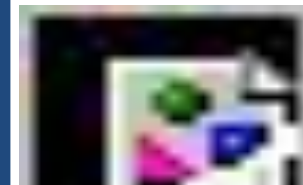
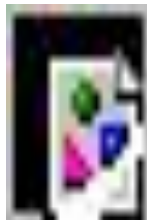


Session de formation 2018



- 12 Mars** Guide de survie à Linux : les commandes de base pour débuter sur un serveur linux
- 13 Mars** Linux avancé : manipuler et filtrer des fichiers sans connaissance de programmation
- 15 Mars** **Initiation à l'utilisation du cluster bioinformatique itrop**
- 23 Mars** Initiation aux gestionnaires de workflow South Green: Galaxy ou TOGGLE
- 26 Mars** Initiation aux analyses de données transcriptomiques
- 05 Avril** Initiation à git



South Green
bioinformatics platform





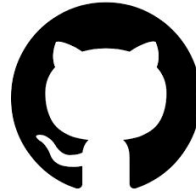
South Green
bioinformatics platform



plateau i-trop



www.southgreen.fr



<https://github.com/SouthGreenPlatform>



The South Green portal: a comprehensive resource for tropical and Mediterranean crop genomics, Current Plant Biology, 2016

Session de formation 2018



- Toutes nos formations :
<https://southgreenplatform.github.io/trainings/>
- Topo & TP : [HPC IRD](#)
- Environnement de travail : [Logiciels à installer](#)

Initiation HPC cluster

www.southgreen.fr

<https://southgreenplatform.github.io/trainings>



Objectif

Acquérir les bonnes pratiques pour utiliser le cluster de calcul Itrop !

Applications

- Connaître l'architecture du cluster
- Connaître le rôle des différentes partitions
- Utiliser SGE (qusb, qrsh, qhost, qacct, qstat, qqdel)
- Utiliser les modules environment
- Faire du scripting de base

ARCHITECTURE

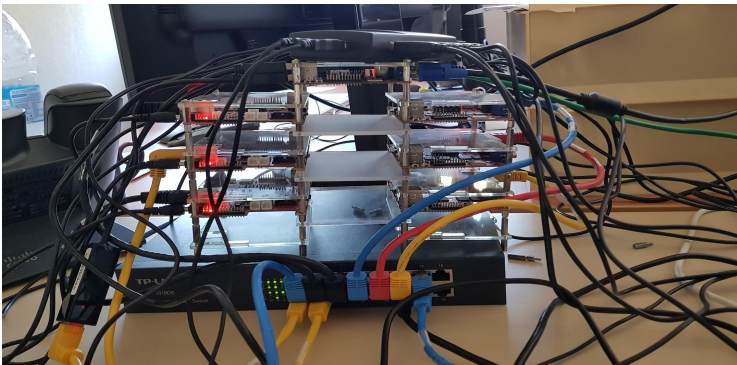
- Un cluster est une unité logique constituée de plusieurs serveurs
- Agit comme une unique machine puissante
- Permet d'obtenir une puissance de calcul élevée
- Une plus grande capacité de stockage
- Une fiabilité supérieure
- Une plus grande disponibilité des ressources

- Un cluster est une unité logique constituée de plusieurs serveurs
- Agit comme une unique machine puissante
- Permet d'obtenir une puissance de calcul élevée
- Une plus grande capacité de stockage
- Une fiabilité supérieure
- Une plus grande disponibilité des ressources



Qu'est ce qu'un cluster?

- Un cluster est une unité logique constituée de plusieurs serveurs
- Agit comme une unique machine puissante
- Permet d'obtenir une puissance de calcul élevée
- Une plus grande capacité de stockage
- Une fiabilité supérieure
- Une plus grande disponibilité des ressources



- Nœud maître : Ordonnanceur.
Gère les ressources et les priorités des jobs
- Nœuds de calcul : Ressources (CPU ou mémoire RAM) utilisées par le master

CALCUL



- Nœud maître : Ordonnanceur.
Gère les ressources et les priorités des jobs
- Nœuds de calcul : Ressources (CPU ou mémoire RAM) utilisées par le master
- Serveur(s) NAS :
Stockent les données utilisateurs et les résultats d'analyses

CALCUL



STOCKAGE





bioinfo-master
.ird.fr
91.203.34.148

Rôle : Lancer et prioriser les jobs sur les nœuds de calcul

Accessible depuis Internet

Connexion : `ssh login@bioinfo-master.ird.fr`



**bioinfo-master.
ird.fr**
91.203.34.148

Rôle : Lancer et prioriser les jobs sur les nœuds de calcul

Accessible depuis Internet

Connexion : ssh login@bioinfo-master.ird.fr



**22 noeuds
avec
bioinfo-inter.ir
d.fr**
91.203.34.150

Rôle : Utilisés par le maître pour exécuter des jobs

Pas accessibles depuis Internet

node0 à node22

Noeud interactif : bioinfo-inter.ird.fr

Connexion : ssh login@bioinfo-inter.ird.fr



**bioinfo-master.
ird.fr**
91.203.34.148

Rôle : Lancer et prioriser les jobs sur les nœuds de calcul

Accessible depuis Internet

Connexion : `ssh login@bioinfo-master.ird.fr`



**22 noeuds
avec
bioinfo-inter.ird.
fr**
91.203.34.150

Rôle : Utilisés par le maître pour exécuter des jobs

Pas accessibles depuis Internet

node0 à node22

Noeud interactif : `bioinfo-inter.ird.fr`

Connexion : `ssh login@bioinfo-inter.ird.fr`



bioinfo-nas3.ird.fr
91.203.34.180



Rôle : Stocker les données utilisateurs

Accessibles depuis Internet

Connexion : `filezilla` ou `scp`

bioinfo-nas.ird.fr
91.203.34.157

bioinfo-nas2.ird.fr
91.203.34.160

/home : Votre répertoire personnel
Quota 100Go
Hébergée sur : bioinfo-nas.ird.fr
Partagée sur toutes les machines

/teams : Données projet partagées
entre plusieurs utilisateurs
d'une même équipe
Quota 200Go
Hébergée sur : bioinfo-nas.ird.fr
Partagée sur toutes les machines

/data2 : Données projet partagées
entre plusieurs utilisateurs
Quota 500Go à 1To
Hébergée sur : bioinfo-nas.ird.fr
Partagée sur toutes les machines

/home : Votre répertoire personnel
Quota 100Go
Hébergée sur : bioinfo-nas.ird.fr
Partagée sur toutes les machines

/data : Données projet partagées
entre plusieurs utilisateurs
Quota 500Go à 1To
Hébergée sur : bioinfo-nas2.ird.fr
Partagée sur toutes les machines

/team : Données projet partagées
entre plusieurs utilisateurs
d'une même équipe
Quota 200Go
Hébergée sur : bioinfo-nas.ird.fr
Partagée sur toutes les machines

/data3 : Données projet partagées
entre plusieurs utilisateurs
Quota 500Go à 1To
Hébergée sur : bioinfo-nas3.ird.fr
Partagée sur toutes les machines

/data2 : Données projet partagées
entre plusieurs utilisateurs
Quota 500Go à 1To
Hébergée sur : bioinfo-nas.ird.fr
Partagée sur toutes les machines

/home : Votre répertoire personnel
Quota 100Go
Hébergée sur : bioinfo-nas.ird.fr
Partagée sur toutes les machines

/data : Données projet partagées
entre plusieurs utilisateurs
Quota 500Go à 1To
Hébergée sur : bioinfo-nas2.ird.fr
Partagée sur toutes les machines

/team : Données projet partagées
entre plusieurs utilisateurs
d'une même équipe
Quota 200Go
Hébergée sur : bioinfo-nas.ird.fr
Partagée sur toutes les machines

/data3 : Données projet partagées
entre plusieurs utilisateurs
Quota 500Go à 1To
Hébergée sur : bioinfo-nas3.ird.fr
Partagée sur toutes les machines

/data2 : Données projet partagées
entre plusieurs utilisateurs
Quota 500Go to 1To
Hébergée sur : bioinfo-nas.ird.fr
Partagée sur toutes les machines

/scratch : Répertoire temporaire de travail
1To à 5To
Hébergée sur : **chaque noeud**
Pas partagée mais uniquement en local
Données conservées **3 semaines**



SUN GRID ENGINE (SGE)

- SGE (SUN Grid Engine) est un gestionnaire de ressources de calcul sous linux, capable de gérer de deux à des milliers de serveurs et des centaines de clusters de plusieurs nœuds à la fois.
- Un outil opensource
- 3 fonctions principales :
 - Alloue les ressources (CPU, RAM) aux utilisateurs pour qu'ils puissent lancer leurs analyses
 - Fournit un cadre pour lancer, exécuter et monitorer les jobs sur l'ensemble des nœuds alloués
 - Gère la priorité des jobs en file d'attente

Bioinfo.q : queue par défaut
Noeuds: node2, node8, node9, node10,
node11,node12,node13,
,node14,node15,node16,node17,
node19,node20
RAM: de 48Go à 64Go
Cœurs: de 12 à 20 cœurs

dynadiv.q : priorité pour l'équipe
dynadiv
Noeuds: node2, node10
RAM: 48Go
Cœurs: 12 cœurs
/scratch de 5To pour node10

dynadiv.q : priorité pour thomas
Couvreur
Noeuds: node20
RAM: 64Go
Cœurs: 20 cœurs

r900.q : queue avec noeud **DELL**
Noeuds: node5, node21
RAM: 32Go
Cœurs: 16 cœurs

longjob.q : jobs longs ou > à 10 jobs
Noeuds: node0, node1, node11
RAM: 48Go
Cœurs: 12 cœurs

alizon.q : priorité pour l'équipe de
samuel Alizon
Noeuds: node8, node9, node12
RAM: 48Go
Cœurs: 12 cœurs

alizon.q : priorité pour le logiciel
smrtportal
Noeuds: node17, node18
RAM: 64Go
Cœurs: 12 cœurs

Actions

- Réserver un coeur sur un nœud de manière interactive
- Réserver un coeur sur un noeud en particulier
- Réserver X coeur sur un noeud

Commandes

\$ **qrsh**

\$ **qrsh -l hostname=nodeX**

Avec X le numéro du noeud

\$~ **qrsh -pe ompi X**

Avec X : le nombre de processeurs de 0 à 12

Actions

- Lancer un script en mode batch
- Propager l'environnement chargé au noeud
- Donner un nom à votre job
- Utiliser plusieurs processeurs
- Demander un certain montant de RAM
- Demander un noeud en particulier
- Lancer directement une commande avec qsub

Commandes

`$qsub + script.sh`

`$qsub -V script.sh`

`$~ qsub -N job_name script.sh`

`$~ qsub -pe ompi X script.sh`

Avec X le nombre de coeurs à utiliser

`$~ qsub -l mem_free=XG script.sh`

Avec X le montant de mémoire à réserver

`$~ qsub -l hostname=nodeX script.sh`

`$~ qsub -b y command`

Actions

- Informations sur l'état des noeuds
- Voir ses jobs en cours
- Informations sur les jobs lancés
- Informations sur les jobs terminés
- Informations globales sur les queues

Commandes

\$ qhost

\$~ qstat

\$~ qstat -j <JOB_ID>

With JOB_ID :the job number

\$~ qacct -j <JOB_ID>

With JOB_ID :the job number

\$~ qstat -g c

Actions

- Suppression d'un job

Commandes

`$~ qdel <JOB_ID>`

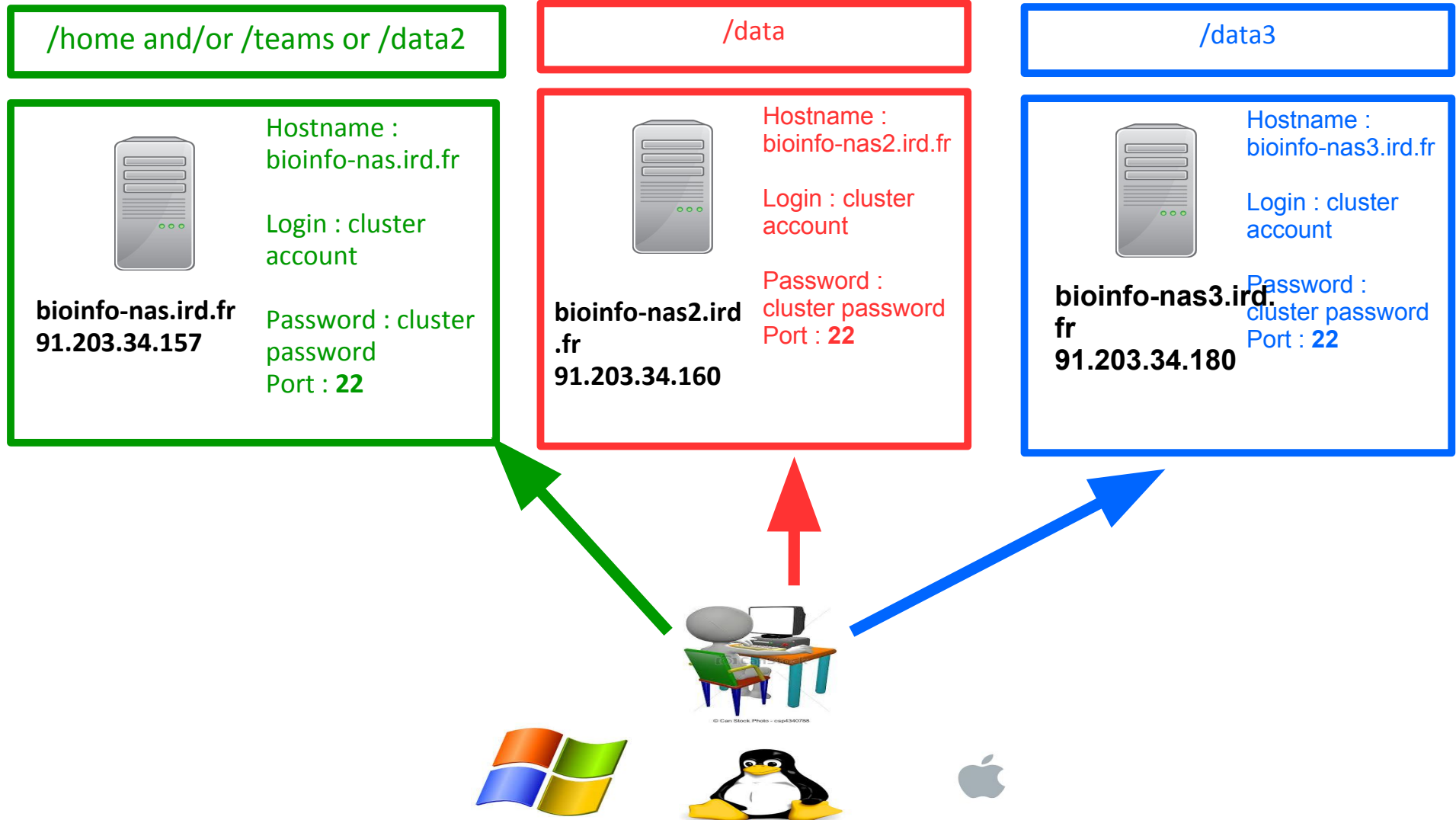
avec JOB_ID : l'identifiant du job

1



**TP: Lancer une analyse blast
de manière interactive**

Transferts de données sur le cluster itrop



Ouvrir filezilla et récupérer le fichier « HPC_french.pdf »
dans /data/projects/tp-cluster/training_2018

Ouvrir filezilla et récupérer le fichier « HPC_french.pdf »
dans /data/projects/tp-cluster/training_2018

Renseigner les paramètres suivants :

Hostname : **bioinfo-nas2.ird.fr**

Login : votre login

Mot de passe : votre login

Port :**22**

Naviguer dans la fenêtre de droite jusqu'à /data/projects/tp-cluster/training_2018

Récupérer le fichier HPC_french.pdf en faisant un glisser-déposer

Launch scripts to several nodes



bioinfo-master.ird.fr
91.203.34.148

Use the
qsub
command

Hostname :
bioinfo-master.ird.fr

Login : cluster
account

Password : cluster
password
Port : 22

Test your script(s)



bioinfo-inter.ird.fr
91.203.34.150

Hostname :
bioinfo-inter.ird.fr

Login : cluster
account

Password : cluster
password

Port : 22

Or use the qrun command on
bioinfo-master.ird.fr



With Putty
Use parameters above



© Can Stock Photo - csp4340788



With terminal
Use the ssh command

Se connecter à bioinfo-master.ird.fr via ssh

Taper :

\$~ssh login@bioinfo-master.ird.fr sur Apple ou Linux

Sous windows : télécharger Mobaxterm à l'URL :

<https://mobaxterm.mobatek.net/download-home-edition.html>

Puis se connecter à bioinfo-master.ird.fr

On peut réserver un nœud par lancer une analyse
pendant une durée limitée en utilisant la commande qsrh
Taper la commande qstat et analyser le résultat

On peut réserver un processeur sur un nœud pour lancer une analyse pendant une durée limitée en utilisant la commande qcrsh
Taper la commande qstat et analyser le résultat

Type :
\$~qcrsh
Vérifier sur quel noeud vous êtes avec la commande
\$~ uname -a
\$~ qstat

Se déplacer dans le répertoire d'accueil des données temporaires /scratch
Créer un répertoire pour y accueillir nos données

Se déplacer dans le répertoire d'accueil des données temporaires /scratch
Créer un répertoire pour y accueillir nos données

Taper les commandes :
\$~cd /scratch
\$~ mkdir login (avec le login le mot de répertoire de son choix)

En étant connecté à A

Répertoire distant à transférer : /data/projects/tp-cluster/training_2018

Login : login

répertoire de destination sur le noeud : /scratch/tando

Copier le répertoire distant depuis le serveur B vers le serveur local A



**Destination
ServerA**



**Source
ServerB**

En étant connecté à A

Répertoire distant à transférer : /data/projects/tp-cluster/training_2018

Login : tando

répertoire de destination sur le noeud : /scratch/tando

Copier le répertoire distant depuis le serveur B vers le serveur local A

scp -r login@source_server:/remote_path



**Destination
ServerA**



**Source
ServerB**

/data/projects/tp-cluster/training_2018

En étant connecté à A

Répertoire distant à transférer : /data/projects/tp-cluster/training_2018

Login : tando

répertoire de destination sur le noeud : /scratch/tando

Copier le répertoire distant depuis le serveur B vers le serveur local A

```
scp -r login@source_server:/remote_path local_folder
```

/scratch/tando



**Destination
ServerA**



**Source
ServerB**

/data/projects/tp-cluster/training_2018

En étant connecté à A

Répertoire distant à transférer : /data/projects/tp-cluster/training_2018

Login : tando

répertoire de destination sur le noeud : /scratch/tando

Copier le répertoire distant depuis le serveur B vers le serveur local A

```
scp -r login@source_server:remote_path local_folder
```

/scratch/tando



**Destination
ServerA**



**Source
ServerB**

/data/projects/tp-cluster/training_2018

```
scp -r login@serverB:/data/projects/tp-cluster/training_2018
```


Transfert de données avec scp

En étant connecté à A

Répertoire distant à transférer : /data/projects/tp-cluster/training_2018

Login : tando

répertoire de destination sur le noeud : /scratch/tando

Copier le répertoire distant depuis le serveur B vers le serveur local A

`scp -r login@source_server:/remote_path local_folder`

/scratch/tando



**Destination
ServerA**



**Source
ServerB**

/data/projects/tp-cluster/training_2018

`scp -r login@serverB:/data/projects/tp-cluster/training_2018 /scratch/tando`

Etape4 : copie des données dans le répertoire d'analyses

Copier le répertoire /data/projects/tp-cluster/training_2018/Blast dans /scratch/login

Etape4 : copie des données dans le répertoire d'analyses

Copier le répertoire /data/projects/tp-cluster/training_2018/Blast dans /scratch/login

Taper les commandes :

```
$~cd /scratch/login
```

```
$~ scp -r login@bioinfo-nas2.ird.fr :/data/projects/tp-cluster/training_2018/Blast  
/scratch/login
```

Aller dans le répertoire /scratch/login/Blast
Lister les fichiers du répertoire

Aller dans le répertoire /scratch/login/Blast
Lister les fichiers du répertoire

Taper :
\$~cd /scratch/login/Blast
\$~ls -ali

- Permet de choisir la version du logiciel que l'on veut utiliser
- 2 types de logiciels :
 - bioinfo : désigne les logiciels de bioinformatique (exemple BEAST)
 - system : désigne tous les logiciels systèmes(exemple JAVA)
- Surpassent les variables d'environnement
- 5 types de commandes :

Voir les modules disponibles : `module avail`

Obtenir une info sur un module en particulier : `module whatis + module name`

Par exemple `module whatis bioinfo/blast/2.4.0+`

Charger un module : `module load + modulename`

Par exemple `module load bioinfo/blast/2.4.0+`

Lister les modules chargés : `module list`

Décharger un module : `module unload + modulename`

Par exemple `module unload bioinfo/blast/2.4.0+`

Décharger tous les modules :

`Module purge`

Charger le module version 2.4.0+
Utiliser la commande blastn pour lancer une analyse blast
qui fournira le fichier de sortie appelé blastn.out

Charger le module version 2.4.0+
Utiliser la commande blastn pour lancer une analyse blast
qui fournira le fichier de sortie appelé blastn.out

Taper :
\$~ module load bioinfo/blast/2.4.0+
\$~ blastn -db All-EST-cofea.fasta -query sequence-NMT.fasta -out blastn.out

Editer le fichier blastn.out avec l'utilitaire nano

Editer le fichier blastn.out avec l'utilitaire nano

Taper :
`$~ nano blastn.out`

Copier le fichier blastn.out vers son répertoire home utilisateur
Vérifier que le fichier est bien copié

Copier le fichier blastn.out vers son répertoire home utilisateur
Vérifier que le fichier est bien copié

Taper :
`$~scp blastn.out login@bioinfo-nas.ird.fr:/home/login`
`$~ls -ali /home/login`

Se déplacer dans le répertoire
Supprimer le répertoire de travail

Taper:
\$~cd /scratch
\$~ rm -rf *login*



**TP: Lancer un bwa de
manière interactive**

- Suivre les étapes du TP précédent et les adapter à celui-ci
- Le répertoire à copier est: /data/projects/training_2018/bwa
 - La version de bwa à utiliser est la 0.7.12
 - Les commandes à lancer sont:
bwa index referencelrigin.fasta
bwa mem referencelrigin.fasta irigin1_1.fastq irigin1_2.fastq >mapping.sam
- Récupérer le fichier mapping.sam et le mettre dans son /home/login

Cf solution: [practice2](#)



TP: Lancer une analyse à l'aide d'un script

- C'est le fait d'exécuter un script bash via sge
- On utilise la commande:

```
$~ qsub script.sh
```

Avec `script.sh` : le nom du script

Dans la première partie du script on renseigne les options d'exécution de sge avec le mot clé # $\$$ (partie en vert)

```
#!/bin/sh

##### SGE CONFIGURATION #####
# Ecrit les erreurs dans le fichier de sortie standard
#$ -j y

# Shell que l'on veut utiliser
#$ -S /bin/bash

# Email pour suivre l'exécution
#$ -M prenom.nom@ird.fr ##### Mettre son adresse mail

# Type de message que l'on reçoit par mail
# - (b) un message au démarrage
# - (e) à la fin
# - (a) en cas d'abandon
#$ -m bea

# Queue que l'on veut utiliser
#$ -q bioinfo.q

# Nom du job
#$ -N Nom_a_choisir
#####
```

Dans la 2e partie du script on renseigne les actions à effectuer

```
path_to_dir="/data/projects/rep_a_choisir";
path_to_tmp="/scratch/nom_rep_a_choisir-$JOB_ID"

##### Creation du repertoire temporaire sur noeud et chargement du module blast
module load bioinfo/blastn/2.4.0+
mkdir $path_to_tmp
scp -rp nas2:$path_to_dir/* $path_to_tmp # choisir nas pour/home, /data2 et /teams ou nas2 pour /data ou nas3 pour /data3
echo "transfert donnees master -> noeud";
cd $path_to_tmp

##### Execution du programme
cmd="blastn -db All-EST-cofea.fasta -query sequence-NMT.fasta -num_threads $NSLOTS -out blastn1-$JOB_ID.out";
echo "Commande executee : $cmd";
$cmd;

##### Transfert des données du noeud vers master
scp -rp $path_to_tmp/ nas:$path_to_dir/
echo "Transfert donnees node -> master";

#### Suppression du repertoire tmp noeud
rm -rf $path_to_tmp
echo "Suppression des donnees sur le noeud";
```

- Reprendre le TP 1 et le mettre sous forme de script en s'aidant du script exemple précédent
- Lancer le script avec la commande `qsub`
- Observer le déroulement avec la commande `watch qstat`

solution script blastn

- Reprendre le TP 2 et le mettre sous forme de script en s'aidant du script exemple précédent
- Lancer le script avec la commande `qsub`
- Observer le déroulement avec la commande `watch qstat`

solution script bwa