Examen final Modules 4 et 5

DUBii 2021

Olivier Rué - Valentin Loux - étudiant: Camille Dejos

09 April, 2021

Contents

Consignes

Complétez ce document en remplissant les chunks vides pour écrire le code qui vous a permis de répondre à la question. Les réponses attendant un résultat chiffré ou une explication devront être insérés entre le balises html `code`. Par exemple pour répondre à la question suivante :

```
La bioinfo c'est : <code>MERVEILLEUX</code>.
```

N'hésitez pas à commenter votre code, enrichier le rapport en y insérant des résultats ou des graphiques/images pour expliquer votre démarche. N'oubliez pas les **bonnes pratiques** pour une recherche **reproductible** ! Nous souhaitons à minima que l'analyse soit reproductible sur le cluster de l'IFB.

Introduction

Vous allez travailler sur des données de reséquençage d'un génome bactérien : *Bacillus subtilis*. Les données sont issues de cet article :

- Complete Genome Sequences of 13 Bacillus subtilis Soil Isolates for Studying Secondary Metabolite Diversity

Analyses

Organisation de votre espace de travail

```
# choix du répertoire de travail
cd /shared/projects/dubii2021/cdejos/Module_4_production_omics/EvaluationM4M5-main

# organisation du répertoire de travail
mkdir -p ./CLEANING ./FASTQ ./MAPPING ./QC

#réserver des cpus
salloc --cpus-per-task=6 --mem=1G
#exit dans le terminal à la fin de la session
```

Téléchargement des données brutes

Récupérez les fichiers FASTQ issus du run **SRR10390685** grâce à l'outil sra-tools [1]

Dans l'article on trouve les informations suivantes:

Data availability. This whole-genome shotgun project has been deposited at DDBJ/ENA/GenBank under the accession number VNIP00000000, and the version described in this paper is version VNIP01000000. The SRA accession number for the raw data is PRJNA556568.

Puis sur le site du NCBI (https://www.ncbi.nlm.nih.gov/Traces/study/?acc=SRR10390685&o=acc_s%3Aa): Dans le BioProject PRJNA587401, SRA Study SRP228290, BioSample SAMN13190557, on trouve le run SRR10390685.

```
#téléchargement des fichiers à l'aide de la commande fasterq-dump de l'outils sra-tools
module load sra-tools
srun --cpus-per-task=6 fasterq-dump --split-files -p SRR10390685 --outdir FASTQ
#fasterq-dump version 2.10.3

#compression des fichiers SRR10390685_1.fastq et SRR10390685_2.fastq
cd FASTQ
srun gzip *.fastq
```

Combien de reads sont présents dans les fichiers R1 et R2 ?

Dans un fichier FASTQ, on a 4 lignes par read. Donc on divise le nombre total de lignes par 4.

```
cd FASTQ

#premier fichier
zcat SRR10390685_1.fastq.gz | echo $((`wc -l` / 4))

#second fichier
zcat SRR10390685_2.fastq.gz | echo $((`wc -l` / 4))
```

Les fichiers FASTQ contiennent 7 066 055 reads.

Téléchargez le génome de référence de la souche ASM904v1 de *Bacillus subtilis* disponible à cette adresse

```
#téléchargement du génome de référence de la souche ASM904v1 de Bacillus subtilis
wget https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/009/045/GCF_000009045.1_ASM904v1/GCF_000009045.1_
```

Quelle est la taille de ce génome ?

```
cd MAPPING

#chargement du module seqkit
module load seqkit
seqkit stat GCF_000009045.1_ASM904v1_genomic.fna.gz
```

La taille de ce génome est de 4 215 606 paires de bases.

Téléchargez l'annotation de la souche ASM904v1 de *Bacillus subtilis* disponible à cette adresse

```
#téléchargement des annotations de la souche ASM904v1 de Bacillus subtilis
wget https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/009/045/GCF_000009045.1_ASM904v1/GCF_000009045.1_
```

Combien de gènes sont connus pour ce génome ?

```
#décompresse le fichier gff pour pouvoir extraire les infos
cd MAPPING
gzip -d GCF_000009045.1_ASM904v1_genomic.gff.gz

#sélectionne la colonne 9 du fichier gff, qui contient dans sa première partie en amount du ";" des ID,
cut -f 9 GCF_000009045.1_ASM904v1_genomic.gff | cut -d ";" -f 1 | grep "ID=gene" | sort -u | wc -l
```

4 536 gènes sont recensés dans le fichier d'annotation.

Contrôle qualité

Lancez l'outil fastqc [2] dédié à l'analyse de la qualité des bases issues d'un séquençage haut-débit

```
#chargement du module fastqc
module load fastqc
fastqc --version
#fastQC v0.11.9

#analyse qualité du fichier R1, redirection des résultats dans le dossier QC
```

```
srun --cpus-per-task 6 fastqc FASTQ/SRR10390685_1.fastq.gz -o QC/ -t 8

#idem fichier R2
srun --cpus-per-task 6 fastqc FASTQ/SRR10390685_2.fastq.gz -o QC/ -t 8
```

La qualité des bases vous paraît-elle satisfaisante ? Pourquoi ?

- ☒ Oui
- ☐ Non

car il y a le même nombre de reads par fichier (7 066 055 reads) comme le montre la ligne “total sequences” du tableau “basic stastistics”

car les reads ont une taille attendue comme le montre la ligne “sequence length” du tableau “basisc statistics” (35-151 bp pour R1, et 130-151 bp pour R2)

car le profil qualité est bon comme le montre le pourcentage de GC observé, et le score de qualité supérieur à 28

car on ne détecte pas la présence d’adaptateurs comme le montre le graphique “adaptater content”

```
#création du rapport MultiQC
module load multiqc
#multiqc, version 1.9
srun multiqc -d . -o .
```

Lien vers le rapport MulitQC

Est-ce que les reads déposés ont subi une étape de nettoyage avant d’être déposés ? Pourquoi ?

- ☒ Oui
- ☐ Non

car dans les données brutes, les lectures sont de la même taille.

Quelle est la profondeur de séquençage (calculée par rapport à la taille du génome de référence) ?

```
# profondeur de séquençage = (nombre total de reads * taille moyenne d'un read) / taille du génome de r
DoS <- (7066055 * 150) / 4215606
DoS
```

La profondeur de séquençage est de : 250 X.

Nettoyage des reads

Vous voulez maintenant nettoyer un peu vos lectures. Choisissez les paramètres de fastp [3] qui vous semblent adéquats et justifiez-les.

```
#charger le module fastp
module load fastp
fastp --version
#fastp 0.20.0
```

```
#nettoyage et filtrage des reads
srun --cpus-per-task 6 fastp --in1 FASTQ/SRR10390685_1.fastq.gz --in2 FASTQ/SRR10390685_2.fastq.gz --ou

#combien de reads restent-ils dans R1 et R2?
seqkit stats CLEANING/*fastq.gz
```

Les paramètres suivants ont été choisis :

Parametre	Valeur	Explication
qualité des reads	>= 30	score de qualité d'au moins 30
longueur des reads	>=100	ne garder que les reads d'au moins 100 pb
paires	dans R1 et R2	ne garder que les reads présents à la fois dans R1 et R2

Ces paramètres ont permis de conserver 6 777 048 reads pairés, soit une perte de 4% des reads bruts.

Alignement des reads sur le génome de référence

Maintenant, vous allez aligner ces reads nettoyés sur le génome de référence à l'aide de bwa [4] et samtools [5].

```
#charger le module bwa pour indexer le génome de référence
module load bwa
#Version: 0.7.17-r1188

#indexer le génome de référence
srun bwa index MAPPING/GCF_000009045.1_ASM904v1_genomic.fna.gz

#aligner les reads nettoyés et filtrés de R1 et R2 sur le génome de référence indexé
srun --cpus-per-task=6 bwa mem MAPPING/GCF_000009045.1_ASM904v1_genomic.fna.gz CLEANING/SRR10390685_1.cl

#charger le module samtools
module load samtools
samtools --version
#samtools 1.10
#Using htslib 1.10.2

#convertir le fichier sam en bam
srun --cpus-per-task=6 samtools view --threads 6 SRR10390685_on_ASM904v1.sam -b > SRR10390685_on_ASM904

#trier le fichier bam
srun samtools sort SRR10390685_on_ASM904v1.bam -o SRR10390685_on_ASM904v1.sort.bam

#indexer le fichier bam
srun samtools index SRR10390685_on_ASM904v1.sort.bam

#déplacer les fichiers bam et sam vers le dossier MAPPING
mv SRR* MAPPING/
```

Combien de reads ne sont pas mappés ?

```
#statistiques sur le mapping
srun samtools idxstats MAPPING/SRR10390685_on_ASM904v1.sort.bam > MAPPING/SRR10390685_on_ASM904v1.sort.
srun samtools flagstat MAPPING/SRR10390685_on_ASM904v1.sort.bam > MAPPING/SRR10390685_on_ASM904v1.sort.

# total: 13 571 369
# mapped : 12 826 829 (94.5%)
# non_mappés = total - mapped = 13 571 369 - 12 826 829 = 744 540 (soit 5.5%)
```

744 540 reads ne sont pas mappés.

Croisement de données

Calculez le nombre de reads qui chevauchent avec au moins 50% de leur longueur le gène *trmNF* grâce à l'outil bedtools [6]:

```
#Etape1: récupérer les positions de la séquence génomique du gène trmNF
#retrouver la ligne qui contient l'identifiant de gène trmNF dans le fichier d'annotation
grep trmNF MAPPING/GCF_000009045.1_ASM904v1_genomic.gff | awk '$3=="gene"' > MAPPING/trmNF_gene.gff
#le gène trmNF (NC_000964.3) se trouve aux positions: 42917-43660


#Etape 2: croissement des données des reads mappés sur le génome avec le gène trmNF
#récupérer les alignements sur le gène
#option -f 0.5, pour avoir au moins 50% de la séquence d'un read aligné sur le gène
srun bedtools intersect -f 0.5 -bed -a MAPPING/SRR10390685_on_ASM904v1.sort.bam -b MAPPING/trmNF_gene.g

#compter le nombre de lignes (1 read par ligne)
wc -l MAPPING/SRR10390685_on_TrmNF.bed
```

2 801 reads chevauchent le gène d'intérêt.

Visualisation

Utilisez IGV [7] sous sa version en ligne pour visualiser les alignements sur le gène. Faites une capture d'écran du gène entier.

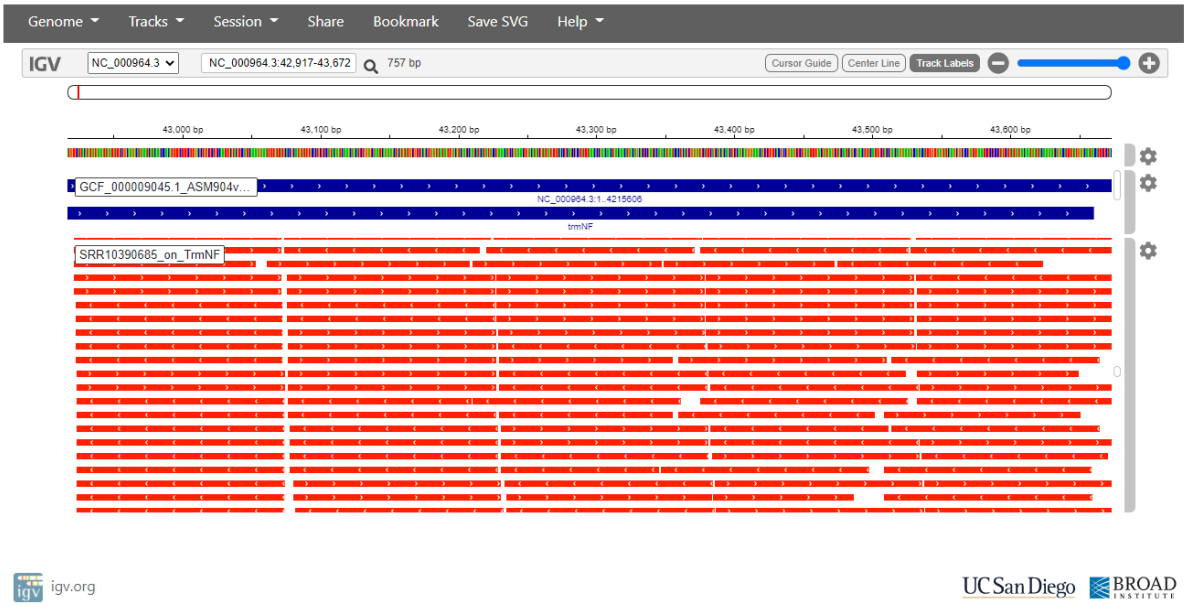

Figure 1: Visualisation IGV des lectures sur le gène TrmNF

```
.
├── CLEANING
│   ├── fastp.html
│   ├── fastp.json
│   ├── SRR10390685_1.cleaned_filtered.fastq.gz
│   └── SRR10390685_2.cleaned_filtered.fastq.gz
├── core.43818
├── css
│   └── style.css
├── Evaluation_cdejos.html
├── Evaluation_cdejos.log
├── Evaluation_cdejos.Rmd
├── Evaluation_cdejos.tex
├── Evaluation.html
├── EvaluationM4M5.Rproj
├── Evaluation.Rmd
├── FASTQ
│   ├── SRR10390685_1.fastq.gz
│   └── SRR10390685_2.fastq.gz
├── images
│   ├── IGV_cdejos.png
│   ├── inrae.png
│   └── migale-orange.png
├── MAPPING
│   ├── GCF_000009045.1_ASM904v1_genomic_cp.fna
│   ├── GCF_000009045.1_ASM904v1_genomic_cp.fna.fai
│   ├── GCF_000009045.1_ASM904v1_genomic.fna.gz
│   ├── GCF_000009045.1_ASM904v1_genomic.fna.gz.amb
│   ├── GCF_000009045.1_ASM904v1_genomic.fna.gz.ann
│   ├── GCF_000009045.1_ASM904v1_genomic.fna.gz.bwt
│   ├── GCF_000009045.1_ASM904v1_genomic.fna.gz.pac
│   ├── GCF_000009045.1_ASM904v1_genomic.fna.gz.sa
│   ├── GCF_000009045.1_ASM904v1_genomic.gff
│   ├── SRR10390685_on_ASM904v1.bam
│   ├── SRR10390685_on_ASM904v1.sam
│   ├── SRR10390685_on_ASM904v1.sort.bam
│   ├── GCF_000009045.1_ASM904v1_genomic.fna.gz.ann
│   ├── GCF_000009045.1_ASM904v1_genomic.fna.gz.bwt
│   ├── GCF_000009045.1_ASM904v1_genomic.fna.gz.pac
│   ├── GCF_000009045.1_ASM904v1_genomic.fna.gz.sa
│   ├── GCF_000009045.1_ASM904v1_genomic.gff
│   ├── SRR10390685_on_ASM904v1.bam
│   ├── SRR10390685_on_ASM904v1.sam
│   ├── SRR10390685_on_ASM904v1.sort.bam
│   ├── SRR10390685_on_ASM904v1.sort.bam.bai
│   ├── SRR10390685_on_ASM904v1.sort.bam.flagstat
```

References

1. toolkit NS. NCBI SRA toolkit. NCBI, GitHub repository. 2019.

2. Andrews S. FastQC a quality control tool for high throughput sequence data. http://www.bioinformatics.babraham.ac.uk/p http://www.bioinformatics.babraham.ac.uk/projects/fastqc/.

3. Zhou Y, Chen Y, Chen S, Gu J. Fastp: An ultra-fast all-in-one FASTQ preprocessor. Bioinformatics. 2018;34:i884–90. doi:10.1093/bioinformatics/bty560.

4. Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv preprint arXiv:13033997. 2013.

5. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The sequence alignment/map format and SAMtools. Bioinformatics. 2009;25:2078–9.

6. Quinlan AR, Hall IM. BEDTools: A flexible suite of utilities for comparing genomic features. Bioinformatics. 2010;26:841–2.

7. Thorvaldsdóttir H, Robinson JT, Mesirov JP. Integrative genomics viewer (IGV): High-performance genomics data visualization and exploration. Briefings in bioinformatics. 2013;14:178–92.