Rapport projet Nextflow

Margaux Dore - BBS

October 11, 2023

Introduction

Nextflow est un outil qui facilite la création et la gestion de workflows. Au cours de ce projet, nous avons pu explorer cet outil au travers de quatre exercices.

Exercice n°1 :

Dans ce premier exercice, nous avons dû télécharger plusieurs fichiers (utilisés lors du TP précédent) :

- Le génome de référence de la tomate
- Le fichier d'annotation GTF de ce génome de référence
- Quatre fichiers fastq. Ces données sont composées d'un réplicat wild type et d'un réplicat mutant (deux fichiers par réplicat)

Figure 1: Fichiers téléchargés

Exercice n°2 :

Dans ce deuxième exercice, nous avons utilisé le pipeline nf-core/rnaseq pour analyser les données téléchargées précédemment. Le projet nf-core a pour but de proposer et de maintenir de manière collaborative des pipelines d'analyses de bioinformatique en Nextflow selon des standards stricts de qualité et de reproductibilité. Le pipeline rnaseq est utilisé pour analyser des données rnaseq à l'aide du génome de l'organisme de référence et de son annotation.

Dans les grandes lignes, il effectue une série de pré-traitements, d'alignements et de post-traitements avant d'effectuer un contrôle qualité suivi d'un pseudo-alignement pour finir par générerer une matrice d'expression génique ainsi qu'un rapport de contrôle qualité détaillé.

Pour pouvoir lancer ce pipeline, il faut en amont créer plusieurs fichiers. Tout d'abord, il nous faut un fichier de configuration qui permet d'enregistrer les informations de suivi du pipeline (tel que les lignes de commandes lancées à chaque étape du pipeline) dans un fichier spécifique (run pipeline.txt). Dans ce fichier de configuration, nous avons aussi défini les champs que nous voulons enregistrer dans le fichier texte grâce à la variable field.

```
gerbera@genologin1 ~/work/Projet/Exo_2 $ more sm_config.cfg
trace {
        enabled = true
        file = 'pipeline_trace.txt'
        fields = 'task_id,name,status,exit,realtime,%cpu,rss,script'
}
```

Figure 2: Fichier de configuration

Il nous faut ensuite un fichier "inputs" qui contient la description des fastq qui seront utilisés en entrée du pipeline. Ce fichier renseigne notamment sur le type des réplicats (ici wild type ou mutant), le chemin d'accès à ces fichiers ainsi que la multiplicité des brins (strandedness).

```
gerbera@genologin1 ~/work/Projet/Exo_2 $ more inputs.csv
group,replicate,fastq_1,fastq_2,strandedness
mutant,1,/home/gerbera/work/Projet/Data_tomates/Fastq/MT_rep1_1_Ch6.fastq.gz,/home/gerbera/work/Projet/Data_tomates/Fastq/MT_rep1_2_Ch6.fastq.gz,unstranded
wild,1,/home/gerbera/work/Projet/Data_tomates/Fastq/WT_rep1_1_Ch6.fastq.gz,/home/gerbera/work/Projet/Data_tomates/Fastq/WT_rep1_2_Ch6.fastq.gz,unstranded
```

Figure 3: Fichier input

Enfin, il nous faut un fichier bash qui va lancer le pipeline. Les premières lignes du fichier permettent de définir le nom du travail (Margaux Dore), sur quelle partition le travail sera executé (workq), la quantité de mémoire auquel il aura accés (6 Go) et la dernière ligne spécifie que le temps maximum que peut prendre le travail est d'une journée. Les deux lignes suivantes permettent de charger le module nf-core de Nextflow. Puis, les quatre lignes suivantes permettent de définir 4 variables correspondant aux 4 fichiers nécessaires pour faire tourner le pipeline rnaseq : le fichier inputs (input), le fichier fasta du génome de référence (fasta), le fichier d'annotation de ce génome (gtf) et le fichier de configuration (config). Chaque variable contient le chemin absolu de chaque fichier. Enfin, la dernière ligne de ce fichier bash est celle qui permet de lancer le pipeline :

- -r 3.0 : indique que nous voulons utiliser la révision 3.0 du pipeline
- -profile genotoul : indique que le profil est génotoul
- –input input : indique que le fichier à prendre en entrée correspond à la variable input.
- –fasta fasta : indique que le chemin vers le génome de référence est dans la variable fasta.
- –gtf gtf : indique que le chemin vers l'annotation du génome est dans la variable gtf.
- –aligner star rsem M : indique que l'algorithme d'alignement à utiliser est star rsem
- –c config : indique que la configuration à suivre se trouve dans la variable config.

```
gerbera@genologin1 ~/work/Projet/Exo_2 $ more run_pipeline.sh
#!/bin/bash
#SBATCH -J MargauxDore
#SBATCH -p workq
#SBATCH --mem=6G

module purge
module load bioinfo/nfcore-Nextflow-v21.04.1

input=/home/gerbera/work/Projet/Exo_2/inputs.csv
gtf=/home/gerbera/work/Projet/Data_tomates/Annotation/ITAG2.3_genomic_Ch6.gtf
fasta=/home/gerbera/work/Projet/Data_tomates/Genome/ITAG2.3_genomic_Ch6.fasta
config=/home/gerbera/work/Projet/Exo_2/sm_config.cfg

nextflow run nf-core/rnaseq -r 3.0 -profile genotoul --input $input --fasta $fasta --gtf $gtf --aligner star_rsem M -c $config
```

Figure 4: Fichier run

Après avoir lancé le travail, il est possible de suivre son avancé à l'aide de la commande seff. Cette commande affiche les ressources utilisées par le travail donné et calcule son efficacité.

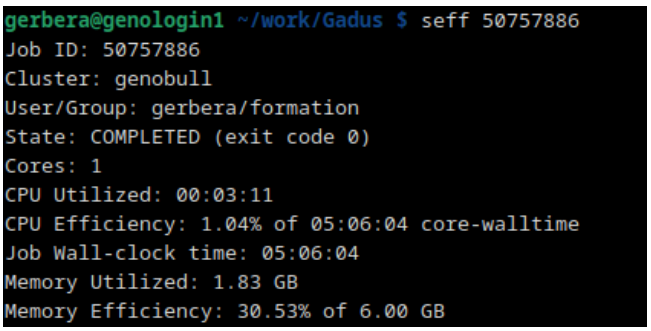

Figure 5: Sortie de la commande seff

L'option resume se place dans le fichier bash, à la fin de la ligne de commande permettant de lancer le pipeline. En cas d'erreur lors du traitement du travail, cette commande permet de reprendre le traitement là où il s'est arrété une fois l'erreur corrigée. Cela permet de ne pas reprendre le traitement depuis le début.

Exercice n°3 :

Une fois le travail terminé, plusieurs fichiers sont créés.

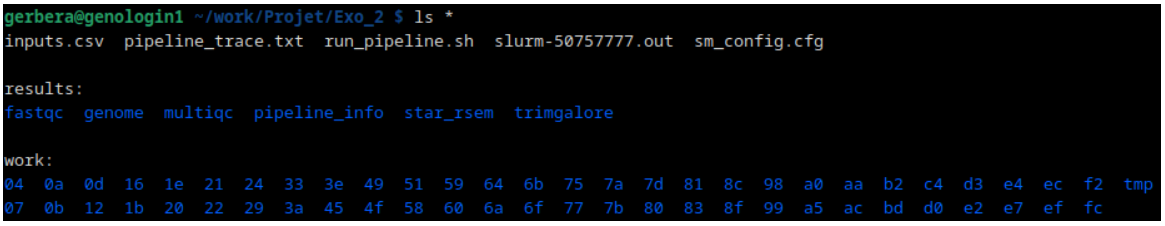

Figure 6: Fichiers après la pipeline

Le fichier pipeline trace.txt contient les informations de suivi du pipeline (cf exercice 1) tandis que le fichier slurm-50757777.out correspond au fichier de sortie et contient aussi des information sur le pipeline et le travail qui vient de terminer dont par exemple, le temps d'execution.

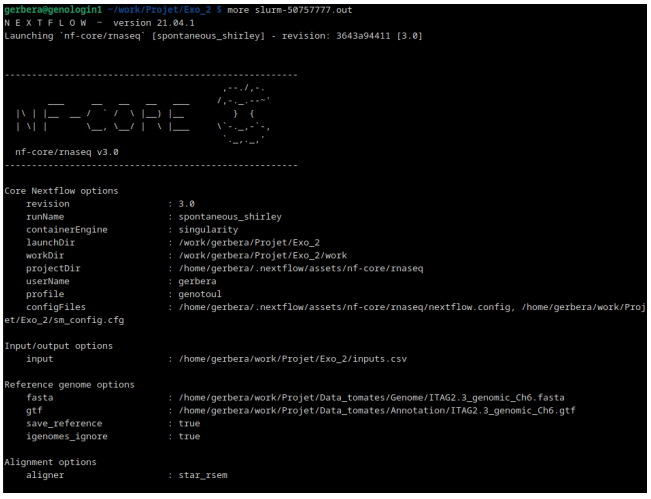

Figure 7: Premières ligne du fichier de sortie

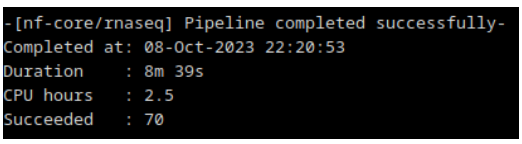

Figure 8: Temps de traitement

En plus de ces deux fichiers, deux nouveaux répertoire sont présents : results et work. Nous nous sommes intéressés au repertoire results qui contient les outputs des différentes étapes du pipeline ainsi que le rapport MultiQC. Ce rapport est un rapport de contrôle qualité du pipeline et permet d'avoir une vue d'ensemble sur les résultats et les métriques des différents outils utilisés (FastQC, Cutadapt, STAR, RSEM...).

Interprétation du rapport MultiQC

Dans cette partie d'analyse des résultats, je me suis concentrée sur ce qui me semblait le plus pertinent. Il y a d'autres résultats sur le rapport multiqc qui ne me semblait pas apporter d'informations clés pour notre analyse.

General Statistics : La première partie du rapport nous donne des métriques sur l'ensemble des jeux de données. Certaines sont plus pertinentes que d'autres dans notre cas comme par exemple la proportion des données rnaseq qui peuvent être alignées sur le génome de référence : il est d'environ 99 pourcents pour le wild type et le mutant ce qui est très bon. Il indique aussi le pourcentage de ARNr (0 pourcent pour nos données) dans les réplicats ainsi que le pourcentage de GC qui sont tous deux satisfaisants pour nos données.

Biotype Counts : Cette partie indique pour chaque échantillon, le nombre de reads alignés sur des biotypes différents du génome de référence. Pour nous, tous les reads s'alignent sur des parties du génome codant pour des protéines.

FastQC (trimmed) - Sequence Quality Histograms : Ce graphique nous permet d'évaluer la qualité des reads, dans notre cas après trimmage. Il représenta la qualité de chaque base pour tous les reads de nos échantillons. Nous remarquons que la qualité des reads est bonne même après trimmage, ce qui nous conforte pour la suite.

FastQC (trimmed) - Per sequence quality scores : Ce graphique représente quant à lui la qualité moyenne par reads. Nous nous attendons à ce que la courbe soit haute pour les scores les plus élevés et c'est ce que nous pouvons observer pour nos données.

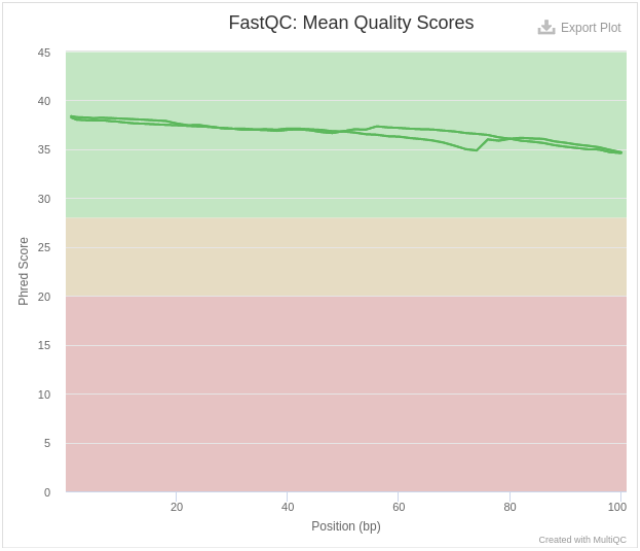

Figure 9: Qualité des bases de chaque reads

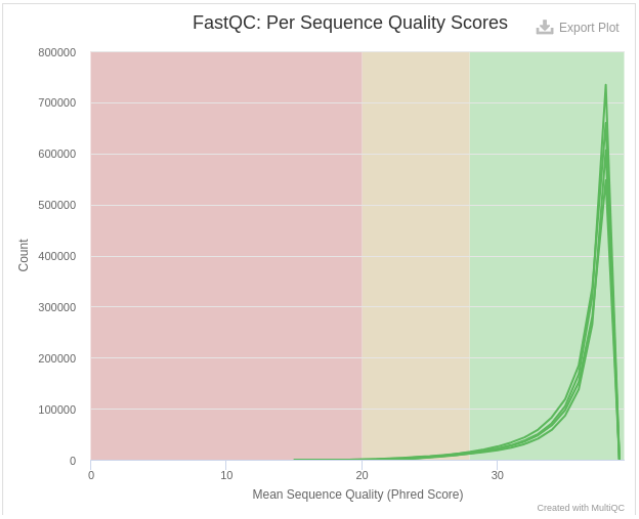

Figure 10: Qualité moyenne par reads

FastQC (trimmed) - Per Base Sequence Content : Cette partie du rapport fastqc représente le pourcentage de chaque base à chaque position des reads. Pour nos données, les résultats sont très mauvais. Il serait intéressant de faire d'autres analyses pour comprendre ce résultat.

FastQC (trimmed) - Sequence Length Distribution : Ce graphique-ci représente la distribution des longueurs des reads. Comme nos données de séquençages ont été obtenus avec Illumina, nous nous attendons à avoir en grande majorité des reads de même taille. C'est ce que l'on peut observer ici, ce qui nous permet de confirmer que le trimmage s'est bien passé.

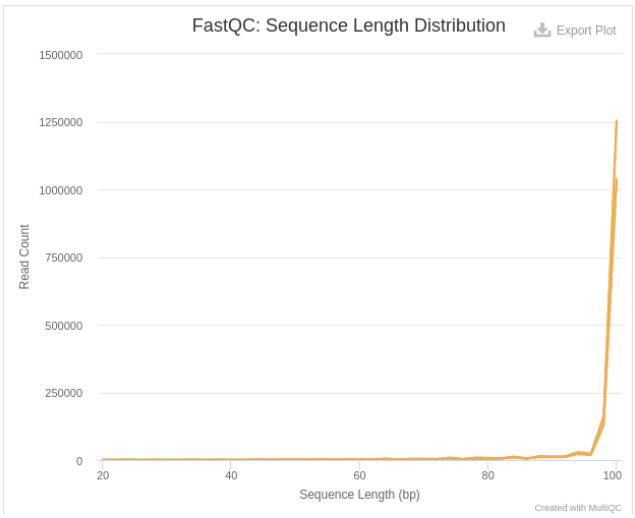

Figure 11: Distribution des longueurs de reads

FastQC - Status Checks : Enfin, nous avons un graphique qui résume l'ensemble des résultats FastQC grâce à un code couleur. Si nous comparons ces résultats avant et après trimmage, nous pouvons voir que dans l'ensemble des données sont de bonne qualité malgré que la composition des reads ne sont pas bonnes. Il serait quand même intéressant de faire quelques analyses pour mieux comprendre ce résultat.

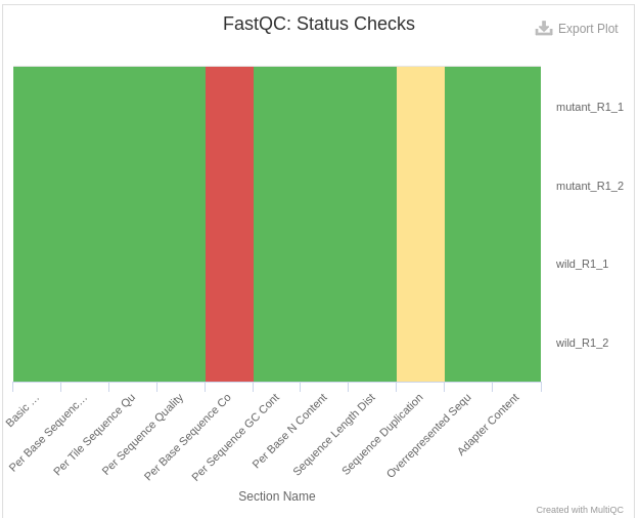

Figure 12: Avant trimmage

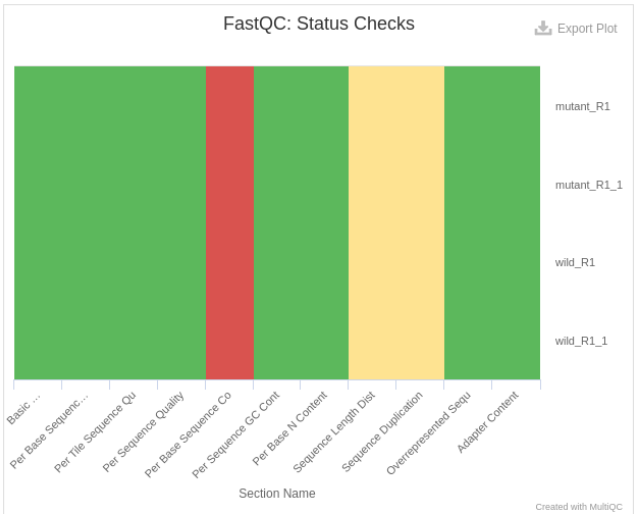

Figure 13: Après trimmage

Malgré tout, nous pouvons conclure que les reads sont d'assez bonne qualité et donc, les analyses faites avec ceux-ci sont exploitables.

Exercice n°4 :

Dans cet exercice, nous avons appliqué le pipeline précédent à de nouvelles données.

Les données de rna-seq que nous avons utilisées sont issues d'un article portant sur l'évolution des gènes et l'évolution de leur expression après une duplication du génome chez les poissons. Pour notre analyse, nous nous sommes concentrés sur trois fichiers de séquençage (Illumina) de la morue de l'atlantique (gadus morhua) correspondant à trois organes de l'animal : son cerveau, ses ovaires et ses branchies (les fichiers étant trop conséquents, nous avons du les téléchargers en deux fichiers, nous avons donc au total 6 fichiers fastq). Notre plan d'expérience pour cet exercice est donc différent de celui des exercices précédents.

Nous avons récupéré le génome de référence de Gadus morhua ainsi que le fichier d'annotation GTF associé sur Ensembl avant de créer les fichiers nécessaires au pipeline nf-core/rnaseq : le fichier de configuration, le

fchier input et le fichier bash pour lancer l'analyse. Ces fichiers sont similaires à ceux créer précédemment, nous avons juste modifié les chemins d'accès.

```
gerbera@genologin1 ~/work/Gadus $ more sm_config.cfg
trace {
        enabled = true
        file = 'pipeline_trace_gadus.txt'
        fields = 'task_id,name,status,exit,realtime,%cpu,rss,script'
}
```

Figure 14: Fichier de configuration

```
gerbera@genologin1 ~/work/Gadus $ more inputs.csv
group,replicate,fastq_1,fastq_2,strandedness
ovary,1,/home/gerbera/work/Gadus/Fastq/SRR2045415_1.fastq.gz,/home/gerbera/work/Gadus/Fastq/SRR2045415_2.fastq.gz,unstranded
brain,1,/home/gerbera/work/Gadus/Fastq/SRR2045416_1.fastq.gz,/home/gerbera/work/Gadus/Fastq/SRR2045416_2.fastq.gz,unstranded
gills,1,/home/gerbera/work/Gadus/Fastq/SRR2045417_1.fastq.gz,/home/gerbera/work/Gadus/Fastq/SRR2045417_2.fastq.gz,unstranded
```

Figure 15: Fichier input

```
gerbera@genologin1 ~/work/Projet/Exo_2 $ more run_pipeline.sh
#!/bin/bash
#SBATCH -J MargauxDore
#SBATCH -p workq
#SBATCH --mem=6G
#SBATCH --time=1-00:00:00

module purge
module load bioinfo/nfcore-Nextflow-v21.04.1

input=/home/gerbera/work/Projet/Exo_2/inputs.csv
gtf=/home/gerbera/work/Projet/Data_tomates/Annotation/ITAG2.3_genomic_Ch6.gtf
fasta=/home/gerbera/work/Projet/Data_tomates/Genome/ITAG2.3_genomic_Ch6.fasta
config=/home/gerbera/work/Projet/Exo_2/sm_config.cfg

nextflow run nf-core/rnaseq -r 3.0 -profile genotoul --input $input --fasta $fasta --gtf $gtf --aligner star_rsem M -c $config
```

Figure 16: Fichier run

```
-
-[nf-core/rnaseq] Pipeline completed successfully-
Completed at: 10-Oct-2023 06:24:23
Duration    : 5h 5m 35s
CPU hours   : 92.1
Succeeded   : 100
```

Figure 17: Temps de traitement

Le travail que nous avons lancé a tourné pendant un peu plus de 5h. De nouveau nous nous sommes intéressés au rapport multiqc pour pouvoir conclure sur cette analyse.

Interprétation du rapport MultiQC

A la différence des données précédentes, cette fois-ci nous avons 3 échantillons.

Les statistiques générales sont plutôt bonnes avec des pourcentages d'alignements supérieurs à 60.

Pour ce qui est des résultats fastQC, la qualité de chaque base pour tous les reads et la qualité moyenne par reads après trimmage sont très très satisfaisantes pour tous les échantillons.

Cependant, le pourcentage de chaque base à chaque position des reads est très mauvais (Per base sequence content), Nous nous attendons à des quantités équivalentes de chaque base mais ce n'est pas le cas ici. De plus, les réplicats du cerveaux sont composés de beaucoup plus de duplications que les autres réplicats.

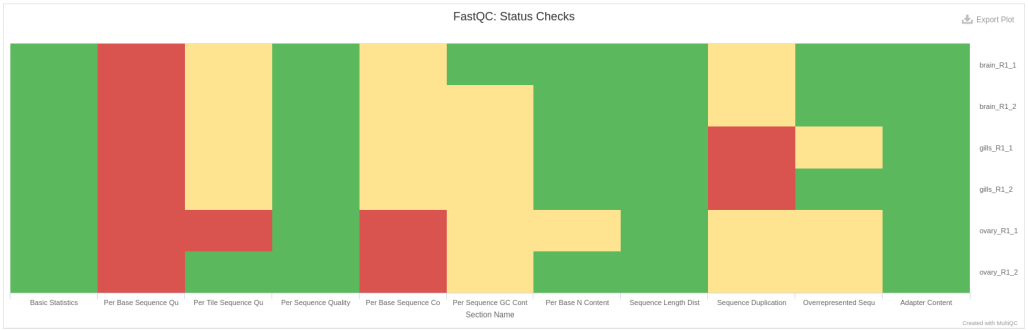

Figure 18: Avant trimmage

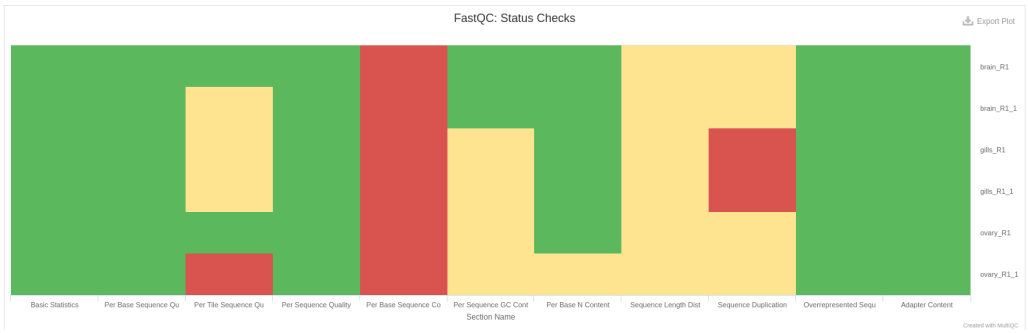

Figure 19: Après trimmage

Ces données ne sont pas excellentes mais devraient être utilisables pour des analyses. les résultats du pipeline sont donc exploitables. Cependant, il serait interessant de faire d'autres analyses pour identifier la source des diférents problèmes de qualité pour essayer de les améliorer et donc obtenir des résultats plus fiables.