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

2. Create in work directory a directory named `cluster` move to it
```
$ cd work/
$ mkdir cluster
$ cd cluster/
```

3. Download the transcript file:
http://genoweb.toulouse.inra.fr/~formation/cluster/data/contigs.fasta.gz
```
$ wget
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
```
srun --pty bash
```

5. Un-compress the file.
```
gunzip contigs.fasta.gz
```

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 ?
```
head contigs.fasta
contigs.fasta  is a fasta file because his format starts with (“>”)
header sequence
It's correspond to sequence data. It could be nucleic or proteic.
Here we have a nucleic file.
```

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`

```
blastx -query contigs.fasta -db ensembl_danio_rerio -evalue 10e-10 -out
contigs.dr_prot
```

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

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

```
squeue
squeue -t R
squeue -t PD
squeue -u <username>
```

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

```
squeue -u mtrotard
```

```
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           1232823     workq     bash     user  R       0:05      1 node129
```

4. Kill your job.

```
scancel 1232823
```

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`)
   cmd.txt contains :

```
#!/bin/sh
module load bioinfo/ncbi-blast-2.6.0+
blastn -db ensembl_danio_rerio -query contigs.fasta -evalue 10e-10
-out contigs.dr_nuc
```

   Launch it with :

```
sbatch -p testq cmd.txt
```

6. Check the execution. When it's over, look at the blast output file and the 2 execution trace files (`slurm-xxxxx.out`). Has the job finished correctly ?

```
squeue -u <username>
more contigs.dr_nuc
more  slurm-XXXXX.out
```

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 ?

```
sbatch -p testq -J blastdr –wrap='module load bioinfo/ncbi-blast-
2.6.0+;blastn -db ensembl_danio_rerio -query contigs.fasta -evalue 10e-10
-out contigs.dr_nuc_command_line'
```

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]
```

```
mkdir out_split
module load bioinfo/exonerate-2.2.0
fastasplit -f contigs.fasta -c 10 -o out_split
```

2. Check the number of files

```
ls out_split/* | wc
```

3. Check if the number of sequences in “contigs.fasta” file correspond to the sum of all sequences in splitted files.

```
grep “>” out_split/* | wc
grep -c ">" contigs.fasta
```

4. Create a command file (cmds.txt) for the job array with one blast command per fasta file.

```
for f in `ls out_split/*`;do echo “module load bioinfo/ncbi-blast-
2.6.0+;blastx -db ensembl_danio_rerio -query $f -evalue 10e-10
 -out $f.blast” >> cmds.txt;done
```

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 qrsh 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) .

```
srun -p testq --pty bash
module load bioinfo/ncbi-blast-2.6.0+
blastx -db ensembl_danio_rerio -query out_split/contigs.fasta_chunk_0000000
-evalue 10e-10 -out out_split/contigs.fasta_chunk_0000000.blast

crtl+C
```

6. Launch the job array by requesting 2GB of memory per job. Check the execution, how many jobs are running simultaneously ?

```
sarray -p testq --mem=2G cmds.txt
squeue -u <username>
```

7. After execution check trace files.

```
ls slurm-<jobid>_*.out
cat slurm-<jobid>_*.out
```

8. Concat all blast result in one file.
```
cat out_split/*.blast > result.blast
```

3 – Parallel environment

1. Launch a blastx of all the contigs againt ensembl_danio_rerio with 8 threads on the same node.
```
sbatch -p testq --cpus-per-task 8 -J blastdr --wrap='module load
bioinfo/ncbi-blast-2.6.0+;blastx -num_threads 8 -db ensembl_danio_rerio
-query contigs.fasta -evalue 10e-10 -out contigs.dr_nuc'
```

2. Check the execution in detail
```
squeue -u <username> -t R -O "jobid:11,name:30,username:15,partition:12,
numnodes:8,numcpus:8,minmemory:12,timelimit:15,timeleft:15,state:12,nodelist:20"
```

```
JOBID       NAME                            USER            PARTITION    NODES    CPUS
MIN_MEMORY  TIME_LIMIT     TIME_LEFT       STATE        NODELIST
1234567     blastdr                         mtrotard        testq        1        8
4000M        180-00:00:00    179-23:59:50    RUNNING      node147

or 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.
```
mkdir out_split_pe
module load bioinfo/exonerate-2.2.0
fastasplit -f contigs.fasta -c 10 -o out_split_pe
```

    2. Build a command file with blastx command and option -num_thread 8
```
for f in `ls out_split_pe/*`;do echo "module load bioinfo/ncbi-blast-
2.6.0+;blastx -db ensembl_danio_rerio -query $f -evalue 10e-10
 -out $f.blast -num_threads 8" >> cmds_pe.txt;done
```

    3. Launch the job array with the --cpus-per-task option.
```
sarray -J blast_ja_pe --cpus-per-task 8 cmds_pe.txt
```