

## 1 - Single execution on a cluster

Goals : Identify genes of a transcript fasta file thanks to the alignment software blast (NCBI)

### Exercise n°1 : Download data

1. Start your machine and open a terminal (putty for window). You can now try to access the genotoul server by using ssh : ssh -X [user\\_name@genologin.toulouse.inra.fr](mailto:user_name@genologin.toulouse.inra.fr)
2. Create in work directory a directory named `cluster` move to it
3. Download the transcript file:  
`http://genoweb.toulouse.inra.fr/~formation/cluster/data/contigs.fasta.gz`
4. Connect to a node in interactive mode.



*When you connect on the cluster in interactive mode you are systematically placed in your home directory*

5. Un-compress the file.



*Manipulating files (compress, zip...) can use a lot of resources, it's necessary to perform it on the cluster.*

6. Display the ten first lines of “contigs.fasta” file. Which is the format file ? Which is the kind of data ?

### Exercise n°2 : Use simple submission command: use of NCBI\_Blast+

1. Load the module: `module load bioinfo/ncbi-blast-2.6.0+`
2. Launch a blast against “ensembl\_danio\_rerio” in interactive mode on the cluster. Your query is genomic, your database is proteic so you need a blastx program. Set the evalue at 10e-10.

**Syntax :** `blastx -query <file.fa> -db <dbname or path> -evalue <evalue> -out <output_file>`



*For more help on blast, type `blastx -help`*



*On Genologin Cluster, ncbi blast databases are available in `/bank/blastdb`, but you don't need to specify the path.*

## Cluster practice - solution


3. Open a new terminal and check all the jobs running or waiting on the cluster. Check your own job.

What is your priority ? On which node are you running ?

4. Kill your job.
5. Use a text editor to create a command file `cmd.txt` with the same module load and the same blast command line (but with a **blastn** instead of `blastx`). The first line of the file is :  
`#!/bin/sh`  
Launch it in batch mode (for the practice **only**, specify the queue `testq : -p testq`)
6. Check the execution. When it's over, look at the blast output file and the execution trace file (`slurm-xxxxx.out`). Has the job finished correctly ?
7. Launch the same command without using a file ( option `--wrap="command"` )  
Check the execution. When it's over, look at the blast output file and the execution trace file (`slurm-xxxxx.out`). Has the job finished correctly ?
8. If you didn't have any error until now, redo the previous submission with an error in the command. Have a look to the trace file.


## 2 – Array of jobs

1. Split the fasta file in 10 fasta files into a directory called `out_split`



```
module load bioinfo/exonerate-2.2.0
fastasplit <path> <dirpath>
Sequence Input Options:
-----
-f --fasta [mandatory] <*** not set ***>
-o --output [mandatory] <*** not set ***>
-c --chunk [2]
```

2. Check the number of files
3. Check if the number of sequences in “contigs.fasta” file correspond to the sum of all sequences in splitted files.
4. Create a command file (cmds.txt) for the job array with one blast command per fasta file.
5. Check the syntax.



The good practice is to check that there is not a syntax error.

To check it, we propose to execute the first line in interactive mode (use the `srun --pty bash` for that).

Cut and paste the first line of the file in the terminal.

As soon as you seen that there is no syntax error, you can kill the process (by using `ctrl+c`).

6. Launch the job array by requesting 2GB of memory per job. Check the execution, how many jobs are running simultaneously ?
7. After execution check trace files.
8. Concat all blast result in one file.

## 3 – Parallel environment

1. Launch a blastx of all the contigs against ensembl\_danio\_rerio with 8 threads on the same node.
2. Check the execution in detail. (squeue –help to see format options or use alias sq\_long -u <username>)
3. Use in the same time the job array and the parallel execution.
  1. Split multifasta in a new directory.
  2. Build a command file with blastx command and option -num\_thread 8
  3. Launch the job array with the --cpus-per-task option.