R Notebook

'Examen final Modules 4 et 5- DU Bioinformatique Intégrative'

title: 'Examen final Modules 4 et 5- DU Bioinformatique Intégrative' author: "Sandrine AUGER" date: "2020-08-26" output: html_document: code_folding: hide fig_caption: yes highlight: zenburn number_sections: no self_contained: no theme: cerulean toc: yes toc_depth: 3 toc_float: yes pdf_document: fig_caption: yes highlight: zenburn toc: yes toc_depth: 3 font-import: http://fonts.googleapis.com/css?family=Risque font-family: Garamond subtitle: DUBii 2020 - Module 6 - Evaluation editor_options: chunk_output_type: console transition: linear —

Description du projet

Les données sont issues de l'article : "Complete Genome Sequences of 13 Bacillus subtilis Soil Isolates for Studying Secondary Metabolite Diversity", 2019, (doi:10.1128/MRA.01406-19).

Matériel et méthodes

Paired-end reads were generated on an Illumina NextSeq sequencer using a TG NextSeq 500/550 high-output kit v. 2 (300 cycles). DNA sequencing was carried out on an Illumina MiSeq machine using V2 sequencing chemistry, resulting in 2 × 250-bp reads. Les données brutes ont été déposées sur GenBank. Le numéro d'accession du BioProject est PRJNA587401. Data Access : https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=SRR10390685

1. Préparation du dossier de travail

Le projet est réalisé sur le cluster de l'IFB.

```
cd ~ # positionnement au début de la racine de mon directory sauger
mkdir mkdir -p ~/M5_examen_final/CLEANING # p is short for --parents - it creates the entire directory
mkdir -p ~/M5_examen_final/MAPPING
mkdir -p ~/M5_examen_final/QC
mkdir -p ~/M5_examen_final/FASTQ
mkdir -p ~/M5_examen_final/CLEANING
tree ~/M5_examen_final # pour voir la structure du répertoire M5_examen_final
```

4 directories, 0 file

1.1. Téléchargement des données

Téléchargement depuis les banques publiques via le web en utilisant wget.

Téléchargement du run SRR10390685

Les données brutes étudiées sont issues du séquençage à haut débit par la technologie Illumina à partir des 2 extrémités ("paired-end sequencing").

```
cd ~/M5_examen_final/FASTQ
wget    https://sra-pub-src-2.s3.amazonaws.com/SRR10390685/P5_B1_S7_R1_001.fastq.gz.1
wget    https://sra-pub-src-2.s3.amazonaws.com/SRR10390685/P5_B1_S7_R2_001.fastq.gz.1
ls -ltrh ~/M5_examen_final/FASTQ/
```

Les fichiers P5_B1_S7_R1_001.fastq.gz.1 et P5_B1_S7_R2_001.fastq.gz.1 font au total 1.3G.

Changement du nom des fichiers

```
mv P5_B1_S7_R1_001.fastq.gz.1 SRR10390685_R1.fastq.gz
mv P5_B1_S7_R2_001.fastq.gz.1 SRR10390685_R2.fastq.gz
```

1.2. Téléchargement du génome de référence NC_000964

```
cd ~/M5_examen_final/MAPPING
wget https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/009/045/GCF_000009045.1_ASM904v1/GCF_000009045.1_
wget https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/009/045/GCF_000009045.1_ASM904v1/GCF_000009045.1_
```

Les données ont été téléchargées.

1.3. Décompression des fichiers gff et fna

```
cd ~/M5_examen_final/MAPPING
gzip -d GCF_000009045.1_ASM904v1_genomic.fna.gz # -d, --decompress
gzip -d GCF_000009045.1_ASM904v1_genomic.gff.gz
ls -ltrh ~/M5_examen_final
# -l, use a long listing format
# -t, sort by modification time, newest first
# -r, --reverse, reverse order while sorting
# -h, --human-readable, with -l, print sizes in human readable format
```

Le fichier GCF_000009045.1_ASM904v1_genomic.fna.gz fait 1.2M et le fichier GCF_000009045.1_ASM904
509K.

2. Contrôle qualité des données brutes

2.1. Recherche de la liste des outils de bioinformatique disponibles sur le cluster de l'IFB

```
module avail -l
```

2.2. Contrôle qualité avec FastQC

La qualité de séquençage est vérifiée avec le programme FastQC (Babraham Institute, Cambridge).

```
#! /bin/bash
#SBATCH -o fastqc-%j-slurm.out
#SBATCH -e fastqc-%j-slurm.err
module load fastqc/0.11.8
srun --cpus-per-task 8 fastqc M5_examen_final/FASTQ/SRR10390685_R1.fastq.gz -o QC/ -t 8
srun --cpus-per-task 8 fastqc M5_examen_final/FASTQ/SRR10390685_R2.fastq.gz -o QC/ -t 8
```

2.3. Résultat

Le **nombre de lecture est d'environ 7.1 M pour R1 et R2**. Le contenu en **GC est de 43%**. La longueur des reads est en moyenne de **150 nts**. Le nombre de lecture unique est d'environ 55 % et le nombre de **lectures dupliquées** est d'environ 45 %. La valeur moyenne de qualité pour chaque position de base dans la lecture est **Phred Scor e= 35**. Le pourcentage d'appels de base à chaque position pour lequel a été appelé N est très faible **(N < 0.1%)**. Dans le fichier R2, la séquence de l'adaptateur universel Illumina est détecté dans une faible proportion des reads. Conclusion : données de bonne qualité, avec une bonne profondeur, les adaptateurs Illumina ont déjà été enlevés.

3. Preprocess : nettoyage et filtrage des séquences avec fastp

L'outil de prétraitement fastp permet d'enlever les adaptateurs, de filtrer par la qualité et la longueur des reads. Le trimming des adaptateurs et leur autodétection sont activés par defaut. ### 3.1. Utilisation de fastp QUALITY FILTER fastp prend en charge la découpe de fenêtre coulissante par lecture en évaluant les scores de qualité moyens dans la fenêtre coulissante. **-W, –cut_window_size** range 1-1000, default=4. **-u, –unqualified_percent_limit**, how many percents of bases are allowed to be unqualified (0~100). Default 40 means 40%. **-q, –qualified_quality_phre**, the quality value that a base is qualified. Default 15 means phred quality >=Q15 is qualified. La limite maximale du nombre de **N** de 5 par read est laissée par défaut, elle peut être modifiée par l'option -n(-n_base_limit).

LENGTH FILTER **-l, –lenght_required**, seuil de longueur requis, par defaut = 15, reads shorter than lenght_limit will be discarded.

```
#! /bin/bash
module load fastp/0.20.0
cd ~/M5_examen_final
srun --cpus-per-task 8 fastp --in1 FASTQ/SRR10390685_R1.fastq.gz --in2 FASTQ/SRR10390685_R2.fastq.gz -l
ls -ltrh ~/M5_examen_final/CLEANING/
```

3.2. FastQC après le trimming et le filtrage

Les fichiers de sortie cleaned.fastq.gz sont analysés avec FastQC.

```
#! /bin/bash
#SBATCH -o fastqc-%j-slurm.out
#SBATCH -e fastqc-%j-slurm.err
cd ~/M5_examen_final/CLEANING
module load fastqc/0.11.8
srun --cpus-per-task 8 fastqc SRR10390685_R1.cleaned.fastq.gz -o CLEANING/ -t 8
srun --cpus-per-task 8 fastqc SRR10390685_R2.cleaned.fastq.gz -o CLEANING/ -t 8
```

3.3. Résultats

Globalement, plus de 96,5% des reads ont passé les filtres. Dans le fichier R2, les reads qui avaient des traces d'adaptateur Illumina ont bien été filtrés.

4. Alignement des reads sur le génome de référence

4.1. Indexation du génome de référence

```
#! /bin/bash
module load bwa/0.7.17
cd ~/M5_examen_final/MAPPING
srun bwa index GCF_000009045.1_ASM904v1_genomic.fna
```

4.2. Mapping des reads sur le génome

BWA (le Burrows-Wheeler Aligner) est un aligneur rapide à lecture courte. Il utilise la transformation burrows-wheeler pour effectuer l'alignement de manière efficace en termes de temps et de mémoire.

L'output est un fichier .sam. Ce fichier contient toutes les informations sur la position de chaque read sur le génome de référence. L'outil SAMTools est ensuite utilisé pour convertir le fichier .sam en .bam, le trier et l'indexer.

```
#! /bin/bash
#SBATCH --cpus-per-task=33
module load bwa/0.7.17
module load samtools/1.9
cd ~/M5_examen_final
srun bwa mem ./MAPPING/GCF_000009045.1_ASM904v1_genomic.fna.gz CLEANING/SRR10390685_R1.cleaned.fastq.gz
# --cpus-per-task: on parallèlse, on prend 32 noeuds pour la commande bwa mem et 1 noeud pour la comman
# on fait un pipe avec la commande samtools view.
# -b indique le format de sortie en .bam
# -h inclure le header
# -S le format de fichier d'entrée est détecté directement
```

```
module load samtools/1.9
samtools flagstat SRR10390685.bam # Get some statistics
samtools sort SRR10390685.bam -o SRR10390685_sorted.bam # Sort BAM file
samtools index SRR10390685_sorted.bam # Index sorted BAM file
```

4.3. Extraction des reads mappant à au moins 50% sur le gène *trmNF*

```
grep trmNF GCF_000009045.1_ASM904v1_genomic.gff | awk '$3=="gene"' > trmNF.gff3
# avec la commande grep on cherche les lignes qui contiennent le nom trmNF dans le fichier gff
# avec la commande awk '$3=="gene"', on s'assure que la colonne 3 contient une feature de type "gene"
# les caractéristiques (coordonnées, sens de transcription, ..) concernant le gène trmNF sont sauvegard
```

```
# avec l'outil bedtools intersect, on cherche combien de reads ont mappé avec le gène trmNF.
# To report the base-pair overlap between sequence alignments and genes: bedtools intersect -a reads.be
# To only report an overlap with at least 50% of the gene: option -f 0.5
module load bedtools/2.29.2
```

```
bedtools intersect -a SRR10390685_sorted.bam -b trmNF.gff3 -f 0.5 > SRR10390685_on_trmNF.bam
samtools view -c SRR10390685_on_trmNF.bam # option -c :instead of printing the alignments, only count t
```