id
```

Cancel all my jobs

```
[hennion@ipop-up ~]$ scancel -u <my_user_name>
```

or

```
[hennion@ipop-up ~]$ scancel --me
```



sacct

Display accounting data for your running and completed jobs:

```
[rey@ipop-up ~]$ sacct --format=JobID,JobName,Start,Elapsed,NCPUS,NodeList,ReqMeM,State --start
↵ 2023-06-01 --end 2023-06-30
```

JobID	JobName	Start	Elapsed	CPUTime	NCPUS	NodeList	ReqMem	State
1010778	sleep	2023-06-08T15:37:19	00:00:30	00:00:30	1	cpu-node133	2000Mc	COMPLETED



seff

Report a job's efficiency:

```
[rey@ipop-up ~]$ seff 981654
Job ID: 981654
Cluster: production
User/Group: lejal/cmpli
State: COMPLETED (exit code 0)
Cores: 1
CPU Utilized: 04:15:58
CPU Efficiency: 99.72% of 04:16:41 core-walltime
Job Wall-clock time: 04:16:41
Memory Utilized: 243.88 GB
Memory Efficiency: 97.01% of 251.40 GB
```



reportseff

Report a job efficiency

```
[hennion@ipop-up]$ module load reportseff
[hennion@ipop-up]$ reportseff 40913604
```

JobID	State	Elapsed	TimeEff	CPUEff	MemEff
40913604	CANCELLED	00:00:37	0.0%	16.2%	20.6%

Analyze a folder of slurm outputs

```
[hennion@ipop-up]$ reportseff --format "+Start,CPUTime,NCPUS,NodeList,MaxRSS,ReqMem" --modified-sort
↪ slurm_output/
```

JobID	State	Elapsed	TimeEff	CPUEff	MemEff	Start	CPUTime	NCPUS	NodeList	MaxRSS	ReqMem
multitqc--41907184.out	COMPLETED	00:00:07	2.3%	57.1%	1.0%	2024-09-24T12:23:17	00:00:07	1	cpu-node-45	3184K	191M
fastqc-sample-SRR3099585_chr18-41907170.out	COMPLETED	00:00:25	8.3%	64.8%	1.1%	2024-09-24T12:22:47	00:00:25	1	cpu-node-38	3284K	286M
fastqc-sample-SRR3105697_chr18-41907180.out	COMPLETED	00:00:15	5.8%	80.8%	1.1%	2024-09-24T12:22:47	00:00:15	1	cpu-node-39	3268K	286M
fastqc-sample-SRR3099587_chr18-41907181.out	COMPLETED	00:00:14	4.7%	85.7%	1.1%	2024-09-24T12:22:47	00:00:14	1	cpu-node-48	3280K	286M
fastqc-sample-SRR3105698_chr18-41907183.out	COMPLETED	00:00:13	4.3%	69.2%	1.1%	2024-09-24T12:22:47	00:00:13	1	cpu-node-48	3192K	286M
fastqc-sample-SRR3099586_chr18-41907183.out	COMPLETED	00:00:12	4.0%	66.7%	1.1%	2024-09-24T12:22:47	00:00:12	1	cpu-node-45	3288K	286M
fastqc-sample-SRR3105699_chr18-41907182.out	COMPLETED	00:00:12	4.0%	66.7%	1.1%	2024-09-24T12:22:47	00:00:12	1	cpu-node-48	3268K	286M
multitqc--41907169.out	OUT OF MEMORY	00:00:06	2.8%	16.7%	3.3%	2024-09-24T12:21:06	00:00:06	1	cpu-node-45	3240K	95M
fastqc-sample-SRR3099585_chr18-41907159.out	COMPLETED	00:00:33	11.8%	54.5%	76.7%	2024-09-24T12:20:25	00:00:33	1	cpu-node-38	22178K	286M
fastqc-sample-SRR3105697_chr18-41907169.out	COMPLETED	00:00:19	6.3%	73.7%	1.1%	2024-09-24T12:20:25	00:00:19	1	cpu-node-39	3212K	286M
fastqc-sample-SRR3105698_chr18-41907163.out	COMPLETED	00:00:17	5.7%	82.4%	1.1%	2024-09-24T12:20:25	00:00:17	1	cpu-node-48	3224K	286M
fastqc-sample-SRR3099587_chr18-41907161.out	COMPLETED	00:00:15	5.0%	80.0%	1.1%	2024-09-24T12:20:25	00:00:15	1	cpu-node-48	3192K	286M
fastqc-sample-SRR3105699_chr18-41907162.out	COMPLETED	00:00:14	4.7%	71.4%	1.1%	2024-09-24T12:20:25	00:00:14	1	cpu-node-48	3244K	286M
fastqc-sample-SRR3099586_chr18-41907159.out	COMPLETED	00:00:15	5.0%	53.3%	1.1%	2024-09-24T12:20:25	00:00:15	1	cpu-node-45	3304K	286M
multitqc--41907141.out	COMPLETED	00:00:24	8.3%	25.8%	8.0%	2024-09-24T12:15:57	00:00:24	1	cpu-node-38	3228K	588M
fastqc-sample-SRR3105698_chr18-41907138.out	COMPLETED	00:00:18	8.2%	72.2%	8.0%	2024-09-24T12:15:35	00:00:18	1	cpu-node-34	3244K	588M
fastqc-sample-SRR3105699_chr18-41907137.out	COMPLETED	00:00:16	8.2%	68.8%	8.0%	2024-09-24T12:15:35	00:00:16	1	cpu-node-34	3228K	588M
fastqc-sample-SRR3099586_chr18-41907133.out	COMPLETED	00:00:15	8.2%	66.7%	8.0%	2024-09-24T12:15:35	00:00:15	1	cpu-node-39	3144K	588M
fastqc-sample-SRR3099585_chr18-41907134.out	COMPLETED	00:00:15	8.2%	73.3%	8.0%	2024-09-24T12:15:35	00:00:15	1	cpu-node-48	3272K	588M
fastqc-sample-SRR3099587_chr18-41907136.out	COMPLETED	00:00:14	8.2%	71.4%	8.0%	2024-09-24T12:15:35	00:00:14	1	cpu-node-48	3288K	588M
fastqc-sample-SRR3105697_chr18-41907135.out	COMPLETED	00:00:15	8.2%	66.7%	8.0%	2024-09-24T12:15:35	00:00:15	1	cpu-node-48	3232K	588M



Job arrays

Job arrays offer a mechanism for launching a lot of tasks at the same time. Each task of the job will have the environment variable `SLURM_ARRAY_TASK_ID` set to its array index value.

`myscript.sbatch`

```
#!/bin/bash
#SBATCH --partition=ipop-up
#SBATCH --account=training
#SBATCH --output=resultat_%a.log
#SBATCH --array=1-3
case "$SLURM_ARRAY_TASK_ID" in
  1) fruit='orange';;
  2) fruit='apple';;
  3) fruit='banana';;
esac
echo $fruit
```

Job arrays

```
[rey@ipop-up ~]$ sbatch myscript.sbatch
```

Results:

```
[rey@ipop-up ~]$ ls
resultat_1.log resultat_2.log resultat_3.log
[rey@ipop-up ~]$ tail resultat_*
==> resultat_1.log <==
orange

==> resultat_2.log <==
apple

==> resultat_3.log <==
banana
```

Job arrays

Another example:

myscript.sbatch

```
#!/bin/bash
#SBATCH --partition=ipop-up
#SBATCH --account=training
#SBATCH --output=resultat_%a.log
#SBATCH --array=0-2000%50
#SBATCH --cpus-per-task=2
INPUTS=(*.fq.gz)
fastqc ${INPUTS[$SLURM_ARRAY_TASK_ID]}
```

Job arrays

```
#SBATCH --array=0-15
```

= 16 jobs (\$SLURM_ARRAY_TASK_ID: from 0 to 15 included).

```
#SBATCH --array=10-16:2
```

= 4 jobs (\$SLURM_ARRAY_TASK_ID: 10,12,14,16).

```
#SBATCH --array=2,3-7:2,11,13
```

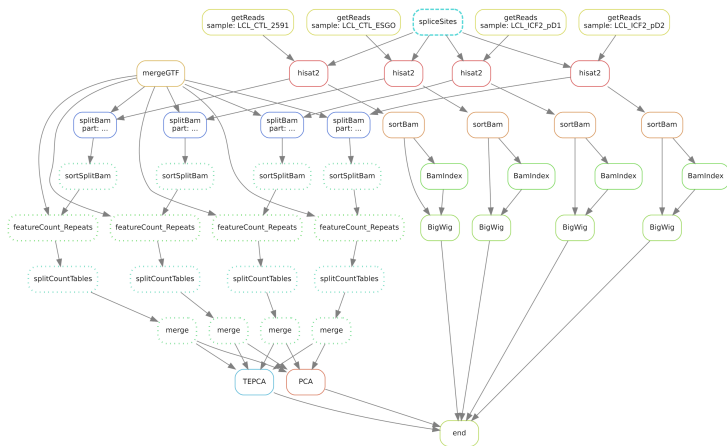
= 6 jobs (\$SLURM_ARRAY_TASK_ID: 2,3,5,7,11,13).

```
#SBATCH --array=1-10000%32
```

= 10 000 jobs, max 32 simultaneous jobs



Complex workflows



Use workflow managers such as Snakemake or Nextflow.

Modules



Tools are installed on the cluster in virtual environments

- each tool has its own dependencies (libraries) and it's not possible to make them all coexist in the same environment
- reproducibility: keep several versions of a tool



Conda environments



Containers (Apptainer)



Modules

They can be loaded with the module command.

Look for the different versions of multiqc:

```
[rey@ipop-up ~]$ module avail multiqc  
multiqc/1.3  multiqc/1.6  multiqc/1.7  multiqc/1.9
```

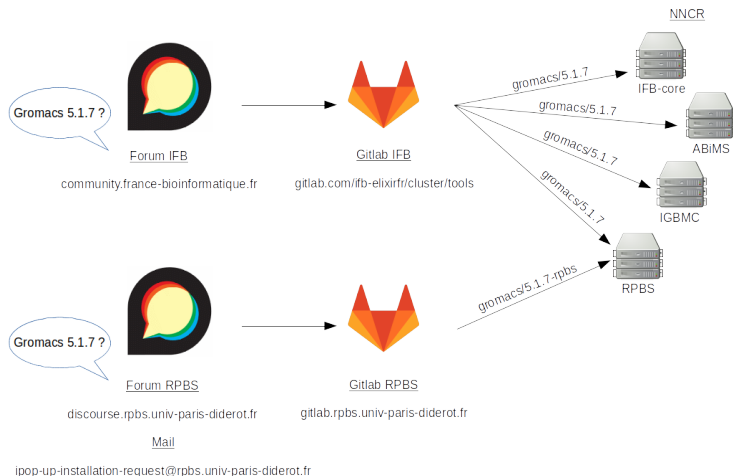
Load an environment:

```
[rey@ipop-up ~]$ module load multiqc/1.9
```

List loaded environments:

```
[rey@ipop-up ~]$ module list  
Currently Loaded Modulefiles:  
  1) multiqc/1.9    2) blast/2.13.0
```

How tools are installed



Useful resources

To find out more, the SLURM manual : `man sbatch` or
<https://slurm.schedmd.com/sbatch.html>

Ask for help or signal problems on the cluster :
<https://discourse.rpbs.univ-paris-diderot.fr/>

iPOP-UP cluster documentation:
<https://ipop-up.docs.rpbs.univ-paris-diderot.fr/documentation/>



Thanks



Alix Silvert

iPOP-UP past and present
technical and steering
committees



Exercises

<https://tinyurl.com/ipop-up-2025>



https://parisepigenetics.github.io/bibs/training/cluster_training_2503