



# Initiation Slurm i-Trop cluster

[www.southgreen.fr](http://www.southgreen.fr)

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





Emmanuelle Beyne  
20% ETP

Aurore COMTE  
20% ETP

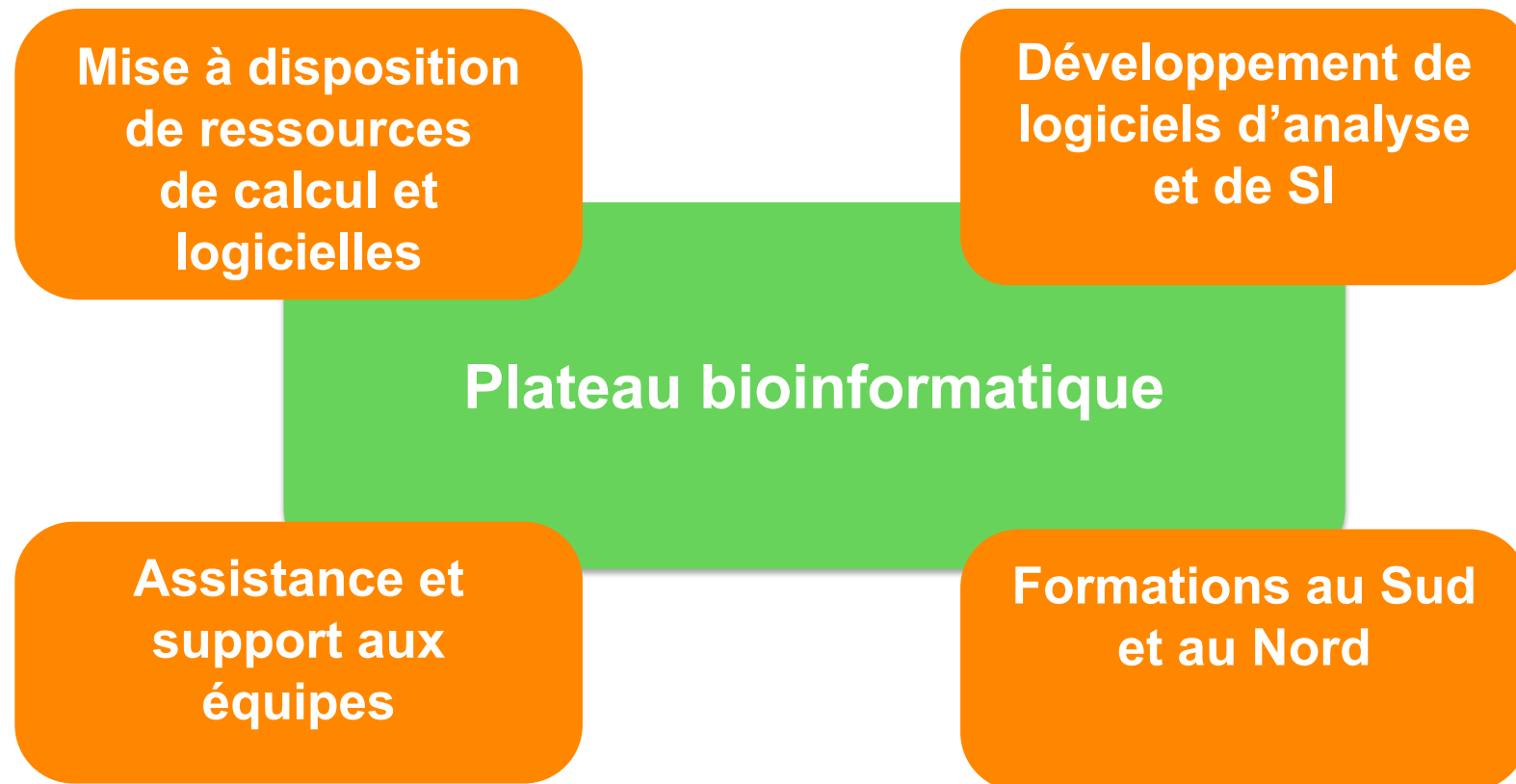
Bruno GRANOUILAC  
50% ETP

Valérie NOEL  
25% ETP

Julie ORJUELA-BOUNOL  
25% ETP

Ndomassi TANDO  
100% ETP

Christine TRANCHANT-DUBREUIL  
20% ETP



- Formulaires de demandes

<https://itrop-glpi.ird.fr/plugins/formcreator/front/formlist.php>

- Comptes
- Installation logiciels
- Projets

- Incidents: contacter [bioinfo@ird.fr](mailto:bioinfo@ird.fr)

- Howtos:

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

- Tutorials Slurm:

<https://southgreenplatform.github.io/tutorials//cluster-itrop/Slurm/>



# Etapes d'une analyse sur le cluster

Connexion à  
bioinfo-mas-  
ter.ird.fr et  
réservation  
de  
ressources

**Etape 1**

Création du  
répertoire  
d'analyse  
/scratch  
dans le  
noeud  
réservé

**Etape 2**

Transfert de  
données  
depuis  
Ordinateur  
personnel  
vers les  
serveurs  
nas

**Etape 3**

Transfert  
des  
données  
depuis les  
nas vers le  
/scratch du  
noeud

**Etape 4**

Charger ses  
logiciels avec  
modules  
environment

**Etape 5**

Lancer les  
analyses sur  
les données

**Etape 6**

Transfert  
des  
résultats sur  
les serveurs  
nas

**Etape 7**

Suppression  
Du répertoire  
d'analyse sur  
le /scratch

**Etape 8**



# Practice

Etape 1 Et 2: Connexion, sinfo

1

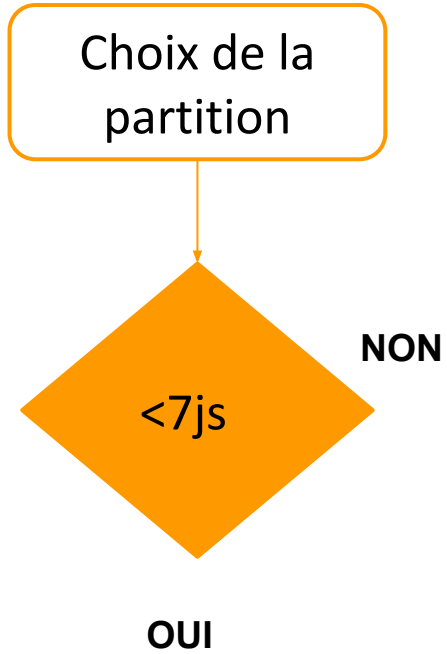
*Aller sur les [Practice 1 Et 2](#) du github*

# Les files d'attente

Partitions	Utilisation	Caractéristiques RAM noeuds	Caractéristiques coeurs noeuds
short	Jobs courts < 1 jour (priorité plus haute, jobs interactif)	48 à 64 Go	12 coeurs
normal	Jobs courts max 7 jours	64 Go à 96 Go	12 à 24 coeurs
r900	Jobs courts max 7 jours	32Go	16 coeurs /scratch 117Go
long	45 jours >Jobs longs > 7 jours	48 Go	12 à 24 coeurs
highmem	Jobs avec besoin de plus de mémoire	144 Go	12 à 24 coeurs
supermem	Jobs avec besoin de beaucoup de mémoire	1To	40 coeurs
gpu	Besoin d'analyses sur des gpus	192Go	24 cpus et 8 coeurs GPUs

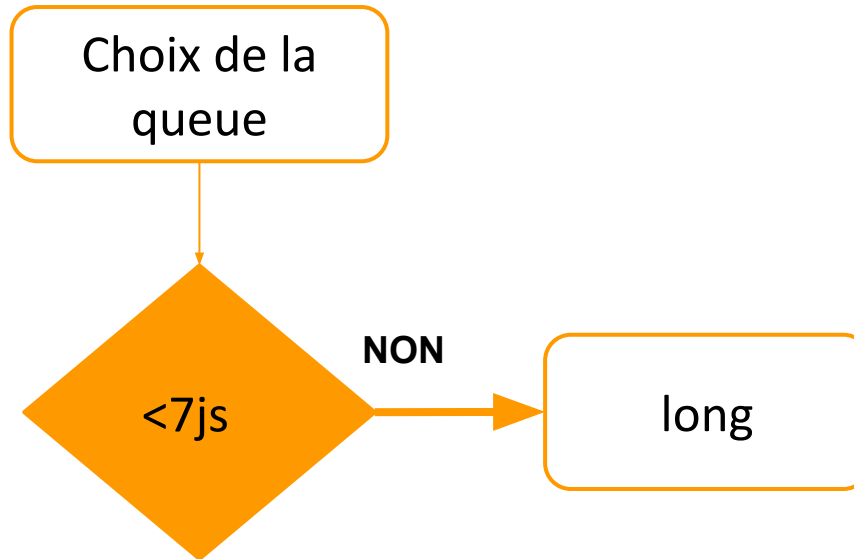
\*Demande à faire avec argumentaire

# Quelle partition choisir?

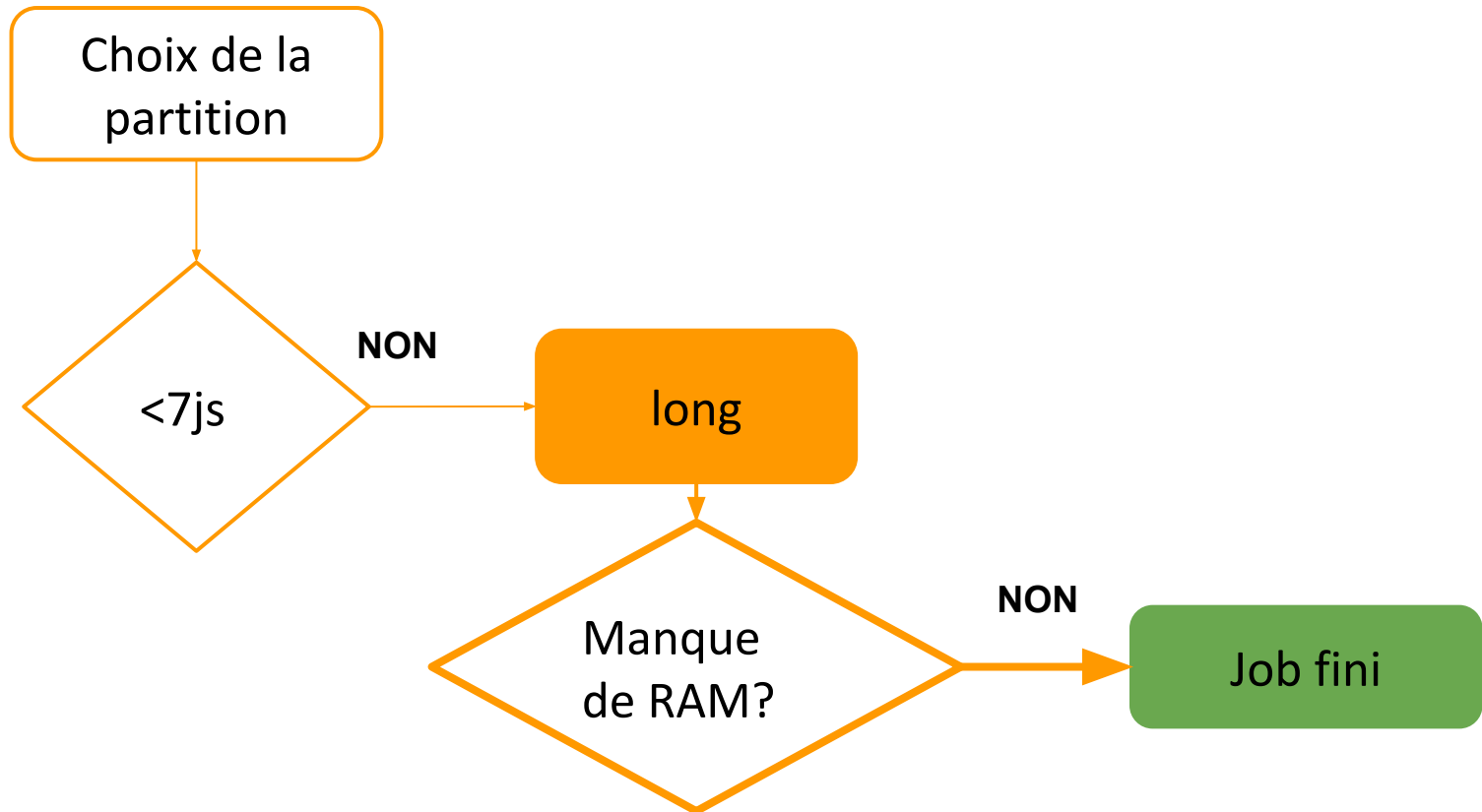




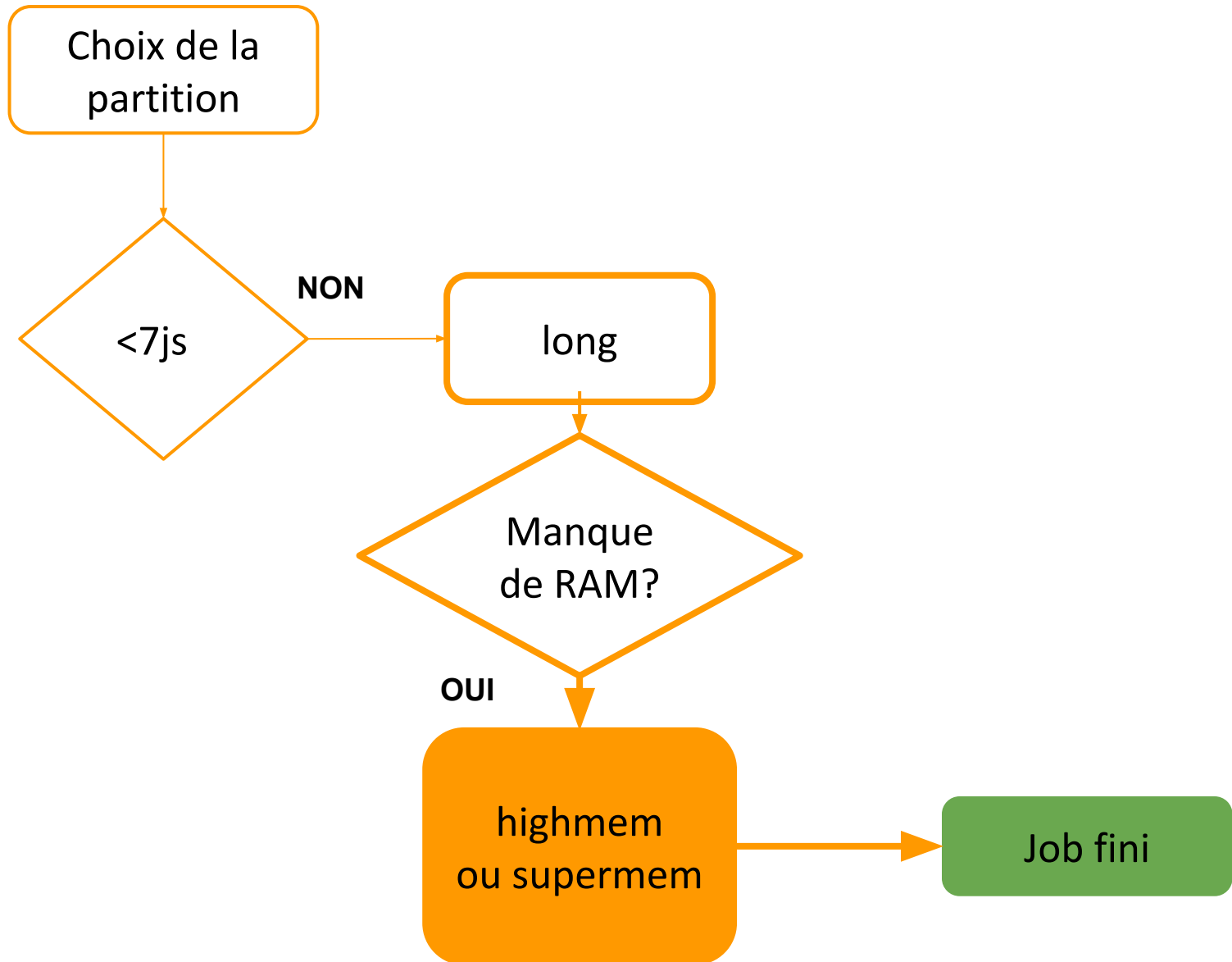
# Quelle partition choisir?



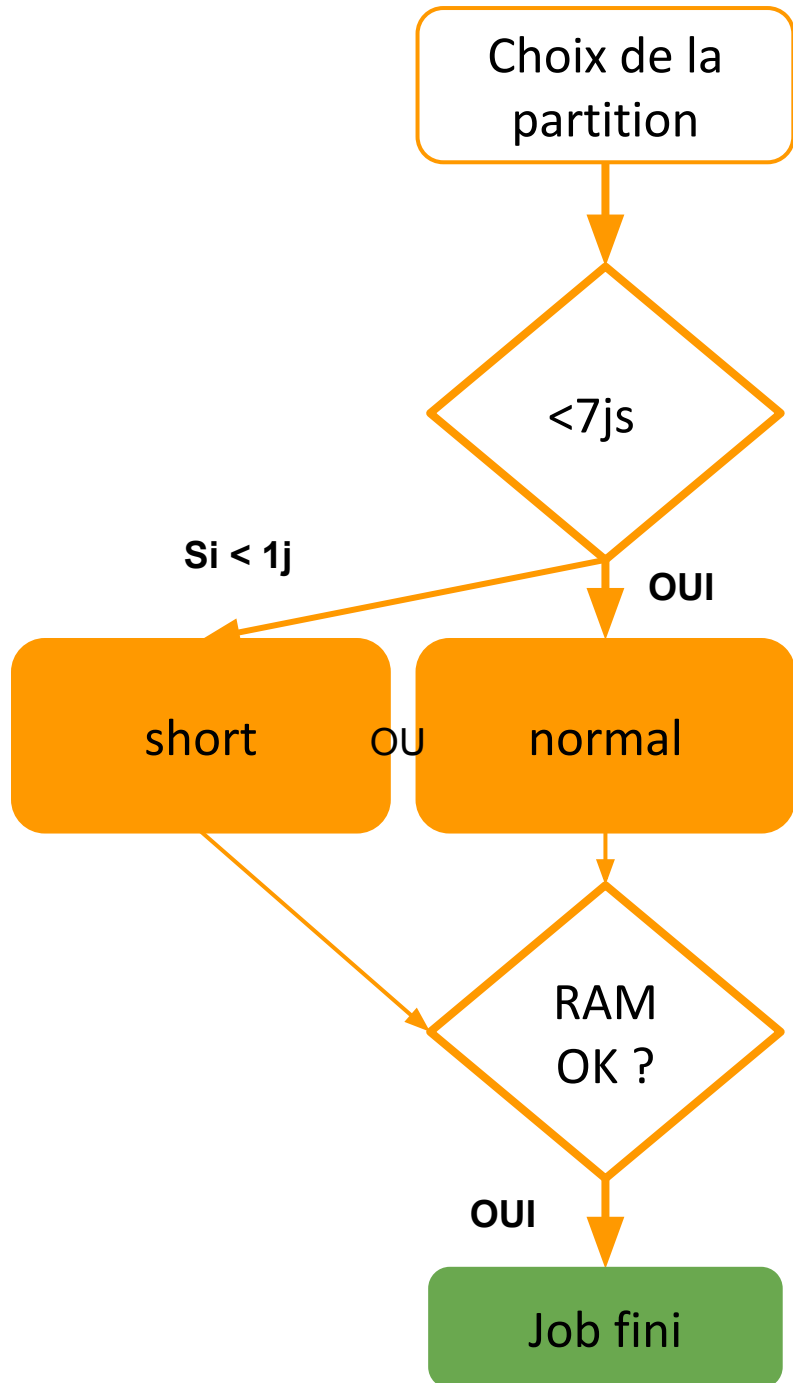
# Quelle partition choisir?



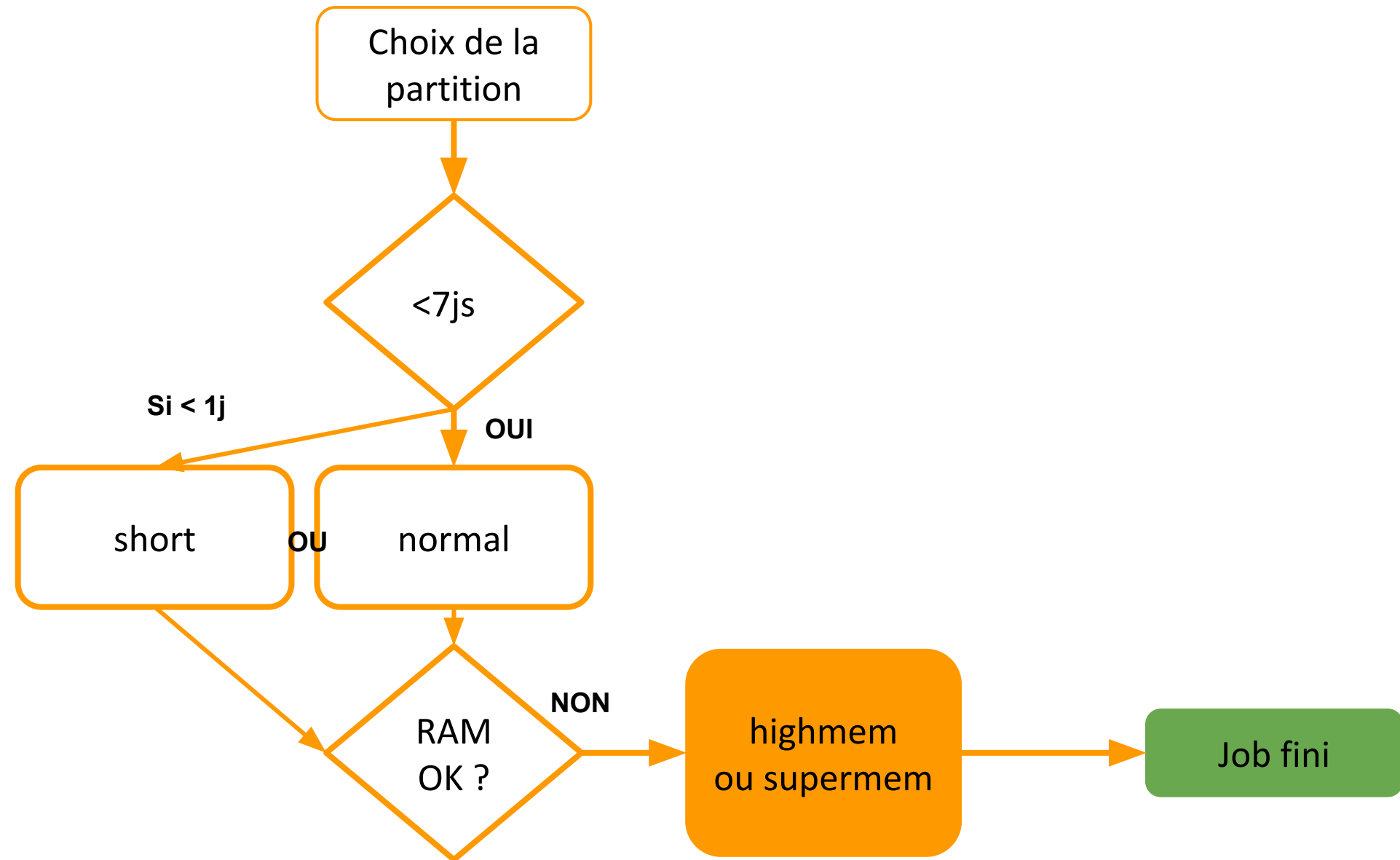
# Quelle partition choisir?



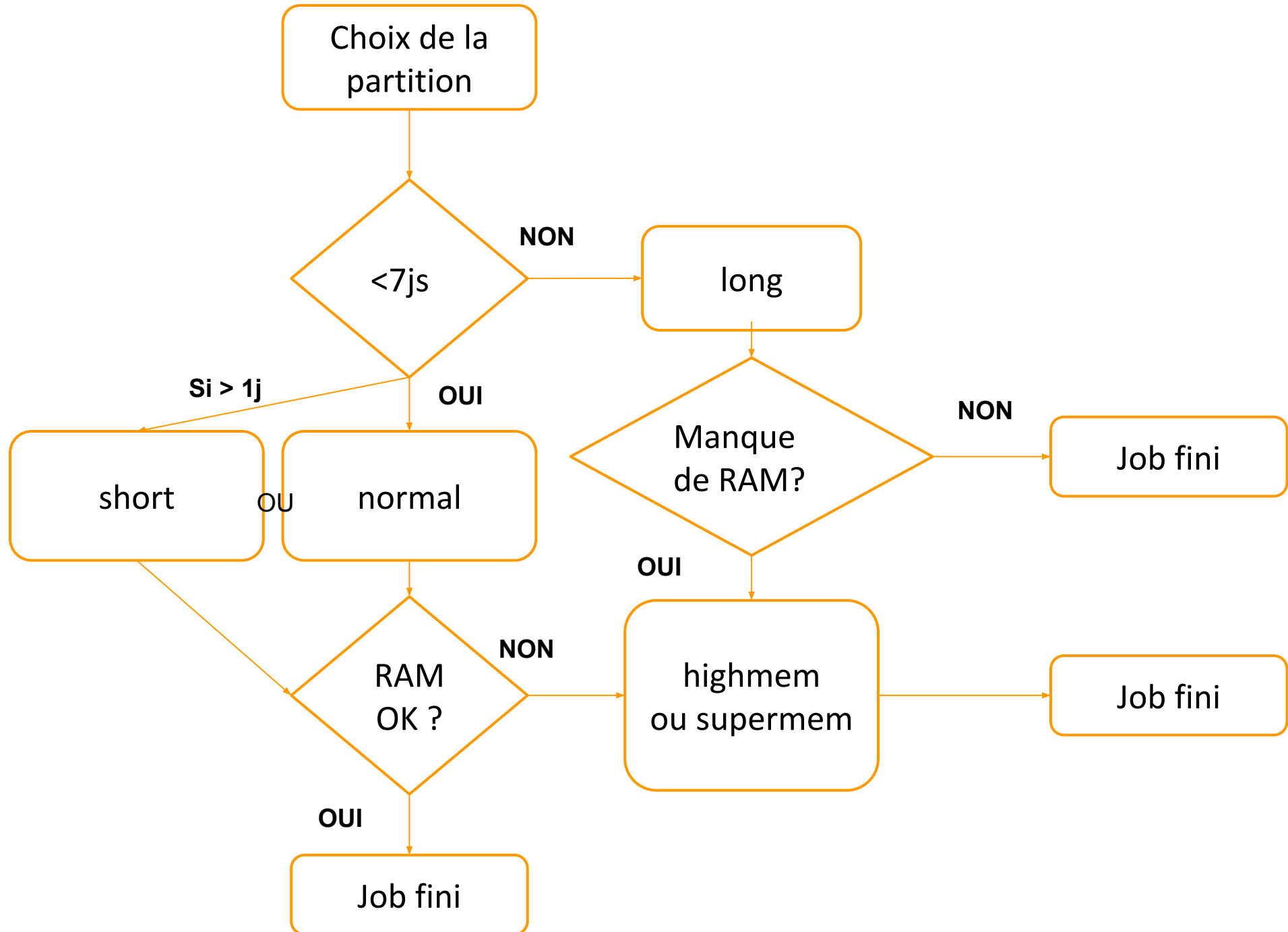
# Quelle partition choisir?



# Quelle partition choisir?



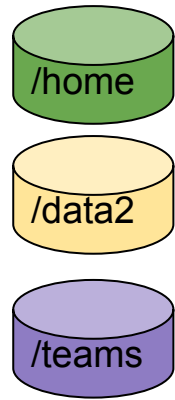
# Quelle partition choisir?



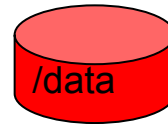
- Partition pour effectuer des travaux sur des processeurs GPUs: basecalling, MiniOn etc..
- Accès restreint au groupe gpu\_account
- Demande d'accès avec argumentaire à faire sur

<https://itrop-glpi.ird.fr/plugins/formcreator/front/formlist.php>

# Partitions disques sur le cluster i-Trop



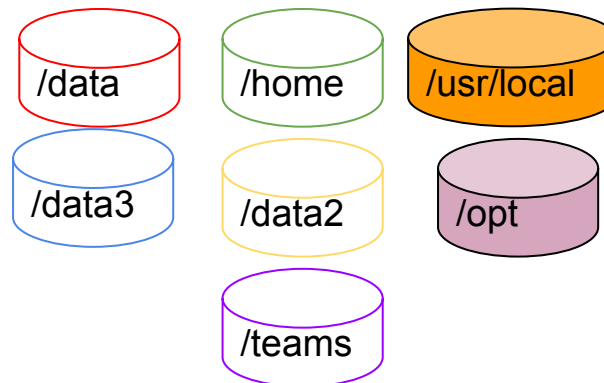
**bioinfo-nas.ird.fr**



**bioinfo-nas2.ird.fr**

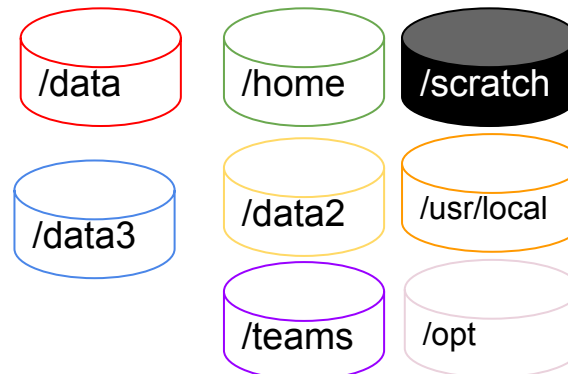


**bioinfo-nas3.ird.fr**



**bioinfo-master.ird.fr**

**Liens virtuels** vers les  
partitions des autres  
machines



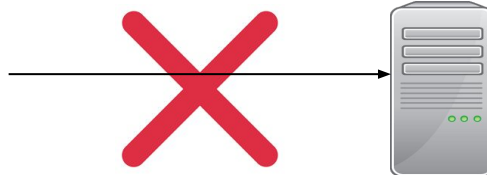
**27 noeuds**







Ordinateur  
personnel



**Transfert direct  
via filezilla  
interdit**



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



# Practice

Etape3 et 4: scp vers noeuds

4

*Aller sur le [Practice4](#) du github*

- 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`

- Charger un module :

`module load + modulename`

- Lister les modules chargés :

`module list`

- Décharger un module :

`module unload + modulename`

- Décharger tous les modules :

`module purge`



# Practice

Etape5: module environment

5

*Aller sur le [Practice5](#) du github*

- Charger la version du logiciel à lancer
- Lancer l'analyse des données

```
$~ commande <options> <arguments>
```

Avec *commande*: la commande à lancer

- Exécuter une commande bash via srun
- Lance la commande sur un noeud
- On utilise la commande:

```
$~ srun <options> <commande>
```

Avec *commande*: la commande à lancer



# Practice

Etape6: lancer l'analyse

6

*Aller sur le [Practice6](#) du github*



Commande	Description	Exemple
<code>srun --time=0X:00 --pty bash -i</code>	Se connecter de manière interactive à un noeud pendant X minutes	<code>srun --time=02:00:00 --pty bash -i</code> Connexion pendant 2 heures
<code>salloc --time=0X:00</code>	S'allouer un ou plusieurs noeuds pour une utilisation future	<code>salloc -N 2 --p short --time=05:00</code>
<code>sbatch</code>	Lancer une analyse via script en arrière plan	<code>sbatch script.sh</code>
<code>sinfo</code>	Informations sur les partitions	<code>sinfo</code>
<code>scancel</code>	Suppression d'un job <job_id>	<code>scancel 1029</code>
<code>squeue</code>	Infos sur tous les jobs	<code>squeue -u tando</code>
<code>scontrol show job &lt;job_id&gt;</code>	Infos sur le job actif <job_id>	<code>scontrol show job 1029</code>

Options	Description	Exemple
<code>--job-name=&lt;name&gt;</code>	Donner un nom au job	<code>sbatch --job-name=tando_blast</code>
<code>-p &lt;partition&gt;</code>	Choisir une partition	<code>sbatch -p highmem</code>
<code>--odelist=&lt;nodeX&gt;</code>	Choisir un noeud en particulier	<code>sbatch -p normal --odelist=node14</code>
<code>-n &lt;nbre_taches&gt;</code>	Lancer plusieurs instance d'une commande	<code>srun -n 4 hostname</code>
<code>-c &lt;nb_cpu_par_tache&gt;</code>	Allouer le nombre de cpus par tâche	<code>srun -n 4 -c 2 hostname</code>
<code>--mail-user=&lt;emailaddress&gt;</code>	Envoyer un mail	<code>sbatch --mail-user=ndomassi@ird.fr</code>
<code>--mail-type=&lt;event&gt;</code>	Envoyer un mail quand: END: fin du job FAIL: abandon BEGIN: début du job ALL: tout	<code>sbatch ---mail-type=BEGIN</code>
<code>--workdir=[dir_name]</code>	Préciser le répertoire de travail	<code>sbatch s--workdir=/scratch/tando script.sh</code>



# Practice

## Etape7: Récupérer les résultats

7

*Aller sur le [Practice7](#) du github*

- Scratch= espaces temporaires
- Vérifier la copie des résultats avant
- Utiliser la commande rm

```
cd /scratch  
rm -rf nom_rep
```



# Practice

Etape8: suppression des données

8

*Aller sur le [Practice8](#) du github*

# Scripts pour visualiser/supprimer données temporaires

- Emplacement des scripts: /opt/scripts/scratch-scripts
- Visualiser ses données sur les scratches: scratch\_use.sh

```
sh /opt/scripts/scratch-scripts/scratch_use.sh
```

- Supprimer ses données sur les scratches: clean\_scratch.sh

```
sh /opt/scripts/scratch-scripts/clean_scratch.sh
```

# LANCER UN JOB

- Le scheduler choisit les ressources automatiquement
- Lancer des jobs utilisant jusqu'à 24 coeurs
- Possibilité de paramétrer ce choix
- Jobs lancés en arrière plan
  - possibilité d'éteindre son ordinateur
  - récupération des résultats automatique



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

```
$~ sbatch script.sh
```

Avec `script.sh` : le nom du script

Options	Description	Exemple
<code>--job-name=&lt;name&gt;</code>	Donner un nom au job	<code>sbatch --job-name=tando_blast</code>
<code>-p &lt;partition&gt;</code>	Choisir une partition	<code>sbatch -p highmem</code>
<code>--odelist=&lt;nodeX&gt;</code>	Choisir un noeud en particulier	<code>sbatch -p normal --odelist=node14</code>
<code>-n &lt;nbre_taches&gt;</code>	Lancer plusieurs instance d'une commande	<code>srun -n 4 hostname</code>
<code>-c &lt;nb_cpu_par_tache&gt;</code>	Allouer le nombre de cpus par tâche	<code>srun -n 4 -c 2 hostname</code>
<code>--mail-user=&lt;emailaddress&gt;</code>	Envoyer un mail	<code>sbatch --mail-user=ndomassi@ird.fr</code>
<code>--mail-type=&lt;event&gt;</code>	Envoyer un mail quand: END: fin du job FAIL: abandon BEGIN: début du job ALL: tout	<code>sbatch ---mail-type=BEGIN</code>
<code>--workdir=[dir_name]</code>	Préciser le répertoire de travail	<code>sbatch s--workdir=/scratch/tando script.sh</code>

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

```
#!/bin/bash

##### Configuration SLURM#####
## On définit le nom du job
#SBATCH --job-name=test
## On définit le nom du fichier de sortie
#SBATCH --output=res.txt
## On définit le nombre de tâches
#SBATCH --ntasks=1
## On définit le temps limite d'exécution
#SBATCH --time=10:00
#####
```

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

```
#####Partie exécution des commandes #####
```

```
nom_variable1="valeur_variable1"  
nom_variable2="valeur_variable2"
```

```
sleep 30  
hostname
```



# Practice

Lancer un script avec qsub

9

*Aller sur le [Practice9](#) du github*

Merci de compléter l'enquête à cette adresse:

<https://itrop-survey.ird.fr/index.php/562934?lang=fr>

Si vous utilisez les ressources du plateau i-Trop.

Merci de nous citer avec:

“The authors acknowledge the IRD itrop HPC (South Green Platform) at IRD montpellier  
for providing HPC resources that have contributed to the  
research results reported within this paper.

URL: <https://bioinfo.ird.fr/>- <http://www.southgreen.fr>”

- Pensez à inclure un budget ressources de calcul dans vos réponses à projets
- Besoin en disques dur, renouvellement de machines etc...
- Devis disponibles
- Contactez [bioinfo@ird.fr](mailto:bioinfo@ird.fr) : aide, définition de besoins, devis...



# Merci pour votre attention !



Le matériel pédagogique utilisé pour ces enseignements est mis à disposition selon les termes de la licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions (BY-NC-SA) 4.0 International:

<http://creativecommons.org/licenses/by-nc-sa/4.0/>