DMC1 pipeline training

Session 1

DECEMBER 8

PAULINE

Program of this session

🐾 Introduction to IGH cluster

Connexion to IGH cluster

Navigate into the cluster

Create a conda environment

Open an interactive session

Use bedtools to sort and intersect bed files

Use Nextflow to run a small pipeline

Prepare SSDS nextflow pipeline inputs

Run SSDS nextflow pipeline

Check-up before start

WiFi ? (eduroam)

VPN ?

MobaXterm ?

Gitlab ?

Introduction to IGH cluster

IGH cluster : No graphical interface → access and communication using command line.

From your computer (VPN), you connect to **lakitu** (login node) via ssh protocol. Then you can access **shenron** or **luffy** (computing nodes) using a **scheduler** (slurm). There are 2 accessible queues for computing : **computepart** and **debug**

(Scheme)

- Access **Restricted** : each user needs a login and a password to access the cluster.

and Internal : you need to be connected though IGH network to access (from the IGH or through IGH VPN)

- **Resources** : each user has a dedicated space to work on the cluster, including :

home (/`home/username`) directory with **50 Mo storage**

work (/`work/username` also accessible via `/home/username/work`) directory with **1To storage**

It's recommended to work on the `/work` directory, but it is not made for long term storage.

Online documentation for the cluster : here

Connexion to IGH cluster

⚠ We will use my identifiers and my workspace for this training session.

I set up a temporary password for you to connect

Identifier : `demassyie`

Password : `ilovedmc1`

Connect to the cluster using :

```
ssh demassyie@lakitu.igh.cnrs.fr
```

Then enter the password.

It’s normal not to see the password (for safety !)

Navigate into the cluster

1. **Path and tree of files from root (Unix system)**

Test of the principal commands, and spy on my file organization

```
pwd – ls (options) - cd – date – man
```

2. **Common resources on the cluster**

Common common → accessible to all IGH `/poolzfs`

De massy common → accessible to De Massy team members `/work/commun/demassyTeam`

3. Storage check-up

Remember to check regularly your storage capacity

```
du -csh
```

Navigate into the cluster

4. Create your own project directory in `/work/demassyie/20211207_DMC1_training/firstname`

```
        mkdir
```

As we saw in 1, you can create different subfolders in your project directory.

Then copy the 2 bed files located in `/work/commun/demassyTeam/DMC1_training/` somewhere logical for you, in your project directory.

`touch - cat - cp – mv – head – tail – wc – rm` ☠

`grep – cut - > - echo – vim - scp`

`history` → very important to save your work today

Questions : How many peaks in the file ?

- How many peaks located in chromosome 15 ?

  How many peaks located in all chromosome BUT chromosome 15 ?

  How big are the files ?

  How many columns ?

Extract chromosome 15 peaks into a new file and download this file on your own computer.

Remove the copied bed files and replace them with a symbolic link `ln -s`

Now let’s get serious

5. Create your conda environment. Remember, conda is a package manager that will help managing tools dependancies.

(conda is already installed in my userspace)

```
conda env create –n firstname
```

```
conda activate firstname
```

Now we want to dig into these bed files. We will need to use bedtools https://anaconda.org/bioconda/bedtools

```
conda install –c bioconda bedtools
```

Now let’s get serious

6. Open an interactive job session

For now, we are still working on lakitu, and not on shenron nor luffy. It’s ok for small operations but nor for big ones that requires more resources.

To connect to shenron or luffy (the computing nodes), we can use the interactive mode

```
srun –pty bash
```

Can you tell which node you are connected to ?

And now we can use bedtools.

Usually, to use a tool, you need to write a command line looking like :

```
[tool-command] [input_files] [parameters] > [output_file]
```

Command line bedtools

Let's imagine you have a BED file of ChiP-seq peaks from two different experiments. You want to identify peaks that were observed in *both* experiments (requiring 50% reciprocal overlap) and for those peaks, you want to find to find the closest, non-overlapping gene. Such an analysis could be conducted with two, relatively simple bedtools commands.

1. Intersect the peaks from both sequencing platform.

Parameter -f 0.50 combined with -r requires 50% reciprocal overlap between the peaks from each experiment.

```
$ bedtools sort –i exp1.bed > exp1_sorted.bed
$ bedtools intersect -a exp1.bed -b exp2.bed -f 0.50 -r > both.bed
```

2. Find the closest, non-overlapping gene for each interval where both experiments had a peak

Parameter -io ignores overlapping intervals and returns only the closest, non-overlapping interval (in this case, genes)

```
$bedtools closest -a both.bed -b genes.bed -io > both.nearest.genes.txt
```

https://bedtools.readthedocs.io/en/latest/

Command line bedtools

Here is the header for bed file

1.chrom - Name of the chromosome (or contig, scaffold, etc.).

2.chromStart - The starting position of the feature in the chromosome or scaffold. The first base in a chromosome is numbered 0.

3.chromEnd - The ending position of the feature in the chromosome or scaffold. The *chromEnd* base is not included in the display of the feature. For example, the first 100 bases of a chromosome are defined as*chromStart=0, chromEnd=100*, and span the bases numbered 0-99.

4.name - Name given to a region (preferably unique). Use '.' if no name is assigned.

5.score - Indicates how dark the peak will be displayed in the browser (0-1000). If all scores were '0' when the data were submitted to the DCC, the DCC assigned scores 1-1000 based on signal value. Ideally the average signalValue per base spread is between 100-1000.

6.strand - +/- to denote strand or orientation (whenever applicable). Use '.' if no orientation is assigned.

7.signalValue - Measurement of overall (usually, average) enrichment for the region.

8.pValue - Measurement of statistical significance (-log10). Use -1 if no pValue is assigned.

9.qValue - Measurement of statistical significance using false discovery rate (-log10). Use -1 if no qValue is assigned.

10.peak - Point-source called for this peak; 0-based offset from chromStart. Use -1 if no point-source called.

How would you do to add the header to the bed files ?

Does it work with bedtools ?

Nextflow pipeline bedtools

7. Nextflow

Git the nextflow pipeline :

```
git clone https://gitlab.igh.cnrs.fr/pauline.auffret/bedtools_training.git
```

Tree of nextflow project ; main.nf ; nextflow.config ; conf/igh etc

Nextflow work directory ; Comment char //

Run it !

sque

-resume

Easier, right ?

Play with parameters (command-line or config file etc)

See the logs and Nextflow Tower

SSDS Nextflow pipeline

Go into ssdsnextflowpipeline.

How many lines for main.nf ?

README

What are the input files for the pipeline ?

What do we need to set up ?

Inspect the `run_pipeline.sh` script : what is it for ?

Bash run_pipeline.sh -h