Workflow & Conclusion

Pierre Pericard, Rachel Legendre, Emilie Drouineau, Thibault Dayris,
Claire Toffano-Nioche, Charlotte Berthelier, Audrey Onfroy

École de Bioinformatique IFB-INSERM 2024

Reprise du workflow : définition

Workflow = enchaînement d'étapes individuelles

- Écriture sous forme de **scripts** en bash

  → Commence par un **"she-bang" (#!)** qui indique l'interpréteur du script (**#!/bin/bash**)

  → Les lignes commençant par un **"#"** sont des commentaires et ne sont pas interprétées

  → Créer des **variables** pour généraliser votre script (pas spécifique à un échantillon)

Reprise du workflow : définition

Workflow = enchaînement d'étapes individuelles

- Écriture sous forme de **scripts** en bash

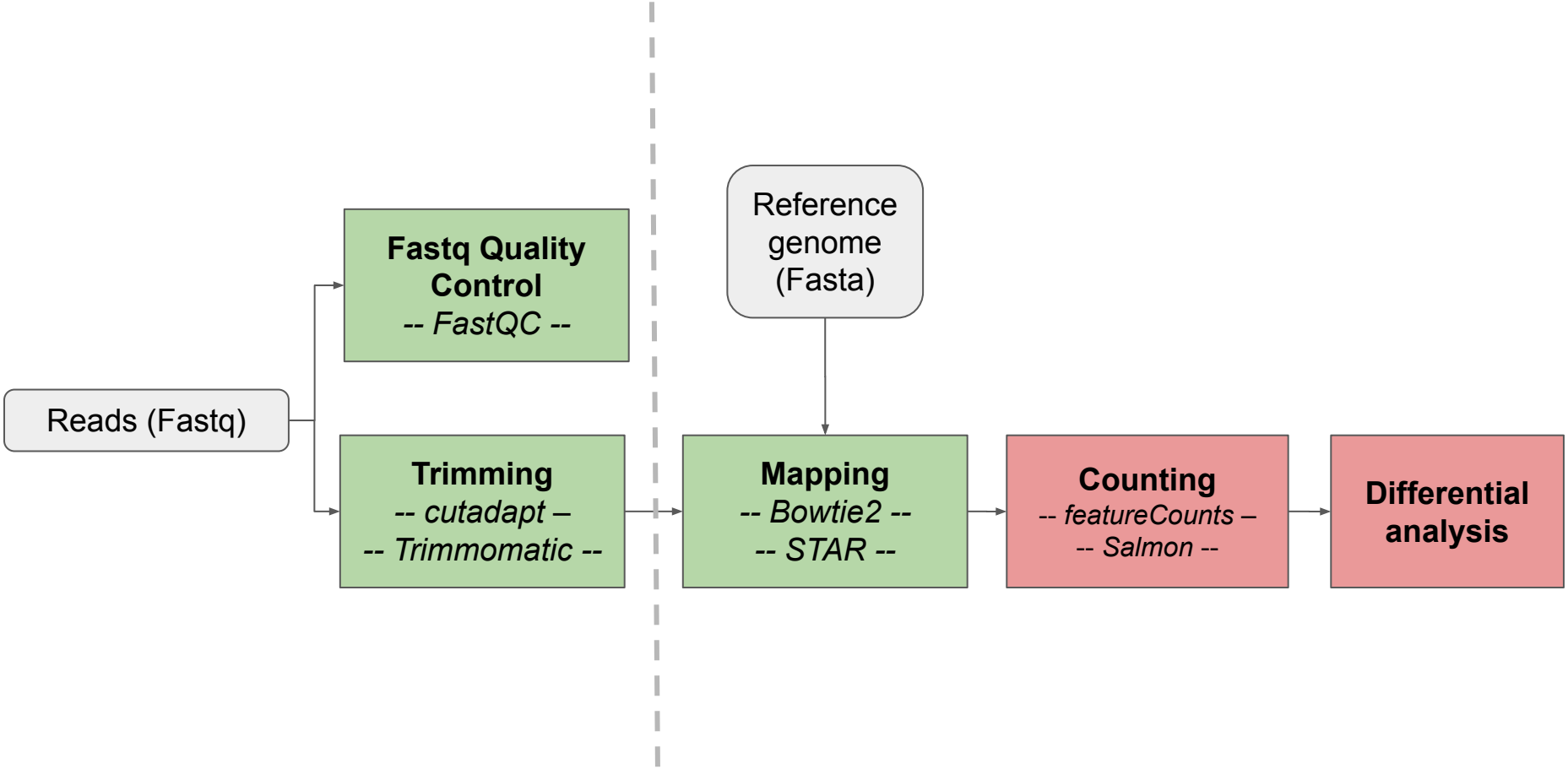
  → Commence par un **"she-bang" (#!)** qui indique l'interpréteur du script (**#!/bin/bash**)

  → Les lignes commençant par un "**#**" sont des commentaires et ne sont pas interprétées

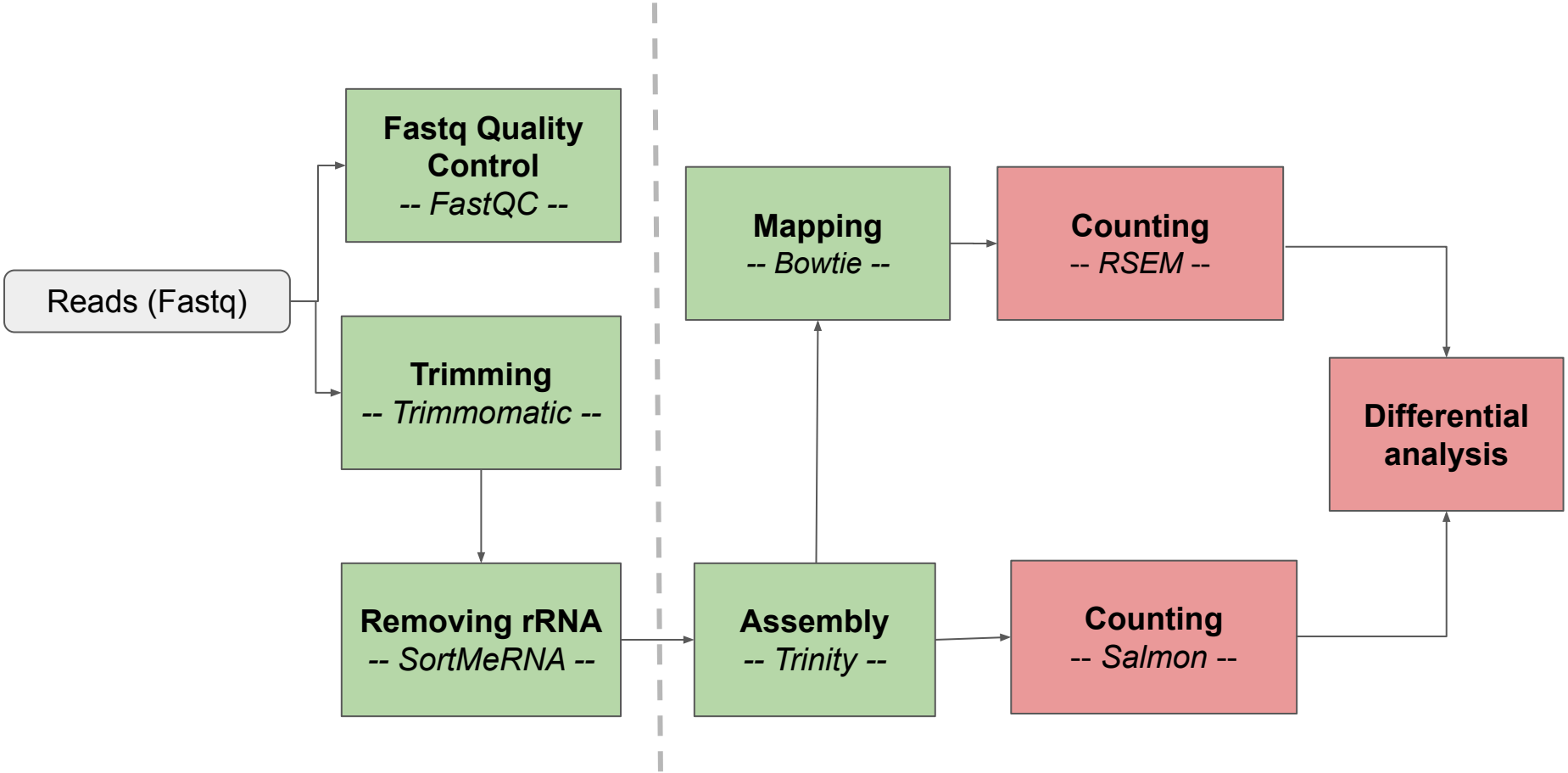
  → Créer des **variables** pour généraliser votre script (pas spécifique à un échantillon)

- Utilisation d'un gestionnaire de workflow
  (Galaxy, Nextflow, Snakemake, CWL, …)

Workflow - avec un génome de référence



Workflow - sans génome de référence



Exercice

Objectif : lancer le même outil (FastQC) sur 6 échantillons différents

2422_ebaii_n1

2 CPUs, 2 Go RAM

Exercice

Objectif : lancer le même outil (FastQC) sur 6 échantillons différents

Nécessite :

- Écriture d'un script bash
- Déclaration de **variables** pour généraliser les échantillons et les répertoires de travail
- Réalisation d'une **boucle** pour lancer l'outil sur chaque échantillon

```
$ ls -l /shared/projects/2422_ebaii_n1/atelier_rnaseq/04-Workflow/data

$ cd <path/to/your/project>
$ mkdir -p TP_rnaseq/workflow
$ cd TP_rnaseq/workflow

$ cp /shared/projects/2422_ebaii_n1/atelier_rnaseq/04-Workflow/TP/fastqc.sh .
```

Script1: écriture des lignes de commandes

```
#!/bin/bash

module load fastqc/0.12.1

mkdir -p fastqc_res




fastqc --outdir fastqc_res
/shared/projects/2422_ebaii_n1/atelier_rnaseq/04-Workflow/data/KO1_R1.fastq.gz

fastqc --outdir fastqc_res
/shared/projects/2422_ebaii_n1/atelier_rnaseq/04-Workflow/data/KO1_R2.fastq.gz
```

Lancement du script

Sauvegarder votre script et lancez le dans le terminal

```
$ bash fastqc.sh
```

Utilisation de variables

“Les variables sont des éléments qui associent un nom (l'identifiant) à une valeur, qui sera implantée dans la mémoire du système programmé. Une variable contient une valeur qui peut varier au cours de l'exécution du programme.”

```
$ echo $HOME
/shared/home/jdoe

$ PROJECT=/shared/projects/<your_project>/

$ echo $PROJECT
/shared/projects/<your_project>/
```

Utilisation de variables

Une variable permet d'anonymiser un script.

```
$ PRENOM="Toto"
```

'PRENOM' est le nom de la variable, "Toto" est sa valeur
On peut ensuite utiliser une variable dans une ligne de commande

```
# la commande echo, affiche les arguments qui lui sont donnés
$ echo ${PRENOM}
```

Créez :

- la variable nommée **DATA_DIR** qui prendra comme valeur le chemin du dossier qui contient les fichiers fastq
- 2 variables, nommées **R1** et **R2**, qui correspondront aux filenames des 2 fichiers fastq R1 et R2.

Script2: anonymisation avec des variables

Rappel du Script1

```
#!/bin/bash

module load fastqc/0.12.1

mkdir -p fastqc_res




fastqc --outdir fastqc_res
/shared/projects/2422_ebaii_n1/atelier_rnaseq/04-Workflow/data/KO1_R1.fastq.gz

fastqc --outdir fastqc_res
/shared/projects/2422_ebaii_n1/atelier_rnaseq/04-Workflow/data/KO1_R2.fastq.gz
```

Script2: anonymisation avec des variables

```
#!/bin/bash

module load fastqc/0.12.1

mkdir -p fastqc_res

DATA_DIR="/shared/projects/2422_ebaii_n1/atelier_rnaseq/04-Workflow/data/"
R1="${DATA_DIR}/KO1_R1.fastq.gz"
R2="${DATA_DIR}/KO1_R2.fastq.gz"

fastqc --outdir fastqc_res ${R1}


fastqc --outdir fastqc_res ${R2}
```

Lancement du script

Sauvegardez votre script (**Ctrl + s**) et lancez à nouveau le script dans le terminal

```
$ bash fastqc.sh
```

On vérifie que ça marche et on peut arrêter avec **Ctrl + C**

Utilisation d'une boucle

Une boucle permet d'itérer sur une liste de valeurs pour une variable

```
$ for PRENOM in Audrey Claire Charlotte Pauline Sarah Mathieu Jean-Pascal Lucie
Elise Erwan Gildas Julien
  do
    echo ${PRENOM}
     echo "======="
  done
```

A partir de la liste des fichiers R1 et R2 du dossier DATA_DIR, créez 2 boucles (une pour les fichiers R1 et une pour les fichiers R2) pour lancer la ligne de commande fastqc sur tous les fichiers du dossier fastq.

Indice: pensez à l'utilisation du *wildcard* (*) comme dans *.txt (qui liste tous les fichiers qui se terminent par ".txt")

Script3: automatisation sur plusieurs valeurs

Rappel du Script2

```
#!/bin/bash

module load fastqc/0.12.1

mkdir -p fastqc_res

DATA_DIR="/shared/projects/2422_ebaii_n1/atelier_rnaseq/04-Workflow/data/"
R1="${DATA_DIR}/KO1_R1.fastq.gz"
R2="${DATA_DIR}/KO1_R2.fastq.gz"

fastqc --outdir fastqc_res ${R1}


fastqc --outdir fastqc_res ${R2}
```

Script3: automatisation sur plusieurs valeurs

```
#!/bin/bash

module load fastqc/0.12.1

mkdir -p fastqc_res

DATA_DIR="/shared/projects/2422_ebaii_n1/atelier_rnaseq/04-Workflow/data/"


for R1 in $DATA_DIR/*_R1.fastq.gz
do
  fastqc --outdir fastqc_res ${R1}
done

for R2 in $DATA_DIR/*_R2.fastq.gz
do
  fastqc --outdir fastqc_res ${R2}
done
```

Script3: automatisation sur plusieurs valeurs

```
#!/bin/bash

module load fastqc/0.12.1

mkdir -p fastqc_res

DATA_DIR="/shared/projects/2422_ebaii_n1/atelier_rnaseq/04-Workflow/data/"


for R1 in $DATA_DIR/*_R1.fastq.gz
do
  fastqc --outdir fastqc_res ${R1}
done

for R2 in $DATA_DIR/*_R2.fastq.gz   # Est-ce qu'on ne peut pas encore simplifier ?
do
  fastqc --outdir fastqc_res ${R2}
done
```

Script3: automatisation sur plusieurs valeurs

```
#!/bin/bash

module load fastqc/0.12.1

mkdir -p fastqc_res

DATA_DIR="/shared/projects/2422_ebaii_n1/atelier_rnaseq/04-Workflow/data/"


for FQFILE in $DATA_DIR/*.fastq.gz  # On simplifie encore
do
  fastqc --outdir fastqc_res ${FQFILE}
done
```

Lancement du script

On lance à nouveau le script dans le terminal

```
$ bash fastqc.sh
```

Lancement du script

On lance à nouveau le script dans le terminal

```
$ bash fastqc.sh
```

C’est long non ???

On va vraiment prendre un café à chaque fois qu’on lance un FastQC ??? ☕

Bonus

Ajout des options pour SLURM

On commence à décrire un peu plus précisément les ressources nécessaires à votre étape

```
#!/bin/bash

#SBATCH --account=your_project
#SBATCH --job-name=your_job
#SBATCH --cpus-per-task=1 # Modifier en fonction des besoins
#SBATCH --mem=2GB         # Idem

module load ...

...
```

Plein d'autres options utiles:

voir intro SLURM et https://ifb-elixirfr.gitlab.io/cluster/doc/quick-start/

Script4: ajout des options SLURM

Rappel du Script3

```
#!/bin/bash

module load fastqc/0.12.1

mkdir -p fastqc_res

DATA_DIR="/shared/projects/2422_ebaii_n1/atelier_rnaseq/04-Workflow/data/"


for FQFILE in $DATA_DIR/*.fastq.gz
do
  fastqc --outdir fastqc_res ${FQFILE}
done
```

Script4: ajout des options SLURM

```
#!/bin/bash

#SBATCH --account=2422_ebaii_n1
#SBATCH --job-name=fastqc
#SBATCH --cpus-per-task=1
#SBATCH --mem=2GB

module load fastqc/0.12.1

mkdir -p fastqc_res

DATA_DIR="/shared/projects/2422_ebaii_n1/atelier_rnaseq/04-Workflow/data/"

for FQFILE in $DATA_DIR/*.fastq.gz
do
  fastqc --outdir fastqc_res ${FQFILE}
done
```

Script4: ajout des options SLURM

```
#!/bin/bash

#SBATCH --account=2422_ebaii_n1
#SBATCH --job-name=fastqc
#SBATCH --cpus-per-task=1
#SBATCH --mem=2GB

module load fastqc/0.12.1

mkdir -p fastqc_res

DATA_DIR="/shared/projects/2422_ebaii_n1/atelier_rnaseq/04-Workflow/data/"

for FQFILE in $DATA_DIR/*.fastq.gz
do
  srun fastqc --outdir fastqc_res ${FQFILE} # Optionnel mais facilite le suivi
done
```

Lancement du script

Cette fois ci, on lance le script avec la commande **sbatch**

```
$ sbatch fastqc.sh
```

Et on vérifie si ça se passe bien avec la commande squeue

```
$ squeue
...
$ squeue -u $USER
$ sacct -u $USER
```

Lancement du script

Cette fois ci, on lance le script avec la commande **sbatch**

```
$ sbatch fastqc.sh
```

Et on vérifie si ça se passe bien avec la commande squeue

```
$ squeue
...
$ squeue -u $USER
$ sacct -u $USER
```

C’est toujours aussi long…

On va vraiment prendre un café à chaque fois qu’on lance un FastQC ??? ☕

Parallélisation des tâches

Au lieu de lancer chaque job l'un après l'autre, de manière séquentielle, on va les lancer en parallèle

Comment ça marche ?

```
#!/bin/bash

#SBATCH --account=your_project
#SBATCH --job-name=your_job
#SBATCH --cpus-per-task=1
#SBATCH --mem=2GB
#SBATCH --array=1-4 # Modifier en fonction du nb de taches a lancer en parallele

module load ...

# Je selectionne le Nieme fichier de ma liste
INPUT=$(ls *.txt | awk "NR==${SLURM_ARRAY_TASK_ID}")

srun super_logiciel --input $INPUT
```

Script5: parallélisation des tâches

Rappel du Script4

```
#!/bin/bash

#SBATCH --account=2422_ebaii_n1
#SBATCH --job-name=fastqc
#SBATCH --cpus-per-task=1
#SBATCH --mem=2GB

module load fastqc/0.12.1

mkdir -p fastqc_res

DATA_DIR="/shared/projects/2422_ebaii_n1/atelier_rnaseq/04-Workflow/data/"

for FQFILE in $DATA_DIR/*.fastq.gz
do
  srun fastqc --outdir fastqc_res ${FQFILE}
done
```

Script5: parallélisation des tâches

```
#!/bin/bash

#SBATCH --account=2422_ebaii_n1
#SBATCH --job-name=fastqc
#SBATCH --cpus-per-task=1
#SBATCH --mem=2GB
#SBATCH --array=1-12

module load fastqc/0.12.1

mkdir -p fastqc_res

DATA_DIR="/shared/projects/2422_ebaii_n1/atelier_rnaseq/04-Workflow/data/"

FQFILE=$(ls $DATA_DIR/*.fastq.gz | awk "NR==${SLURM_ARRAY_TASK_ID}")

srun fastqc --outdir fastqc_res ${FQFILE}
```

Lancement du script SLURM

On lance le script avec la commande **sbatch**

```
$ sbatch fastqc.sh
```

Et on vérifie si ça se passe bien avec les commandes **squeue** ou **sacct**

```
$ squeue -u $USER
$ sacct -u $USER
```

Maintenant ça va super vite !
Vous êtes prêt.e.s à spammer le cluster de l'IFB ;-)

Mise en pratique: Alignement avec STAR

On vous propose maintenant de mettre en pratique ce que vous venez de voir et d'écrire un script SLURM pour aligner tous les échantillons sur le génome de référence avec STAR

```
$ module load star/2.7.9a
$ STAR --help
```

https://github.com/alexdobin/STAR/blob/master/doc/STARmanual.pdf

Index STAR: `/shared/bank/arabidopsis_thaliana/TAIR10.1/star-2.7.9a/`
Annotation (GTF):
`/shared/bank/arabidopsis_thaliana/TAIR10.1/gtf/GCF_000001735.4_TAIR10.1_genomic.gtf`

Mise en pratique: Alignement avec STAR

Petite commande bash utile: la substitution dans les variables

https://wiki.bash-hackers.org/syntax/pe#search_and_replace

```
$ FILE_A="~/tp_workflow/toto_a.txt"

$ echo ${FILE_A}
~/tp_workflow/toto_a.txt

$ FILE_B=${FILE_A/_a/_b}

$ echo ${FILE_B}
~/tp_workflow/toto_b.txt
```

Script6: Alignement avec STAR

On ne triche pas, la solution est 3 slides plus loin ;-)

Lancement du script

On voudra lancer le script avec la commande **sbatch**

```
$ sbatch star_align.sh
```

Attention: l'alignement avec STAR est gourmand en ressource.
Il faudra peut-être envisager de donner plus de CPUs et de RAM à votre outil…

Et on n'oublie pas de vérifier si ça se passe toujours bien

```
$ squeue -u $USER
$ sacct -u $USER
```

Script6: Alignement avec STAR

Ok, c’est bon. La solution est juste après

Script6: Alignement avec STAR

```
#!/bin/bash

#SBATCH --account=2422_ebaii_n1
#SBATCH --job-name=star_align
#SBATCH --cpus-per-task=8
#SBATCH --mem=20GB
#SBATCH --array=1-6

module load star/2.7.11a

OUT_DIR="/path/to/my/project/TP_rnaseq/workflow/star_res"
mkdir -p $OUT_DIR

DATA_DIR="/shared/projects/2422_ebaii_n1/atelier_rnaseq/04-Workflow/data/"
STAR_INDEX="/shared/bank/arabidopsis_thaliana/TAIR10.1/star-2.7.9a/"
GTF="/shared/bank/arabidopsis_thaliana/TAIR10.1/gtf/GCF_000001735.4_TAIR10.1_genomic.gtf"

R1IN=$(ls $DATA_DIR/*_R1.fastq.gz | awk "NR==${SLURM_ARRAY_TASK_ID}")
R2IN=${R1IN/_R1/_R2}
BASENAME=${R1IN/_R1.fastq.gz/.STAR_TAIR10.1_}

srun STAR --runThreadN ${SLURM_CPUS_PER_TASK} --genomeDir ${STAR_INDEX} --sjdbGTFfile ${GTF} --readFilesCommand zcat
--readFilesIn ${R1IN} ${R2IN} --outFileNamePrefix ${OUT_DIR}/${BASENAME} --outSAMtype BAM SortedByCoordinate
--outSAMunmapped Within KeepPairs
```