Workflow & Conclusion

Rachel Legendre, Emilie Drouineau, Thibault Dayris, Claire Toffano-Nioche

École de bioinformatique AVIESAN-IFB-INSERM 2021

Reprise du workflow : définition

[Vidéo] : **The 5 minutes IFB Core Cluster tutorial**

[Cheatsheet]
https://ifb-elixirfr.github.io/EBAII/2021/ebaiin1

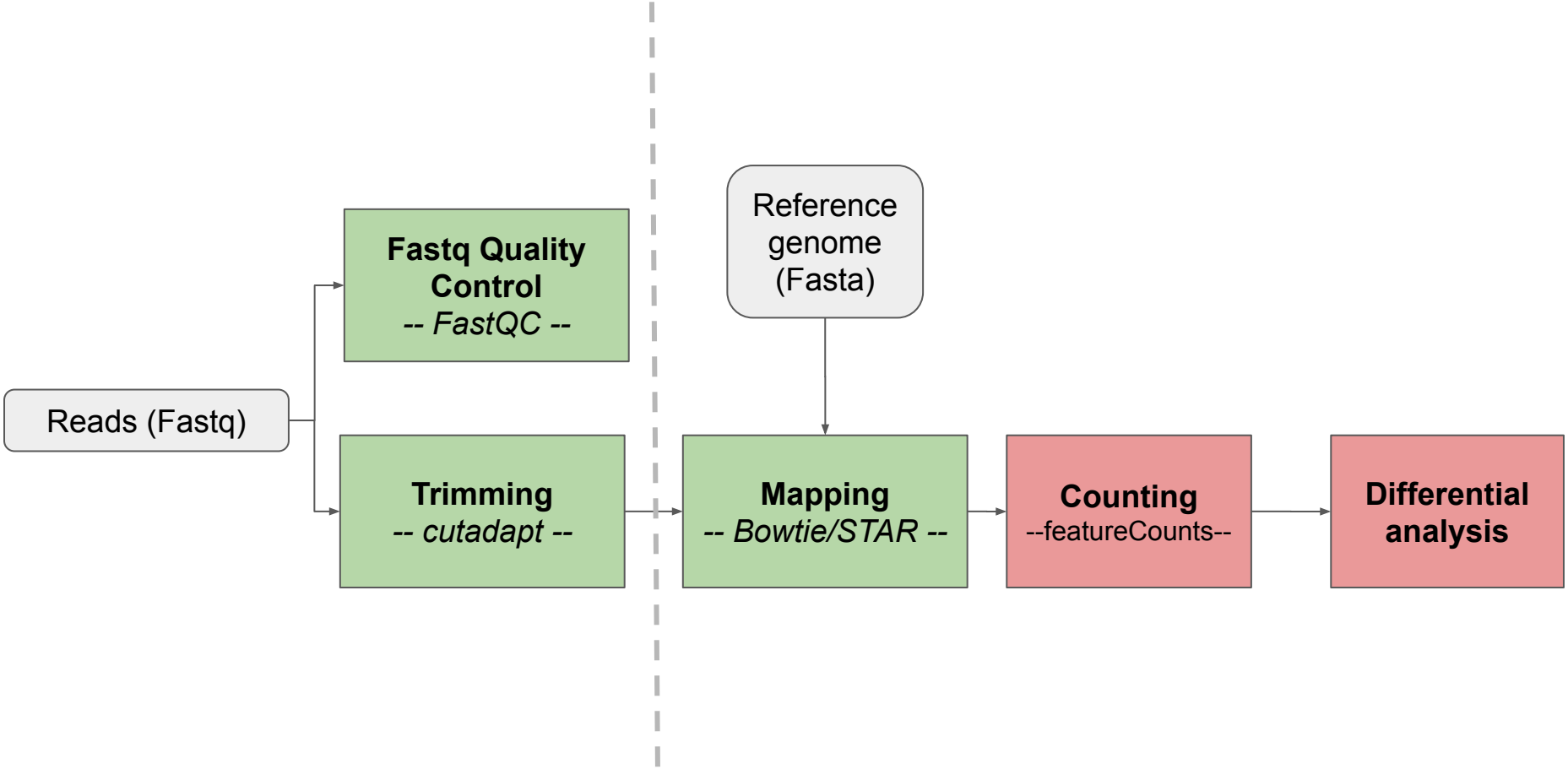
- **Workflow** : enchaînement d'étapes individuelles
- Ecriture sous forme d'un **script** en bash

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

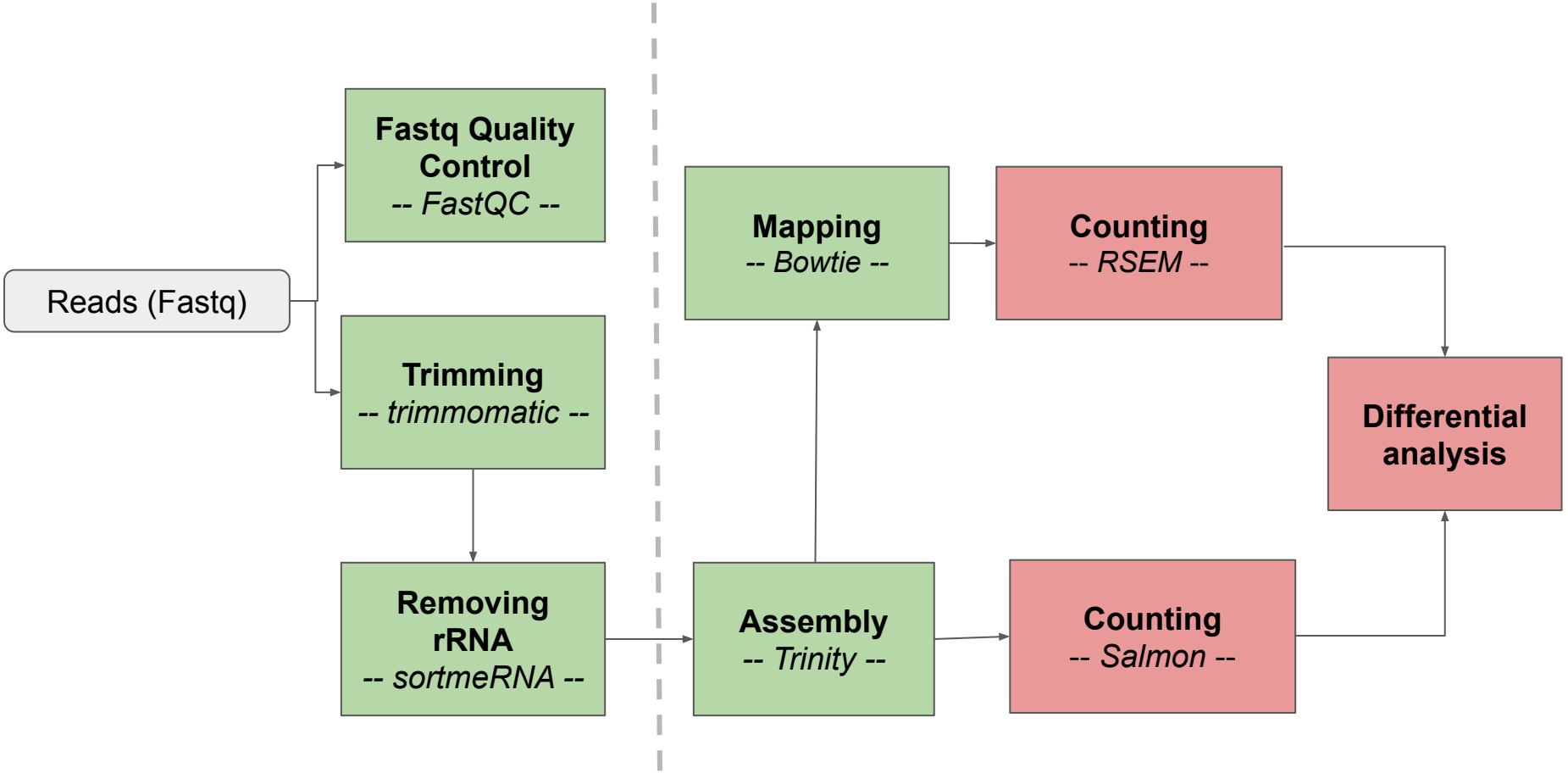
→ Les lignes commençant par un "**#**" sont des commentaires et ne sont pas interprétées

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

Workflow - avec un génome de référence



Workflow - sans génome de référence



Reprise du workflow : bilan des étapes

 # Partie preprocessing des données brutes

- Contrôle qualité avec l'outil fastqc
- [optionnel] Trimming/filtre qualité avec l'outil cutadapt
- [optionnel] Élimination des reads ribosomaux sortmeRNA

 # Partie alignement et comptage

- Alignement des séquences sur le génome de référence avec les outils STAR et samtools
- Comptage des lectures sur les éléments d'annotation featureCounts

Exercice

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

Nécessite :

- Ecriture 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

```
$ mkdir -p $HOME/tp_rnaseq/workflow
$ cd $HOME/tp_rnaseq/workflow

$ ls /shared/projects/form_2021_26/data/atelier_rnaseq/01-Bioinfo/data/

$ touch fastqc.sh
```

Script1: écriture des lignes de commandes

```
#!/bin/bash

mkdir -p fastqc_res


module load fastqc/0.11.9


fastqc --outdir fastqc_res
/shared/projects/form_2021_26/atelier_rnaseq/01-Bioinfo/data/KO1_R1.fastq.gz


fastqc --outdir fastqc_res
/shared/projects/form_2021_26/atelier_rnaseq/01-Bioinfo/data/KO1_R2.fastq.gz
```

Utilisation de variable

Une variable permet d'anonymiser un script.

```
$ PRENOM="Rachel"
```

'PRENOM' est le nom de la variable, 'Rachel' 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 DATA_DIR qui prendra comme valeur le nom du dossier qui contient les fichiers fastq
- 2 variables, R1 et R2, qui correspondront aux noms des 2 fichiers fastq R1 et R2.

Script2: anonymisation avec des variables

```
#!/bin/bash

mkdir -p fastqc_res


module load fastqc/0.11.9


fastqc --outdir fastqc_res
/shared/projects/form_2021_26/data/atelier_rnaseq/01-Bioinfo/data/KO1_R1.fastq.gz


fastqc --outdir fastqc_res
/shared/projects/form_2021_26/data/atelier_rnaseq/01-Bioinfo/data/KO1_R2.fastq.gz
```

Script2: anonymisation avec des variables

```
#!/bin/bash

mkdir -p fastqc_res

DATA_DIR="/shared/projects/form_2021_26/data/atelier_rnaseq/01-Bioinfo/data"
R1="${DATA_DIR}/KO1_R1.fastq.gz"
R2="${DATA_DIR}/KO1_R2.fastq.gz"

module load fastqc/0.11.9

fastqc --outdir fastqc_res ${R1}

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

Utilisation d’une boucle

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

```
$ for PRENOM in Rachel Emilie Thibault Claire Stevenn Erwan
  do
    echo ${PRENOM}
  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.

Script3: automatisation sur plusieurs valeurs

```
#!/bin/bash

mkdir -p fastqc_res

DATA_DIR="/shared/projects/form_2021_26/data/atelier_rnaseq/01-Bioinfo/data"
R1="${DATA_DIR}/KO1_R1.fastq.gz"
R2="${DATA_DIR}/KO1_R2.fastq.gz"

module load fastqc/0.11.9

fastqc --outdir fastqc_res ${R1}

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

Script3: automatisation sur plusieurs valeurs

```
#!/bin/bash

mkdir -p fastqc_res

DATA_DIR="/shared/projects/form_2021_26/data/atelier_rnaseq/01-Bioinfo/data"



module load fastqc/0.11.9

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
```

Lancement du workflow

Lancez votre workflow avec une commande sbatch

/!\ Attention de réserver les ressources clusters dont vous avez besoins /!\

```
$ mkdir logs
$ sbatch -J workflow_fastqc -o logs/workflow_fastqc.out -e
logs/workflow_fastqc.err --cpus-per-task=4 fastqc.sh
```

Pour aller plus loin

Conseils pour écrire un workflow à plusieurs étapes

Conseils pour écrire un workflow à plusieurs étapes

1) écrire un script qui enchaîne l'ensemble des étapes pour 1 seul échantillon:

```
$ touch RNAseq_step.sh
```

2) définir des arguments pour ce script: R1 R2 GENOME ANNOT SAMPLENAME OUT_DIR LOG_DIR

3) écrire un script de lancement du script mapping.sh en boucle sur les différents échantillons (boucle "for" ou copier/coller de la ligne de commande mapping_calling.sh avec les nouvelles valeurs pour les arguments)

```
$ touch launch_RNASeq.sh
```

Définition d'arguments dans un script

Au lieu de donner une valeur à nos variables R1et R2 dans le script, on va indiquer au script d'aller les chercher dans la ligne de commande comme des options du programme.

Dans le script la valeur est remplacée par $1, $2, … $n et dans la ligne de commande on ajoute dans l'ordre les valeurs des variables 1, 2, … n.

echo.sh

```
#!/bin/bash

PRENOM='Rachel'; NOM='LEGENDRE'
echo ${PRENOM} ${NOM}
```

```
$ sh echo.sh
```

echo.sh

```
#!/bin/bash

PRENOM=$1; NOM=$2
echo ${PRENOM} ${NOM}
```

```
$ sh echo.sh Rachel LEGENDRE
```

RNAseq_step.sh

```
#!/bin/bash
R1=$1; R2=$2; GENOME=$3; ANNOT=$4; NAME=$5; OUT_DIR=$6; LOG_DIR=$7

module load fastqc/0.11.9; module load star/2.7.5a; module load subread/1.6.1 ...

mkdir -p ${OUT_DIR}/fastqc_res
fastqc --outdir ${OUT_DIR}/fastqc_res ${R1} 2>&1 ${LOG_DIR}/${NAME}_fastqc_R1.out
fastqc --outdir ${OUT_DIR}/fastqc_res ${R2} 2>&1 ${LOG_DIR}/${NAME}_fastqc_R2.out

mkdir -p ${OUT_DIR}/mapping_res
STAR --genomeDir ${GENOME} --runThreadN 4 --readFilesCommand zcat --outFileNamePrefix
${OUT_DIR}/mapping_res/${NAME}_ --readFilesIn ${R1} ${R2} --outSAMtype BAM
SortedByCoordinate --alignIntronMax 1000 --alignMatesGapMax 10000 --sjdbGTFfile
${ANNOT}

#... et on continue avec les autres étapes, samtools , featureCounts ...
```

launch_RNASeq.sh (version simple)

```
#!/bin/sh
GENOME="/shared/bank/arabidopsis_thaliana/TAIR10.1/star-2.7.5a/"
ANNOT="/shared/bank/arabidopsis_thaliana/TAIR10.1/gff/Arabidopsis_thaliana.TAIR10.1_genomic.gff"
OUT_DIR=$HOME"/tp_rnaseq/results"
DATA_DIR="/shared/projects/form_2021_26/data/atelier_rnaseq/01-Bioinfo/data"
LOG_DIR=${OUT_DIR}/logs; mkdir -p ${LOG_DIR}

# sample WT1
sbatch -J WT1_rnaseq -o ${LOG_DIR}/WT1_rnaseq.out \
     -e ${LOG_DIR}/WT1_rnaseq.err --cpus-per-task=4 --mem=30G \
     RNAseq_step.sh ${DATA_DIR}/WT1_1.fastq.gz \
     ${DATA_DIR}/WT1_2.fastq.gz ${GENOME} ${ANNOT} WT1 ${OUT_DIR} ${LOG_DIR}

# sample KO1
sbatch -J KO1_rnaseq -o ${LOG_DIR}/KO1_rnaseq.out \
     -e ${LOG_DIR}/KO1_rnaseq.err --cpus-per-task=4 --mem=30G \
     RNAseq_step.sh ${DATA_DIR}/KO1_1.fastq.gz \
     ${DATA_DIR}/KO1_2.fastq.gz ${GENOME} ${ANNOT} KO1 ${OUT_DIR} ${LOG_DIR}
```

launch_RNASeq.sh (version avec boucle)

```
#!/bin/sh
GENOME="/shared/bank/arabidopsis_thaliana/TAIR10.1/star-2.7.5a/"
ANNOT="/shared/bank/arabidopsis_thaliana/TAIR10.1/gff/Arabidopsis_thaliana.TAIR10.1_genomic.gff"
OUT_DIR=$HOME"/tp_RNAseq/results"
DATA_DIR="/shared/projects/form_2021_26/data/atelier_rnaseq/01-Bioinfo/data"
LOG_DIR=${OUT_DIR}/logs; mkdir -p ${LOG_DIR}

# all sample
for R1 in ${DATA_DIR}/*_R1.fastq.gz
do
    NAME=$(basename "${R1/_R1.fastq.gz/}")
    sbatch -J ${NAME}_rnaseq -o ${LOG_DIR}/${NAME}_rnaseq.out \
     -e ${LOG_DIR}/${NAME}_rnaseq.err --cpus-per-task=4 --mem=30G \
     RNAseq_step.sh  ${DATA_DIR}/${NAME}_R1.fastq.gz \
     ${DATA_DIR}/${NAME}_R2.fastq.gz ${GENOME} ${ANNOT} ${NAME} ${OUT_DIR} ${LOG_DIR}
done
```

launch_RNASeq.sh (version avec boucle)

```
#!/bin/sh
GENOME="/shared/bank/arabidopsis_thaliana/TAIR10.1/star-2.7.5a/"
ANNOT="/shared/bank/arabidopsis_thaliana/TAIR10.1/gff/Arabidopsis_thaliana.TAIR10.1_genomic.gff"
OUT_DIR="~/tp_RNAseq/results"
DATA_DIR="/shared/projects/form_2021_26/data/atelier_rnaseq/01-Bioinfo/data"
LOG_DIR=${OUT_DIR}/logs; mkdir -p ${LOG_DIR}

for R1 in ${DATA_DIR}/*_R1.fastq.gz
do
    NAME=`basename ${R1} | sed 's/_R1.fastq.gz//'`
    srun -c 4 STAR --genomeDir ${GENOME} --runThreadN 4 --readFilesCommand zcat
--outFileNamePrefix ${OUT_DIR}/${NAME}_ --readFilesIn ${DATA_DIR}/${NAME}_R1.fastq.gz
${DATA_DIR}/${NAME}_R2.fastq.gz --outSAMtype BAM SortedByCoordinate --alignIntronMax
1000 --alignMatesGapMax 10000 --sjdbGTFfile ${ANNOT}
    srun -c 1 samtools index ${OUT_DIR}/${NAME}_Aligned.sortedByCoord.out.bam
    srun -c 4 featureCounts -T 4 -t exon -g gene_id -s 1 -a ${ANNOT} -o
${OUT_DIR}/${NAME}_feature.out ${OUT_DIR}/${NAME}_Aligned.sortedByCoord.out.bam 2>
${OUT_DIR}/${NAME}_counting.logs
done
```

Lancement du workflow

Lancez votre workflow avec une commande sbatch

/!\ Attention de réserver les ressources clusters dont vous avez besoins /!\

```
$ sbatch -J workflow_rnaseq -o logs/workflow_rnaseq.out -e
logs/workflow_rnaseq.err --cpus-per-task=1 launch_RNAseq.sh
```