M2 Bioinformatique et biologie des systèmes

2022/2023

Compte rendu du TP Nextflow

Initiation à genologin, manipulation de Nextflow sous forme de ligne de commande dans le terminal et interprétation de résultats FastQC

Rédigé par :

Naomi SCHICKELE

Introduction

Le flux de données est de plus en plus important, il est donc primordial de traiter les données de façon rapide et efficace. C'est pour cela que des logiciels tel que NextFlow sont créés. Nextflow est en effet un logiciel open source et puissant gestionnaire de flux de travail qui permet la manipulation de données dans différents langages de programmation (Bash, Perl, Python…) et permet de lancer plusieurs travaux en même temps. Globalement, NextFlow facilite assez bien la vie au bioinformaticien.

Lors de ce TP, nous avons travaillé sur des données de tomates. Nous avons utilisé un environnement NextFlow afin de lancer un pipeline de nf-score/rnaseq et d'obtenir en sortie des analyses de qualité contrôle de nos données. Les résultats des analyses sont représentés à l'aide de deux outils : FastQC et MultiQC (comprenant FastQC).

Exercice 1 : Connexion à Genologin, création d'un répertoire de travail, téléchargement de fichiers à traiter.

La première chose à faire durant ce TP a été de se connecter à un compte genotoul. Pour cela nous avions une adresse et un mot de passe relié à un compte défini pour chacun.
Pour ma part, il s'agissait du compte 'anemone'.

Nous devions exécuter la commande ssh suivante :

```
(base) ### Fri Sep 30 07:39  naomi pc15 ~
$ ssh -XY anemone@genologin.toulouse.inrae.fr
anemone@genologin.toulouse.inrae.fr's password:
Last login: Thu Sep 29 14:45:34 2022 from p0-16-4tp4.ups-tlse.fr
```

l'identifiant est l'adresse mail : anemone@genologin.toulouse.inrae.fr et le mot de passe : f1o2r3!
Une fois connectés à notre compte genotoul, nous devions créer un espace de travail. le mien était : /work/anemone/projet_nextflow

Puis nous devions télécharger les différents fichiers qu'on nous allions utiliser pour la suite du TP. Nous avions à disposition les 6 fichiers via un lien internet. Nous n'avions juste qu'à utiliser la commande 'wget' suivi du lien où se trouvait le fichier souhaité pour le télécharger.

```
anemone@genologin2 /work/anemone/projet_nextflow $ wget http://genoweb.toulouse.inra.fr/~sigenae/sarah/UPS/DATA/TP_TOMATES/ITAG2.3_genomic_Ch6.fasta
--2022-09-30 09:32:13--  http://genoweb.toulouse.inra.fr/~sigenae/sarah/UPS/DATA/TP_TOMATES/ITAG2.3_genomic_Ch6.fasta
Resolving genoweb.toulouse.inra.fr (genoweb.toulouse.inra.fr)... 147.99.108.69
Connecting to genoweb.toulouse.inra.fr (genoweb.toulouse.inra.fr)|147.99.108.69|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 46617169 (44M) [text/plain]
Saving to: 'ITAG2.3_genomic_Ch6.fasta'

100%[======================================================================================================================================================>]

2022-09-30 09:32:14 (378 MB/s) - 'ITAG2.3_genomic_Ch6.fasta' saved [46617169/46617169]

anemone@genologin2 /work/anemone/projet_nextflow $ wget http://genoweb.toulouse.inra.fr/~sigenae/sarah/UPS/DATA/TP_TOMATES/ITAG2.3_genomic_Ch6.gtf
--2022-09-30 09:32:41--  http://genoweb.toulouse.inra.fr/~sigenae/sarah/UPS/DATA/TP_TOMATES/ITAG2.3_genomic_Ch6.gtf
Resolving genoweb.toulouse.inra.fr (genoweb.toulouse.inra.fr)... 147.99.108.69
Connecting to genoweb.toulouse.inra.fr (genoweb.toulouse.inra.fr)|147.99.108.69|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2034585 (1.9M) [text/plain]
Saving to: 'ITAG2.3_genomic_Ch6.gtf'

100%[======================================================================================================================================================>]

2022-09-30 09:32:41 (273 MB/s) - 'ITAG2.3_genomic_Ch6.gtf' saved [2034585/2034585]

anemone@genologin2 /work/anemone/projet_nextflow $ wget http://genoweb.toulouse.inra.fr/~sigenae/sarah/UPS/DATA/TP_TOMATES/MT_rep1_1_Ch6.fastq.gz
--2022-09-30 09:33:19--  http://genoweb.toulouse.inra.fr/~sigenae/sarah/UPS/DATA/TP_TOMATES/MT_rep1_1_Ch6.fastq.gz
Resolving genoweb.toulouse.inra.fr (genoweb.toulouse.inra.fr)... 147.99.108.69
Connecting to genoweb.toulouse.inra.fr (genoweb.toulouse.inra.fr)|147.99.108.69|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 128793000 (123M) [application/x-gzip]
Saving to: 'MT_rep1_1_Ch6.fastq.gz'

100%[======================================================================================================================================================>]

2022-09-30 09:33:19 (447 MB/s) - 'MT_rep1_1_Ch6.fastq.gz' saved [128793000/128793000]
```

Les 6 fichiers à télécharger étaient les suivant :

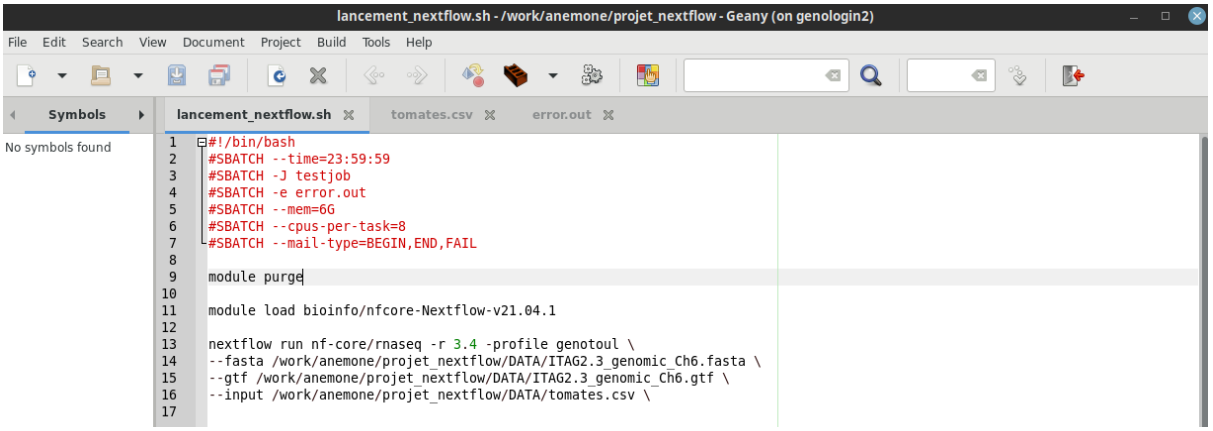
- 2 fichiers contenant le génome de référence au format *.fasta* et *.gtf*
- 2 fichiers contenant les 2 réplicats d'un sample de WT au format *.fastq* compressé
- 2 fichiers contenant les 2 réplicats d'un sample de MT au format *.fastq* compressé

Nous devions également faire attention à organiser notre répertoire en sous-répertoire, c'est donc pour cela que j'ai placé ces fichiers dans un sous-répertoire nommé DATA.

```
anemone@genologin2 /work/anemone/projet_nextflow $ mkdir DATA
anemone@genologin2 /work/anemone/projet_nextflow $ ls
DATA  ITAG2.3_genomic_Ch6.fasta  ITAG2.3_genomic_Ch6.gtf  MT_rep1_1_Ch6.fastq.gz  MT_rep1_2_Ch6.fastq.gz  WT_rep1_1_Ch6.fastq.gz  WT_rep1_2_Ch6.fastq.gz
anemone@genologin2 /work/anemone/projet_nextflow $ mv ITAG2.3_genomic_Ch6.fasta DATA
anemone@genologin2 /work/anemone/projet_nextflow $ mv ITAG2.3_genomic_Ch6.gtf DATA
anemone@genologin2 /work/anemone/projet_nextflow $ mv MT_rep1_1_Ch6.fastq.gz DATA
anemone@genologin2 /work/anemone/projet_nextflow $ mv MT_rep1_2_Ch6.fastq.gz DATA
anemone@genologin2 /work/anemone/projet_nextflow $ mv WT_rep1_1_Ch6.fastq.gz DATA
anemone@genologin2 /work/anemone/projet_nextflow $ mv WT_rep1_2_Ch6.fastq.gz DATA
anemone@genologin2 /work/anemone/projet_nextflow $ ls
```

Exercice 2 : Préparation du fichier bash au lancement NextFlow

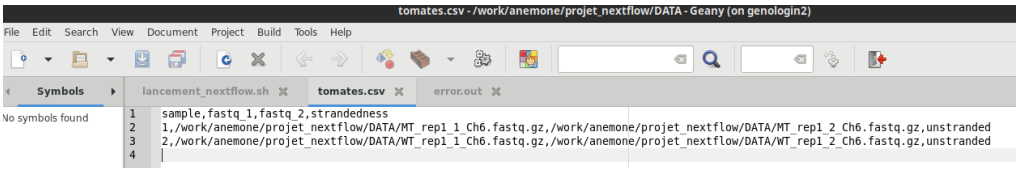
Cet exercice consistait à créer un fichier *.sh* contenant l'ensemble des commandes et les options nécessaires au bon lancement de NextFlow nf-core/rnaseq sur notre jeu de données de tomates.
La première étape à donc été de créer le fichier *lancement_nextflow.sh* à l'aide de la commande 'touch'.
Je me suis aidée d'un des diaporamas de cours ainsi que les instructions données dans le fichier de TP pour écrire le script sbatch.



Je viens de me rendre compte en rédigeant mon rapport que j'ai oublié de spécifier mon nom et mon prénom après l'option -J à la place de testjob.

Pour la ligne de commande run, la révision de la version 3.4 avec la version 22.04.1 de nfcore/nextflow a été utilisée.
-profile : sur quelle interface on lance
--fasta : le chemin du fichier *.fasta* correspondant au génome de référence
--gtf : le chemin du fichier *.gtf* correspondant au génome de référence
--input : le chemin du fichier *.csv* qu'on prend en entrée

le fichier *tomates.csv* a été construit de la façon suivante :



La première ligne correspond aux différentes colonnes de notre tableau renseignant sur les données que l'on va fournir en entrée.
Les deux autres lignes correspondent à nos 2 échantillons. Pour cela on va rentrer : le numéro du sample (qui va correspondre au sample WT ou MT), le chemin où l'on retrouve la séquence de du premier réplicat au format *.fastq* compressé, le chemin où l'on retrouve la séquence de du deuxième réplicat au format *.fastq* compressé et le sens du brin (si inconnu alors mettre "unstranded").

Une fois l'ensemble des fichiers créés, j'ai lancé la commande

> sbatch lancement_nextflow.sh

Cela a permis de lancer le job. Nous pouvions suivre le job en utilisant la commande suivante : seff suivi du numéro du job.

```
anemone@genologin2 /work/anemone/projet_nextflow $ seff 37374828
Job ID: 37374828
Cluster: genobull
User/Group: anemone/formation
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 8
CPU Utilized: 00:03:00
CPU Efficiency: 4.92% of 01:00:56 core-walltime
Job Wall-clock time: 00:07:37
Memory Utilized: 1.86 GB
Memory Efficiency: 31.07% of 6.00 GB
```

L'ensemble des données fournies par le seff permet de donner des informations sur l'avancement du job.

Job ID : numéro/identifiant du job.
Cluster : sur quel cluster à été lancé la commande
User/Group : quel utilisateur (anemone) / quel groupe (formation) a lancé la commande
State : RUNNING / FAILED / COMPLETED, permet de montrer le statut, si le job est en train de tourner, a échoué ou s'il est terminé
nodes : job réalisé sur un noeud
cores per node : les coeurs par noeud correspondent à la puissance d'exécution d'un programme
CPU Utilized : montre l'ensemble des CPU utilisés, ici on voit que l'ensemble des CPU n'ont pas été utilisé.
CPU efficiency : pourcentage d'utilisation des demandes totales de CPU par l'application, ici uniquement 4,92%

Job Wall-clock time : Informe sur le temps que met le job à s'effectuer. Vu que le state est "completed" on peut dire que le job a mis 7 minutes et 37 secondes à s'effectuer
Memory utilized : montre l'ensemble de la mémoire utilisée par le job, ici 1.86GB sur 6 GB
Memory efficiency : mesure de l'utilisation de la mémoire comparer à ses demandes de mémoire, soit ici : 31% du total de la mémoire demandée (6 x 0.31 = 1.86)

En plus des informations fournies par la commande seff, si le job échoue, on a des fichiers tels que *error.out* et *slurm-numerojob.out* qui nous donne des informations sur l'avancement du job. error.out montre où se trouve l'erreur/les erreurs s'il y en a. Le fichier slurm montre l'ensemble de l'avancement du job et s'il y a une erreur, on voit où s'est stoppé le job et pourquoi (précision de l'erreur en question).

Chaque job a un numéro. Et lorsque le job échoue, on ouvre la sortie pour voir ce qu'il y a, on corrige et ensuite on réessaye de relancer un job avec la commande sbatch suivi du nom du fichier *lancement_nextflow.sh*.
L'option *resume* va permettre de toujours travailler sur le même job, donc en conservant le numéro de job. Pour cela il faut taper la commande suivant :

> sbatch .sh /resume

et cela permet de reprendre l'ancien job au lieu d'en créer un nouveau. Cependant j'ai oublié d'utiliser cette option lorsque j'ai fait le tp. J'ai donc un ensemble de job lancé qu'on peut retrouver dans le dossier pipeline_info (décrit dans l'exercice 3)

Exercice 3 : visualisation et interprétation des résultats

Une fois que l'état du job est "completed", cela signifie qu'il est fini. On a alors un ensemble de fichier de sortie se trouvant dans le sous-répertoire "results" qui lui même est divisé en 6 sous-répertoires :

```
anemone@genologin2 /work/anemone/projet_nextflow $ cd results/
anemone@genologin2 /work/anemone/projet_nextflow/results $ ls
fastqc  genome  multiqc  pipeline_info  star_salmon  trimgalore
anemone@genologin2 /work/anemone/projet_nextflow/results $
```

Fastqc : Résultats de qualité de chaque réplicats de chaque samples sous deux formats : en *.html* ou en *.zip*

```
anemone@genologin2 /work/anemone/projet_nextflow/results $ ls -l fastqc
total 4208
-rw-r--r-- 1 anemone formation 658458 Sep 30 11:01 1_1_fastqc.html
-rw-r--r-- 1 anemone formation 416527 Sep 30 11:01 1_1_fastqc.zip
-rw-r--r-- 1 anemone formation 654774 Sep 30 11:01 1_2_fastqc.html
-rw-r--r-- 1 anemone formation 412537 Sep 30 11:01 1_2_fastqc.zip
-rw-r--r-- 1 anemone formation 658630 Sep 30 11:01 2_1_fastqc.html
-rw-r--r-- 1 anemone formation 415677 Sep 30 11:01 2_1_fastqc.zip
-rw-r--r-- 1 anemone formation 654982 Sep 30 11:01 2_2_fastqc.html
-rw-r--r-- 1 anemone formation 412866 Sep 30 11:01 2_2_fastqc.zip
```

Genome : ensemble de fichiers et d'informations sur le génome de référence.

```
anemone@genologin2 /work/anemone/projet_nextflow/results $ ls -l genome
total 2314
drwxr-xr-x 3 anemone formation    4096 Sep 30 11:01 index
-rw-r--r-- 1 anemone formation  320910 Sep 30 11:00 ITAG2.3_genomic_Ch6.bed
-rw-r--r-- 1 anemone formation      29 Sep 30 11:00 ITAG2.3_genomic_Ch6.fasta.fai
-rw-r--r-- 1 anemone formation      20 Sep 30 11:00 ITAG2.3_genomic_Ch6.fasta.sizes
-rw-r--r-- 1 anemone formation 2034585 Sep 30 11:00 ITAG2.3_genomic_Ch6_genes.gtf
drwxr-xr-x 2 anemone formation    4096 Sep 30 11:00 rsem
anemone@genologin2 /work/anemone/projet_nextflow/results $ ls -l genome/index/star
total 513035
-rw-r--r-- 1 anemone formation         9 Sep 30 11:01 chrLength.txt
-rw-r--r-- 1 anemone formation        20 Sep 30 11:01 chrNameLength.txt
-rw-r--r-- 1 anemone formation        11 Sep 30 11:01 chrName.txt
-rw-r--r-- 1 anemone formation        11 Sep 30 11:01 chrStart.txt
-rw-r--r-- 1 anemone formation    396006 Sep 30 11:01 exonGeTrInfo.tab
-rw-r--r-- 1 anemone formation    174928 Sep 30 11:01 exonInfo.tab
-rw-r--r-- 1 anemone formation     53452 Sep 30 11:01 geneInfo.tab
-rw-r--r-- 1 anemone formation  48320606 Sep 30 11:01 Genome
-rw-r--r-- 1 anemone formation       627 Sep 30 11:01 genomeParameters.txt
-rw-r--r-- 1 anemone formation 377398217 Sep 30 11:01 SA
-rw-r--r-- 1 anemone formation  97867203 Sep 30 11:01 SAindex
-rw-r--r-- 1 anemone formation    278238 Sep 30 11:01 sjdbInfo.txt
-rw-r--r-- 1 anemone formation    332597 Sep 30 11:01 sjdbList.fromGTF.out.tab
-rw-r--r-- 1 anemone formation    332535 Sep 30 11:01 sjdbList.out.tab
-rw-r--r-- 1 anemone formation    153751 Sep 30 11:01 transcriptInfo.tab
anemone@genologin2 /work/anemone/projet_nextflow/results $ ls -l genome/rsem/
total 60433
-rw-r--r-- 1 anemone formation       20 Sep 30 11:00 genome.chrlist
-rw-r--r-- 1 anemone formation    12963 Sep 30 11:00 genome.grp
-rw-r--r-- 1 anemone formation  3573367 Sep 30 11:00 genome.idx.fa
-rw-r--r-- 1 anemone formation  3573367 Sep 30 11:00 genome.n2g.idx.fa
-rw-r--r-- 1 anemone formation  3818268 Sep 30 11:00 genome.seq
-rw-r--r-- 1 anemone formation   676102 Sep 30 11:00 genome.ti
-rw-r--r-- 1 anemone formation  3573367 Sep 30 11:00 genome.transcripts.fa
-rw-r--r-- 1 anemone formation 46617169 Sep 30 11:00 ITAG2.3_genomic_Ch6.fasta
```

Star_salmon : alignement rapide du génome sensible à l'épissage et quantification du transcriptome

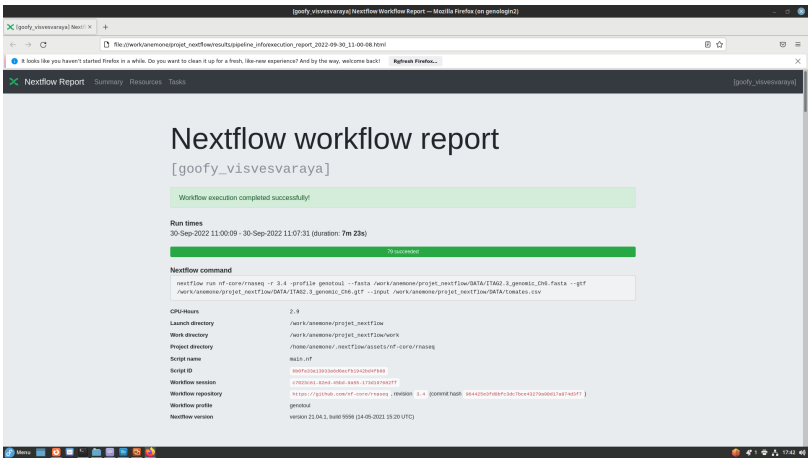
```
anemone@genologin2 /work/anemone/projet_nextflow/results $ ls -l star_salmon/
total 399047
drwxr-xr-x 5 anemone formation      4096 Sep 30 11:04 1
-rw-r--r-- 1 anemone formation 222561812 Sep 30 11:05 1.markdup.sorted.bam
-rw-r--r-- 1 anemone formation     92768 Sep 30 11:05 1.markdup.sorted.bam.bai
drwxr-xr-x 5 anemone formation      4096 Sep 30 11:03 2
-rw-r--r-- 1 anemone formation 184194680 Sep 30 11:04 2.markdup.sorted.bam
-rw-r--r-- 1 anemone formation     83016 Sep 30 11:04 2.markdup.sorted.bam.bai
drwxr-xr-x 2 anemone formation      4096 Sep 30 11:06 bigwig
drwxr-xr-x 3 anemone formation      4096 Sep 30 11:05 deseq2_qc
drwxr-xr-x 7 anemone formation      4096 Sep 30 11:05 dupradar
drwxr-xr-x 2 anemone formation      4096 Sep 30 11:05 featurecounts
drwxr-xr-x 2 anemone formation      4096 Sep 30 11:03 log
drwxr-xr-x 2 anemone formation      4096 Sep 30 11:05 picard_metrics
drwxr-xr-x 3 anemone formation      4096 Sep 30 11:04 preseq
drwxr-xr-x 4 anemone formation      4096 Sep 30 11:07 qualimap
drwxr-xr-x 9 anemone formation      4096 Sep 30 11:05 rseqc
-rw-r--r-- 1 anemone formation    111743 Sep 30 11:05 salmon.merged.gene_counts_length_scaled.rds
-rw-r--r-- 1 anemone formation    182286 Sep 30 11:04 salmon.merged.gene_counts_length_scaled.tsv
-rw-r--r-- 1 anemone formation     92135 Sep 30 11:05 salmon.merged.gene_counts.rds
-rw-r--r-- 1 anemone formation    111816 Sep 30 11:05 salmon.merged.gene_counts_scaled.rds
-rw-r--r-- 1 anemone formation    182270 Sep 30 11:04 salmon.merged.gene_counts_scaled.tsv
-rw-r--r-- 1 anemone formation    125608 Sep 30 11:04 salmon.merged.gene_counts.tsv
-rw-r--r-- 1 anemone formation    154251 Sep 30 11:04 salmon.merged.gene_tpm.tsv
-rw-r--r-- 1 anemone formation    109074 Sep 30 11:05 salmon.merged.transcript_counts.rds
-rw-r--r-- 1 anemone formation    179048 Sep 30 11:04 salmon.merged.transcript_counts.tsv
-rw-r--r-- 1 anemone formation    207691 Sep 30 11:04 salmon.merged.transcript_tpm.tsv
-rw-r--r-- 1 anemone formation    160341 Sep 30 11:04 salmon_tx2gene.tsv
drwxr-xr-x 2 anemone formation      4096 Sep 30 11:06 samtools_stats
drwxr-xr-x 4 anemone formation      4096 Sep 30 11:06 stringtie
```

STAR est l'algorithme d'alignement et Salmon permet la quantification.

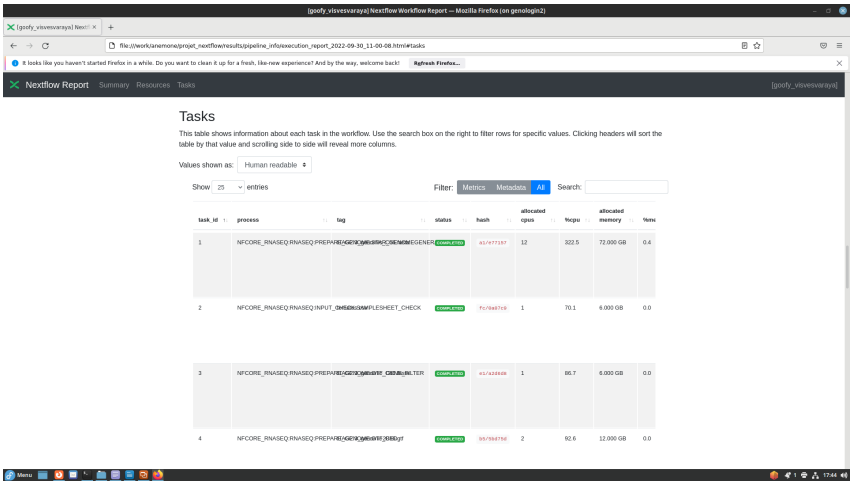
Pipeline_info : fournit des informations sur l'ensemble des jobs lancés, conserve une trace de l'exécution d'un job.

```
anemone@genologin2 /work/anemone/projet_nextflow/results/pipeline_info $ ls
execution_report_2022-09-30_10-35-02.html  execution_report_2022-09-30_11-00-08.html    execution_timeline_2022-09-30_10-55-18.html  execution_trace_2022-09-30_10-53-22.txt  pipeline_dag_2022-09-30_10-47-36.svg  software_versions.yml
execution_report_2022-09-30_10-38-48.html  execution_timeline_2022-09-30_10-35-02.html  execution_timeline_2022-09-30_11-00-08.html  execution_trace_2022-09-30_10-55-18.txt  pipeline_dag_2022-09-30_10-53-22.svg
execution_report_2022-09-30_10-47-36.html  execution_timeline_2022-09-30_10-38-48.html  execution_trace_2022-09-30_10-35-02.txt      execution_trace_2022-09-30_11-00-08.txt  pipeline_dag_2022-09-30_10-55-18.svg
execution_report_2022-09-30_10-53-22.html  execution_timeline_2022-09-30_10-47-36.html  execution_trace_2022-09-30_10-38-48.txt      pipeline_dag_2022-09-30_10-35-02.svg     pipeline_dag_2022-09-30_11-00-08.svg
execution_report_2022-09-30_10-55-18.html  execution_timeline_2022-09-30_10-53-22.html  execution_trace_2022-09-30_10-47-36.txt      pipeline_dag_2022-09-30_10-38-48.svg     samplesheet.valid.csv
```

lorsque l'on ouvre le fichier *.html* du job qui nous intéresse on tombe sur une page qui fournit un ensemble d'informations qui concerne le job lancé :



Puis on a différents graphes qui informent sur les CPU, la mémoire, la durée du job et d'I/O (input/output).
Enfin, on a un petit descriptif de chaque tâche réalisée (il y a 79 tâches au total).



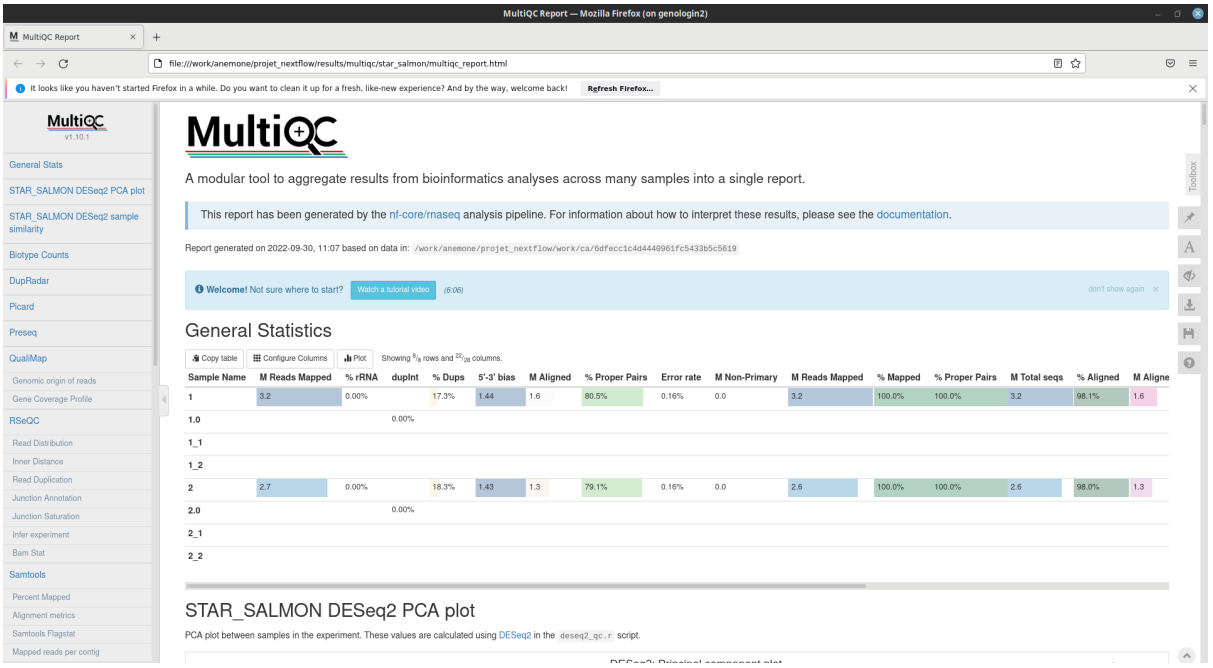
Trimgalore : fournit des informations sur les séquences (les report.txt des 1_1, 1_2, 2_1 et 2_2)

```
anemone@genologin2 /work/anemone/projet_nextflow/results $ ls -l trimgalore/
total 3
-rw-r--r-- 1 anemone formation 3227 Sep 30 11:02 1_1.fastq.gz_trimming_report.txt
-rw-r--r-- 1 anemone formation 3445 Sep 30 11:02 1_2.fastq.gz_trimming_report.txt
-rw-r--r-- 1 anemone formation 3095 Sep 30 11:01 2_1.fastq.gz_trimming_report.txt
-rw-r--r-- 1 anemone formation 3336 Sep 30 11:01 2_2.fastq.gz_trimming_report.txt
drwxr-xr-x 2 anemone formation 4096 Sep 30 11:02 fastqc
anemone@genologin2 /work/anemone/projet_nextflow/results $ ls -l trimgalore/fastqc/
total 4288
-rw-r--r-- 1 anemone formation 676255 Sep 30 11:02 1_1_val_1_fastqc.html
-rw-r--r-- 1 anemone formation 413216 Sep 30 11:02 1_1_val_1_fastqc.zip
-rw-r--r-- 1 anemone formation 676183 Sep 30 11:02 1_2_val_2_fastqc.html
-rw-r--r-- 1 anemone formation 409844 Sep 30 11:02 1_2_val_2_fastqc.zip
-rw-r--r-- 1 anemone formation 678895 Sep 30 11:01 2_1_val_1_fastqc.html
-rw-r--r-- 1 anemone formation 415124 Sep 30 11:01 2_1_val_1_fastqc.zip
-rw-r--r-- 1 anemone formation 675555 Sep 30 11:01 2_2_val_2_fastqc.html
-rw-r--r-- 1 anemone formation 412361 Sep 30 11:01 2_2_val_2_fastqc.zip
```

Multiqc : fournit les résultats obtenus avec Multiqc

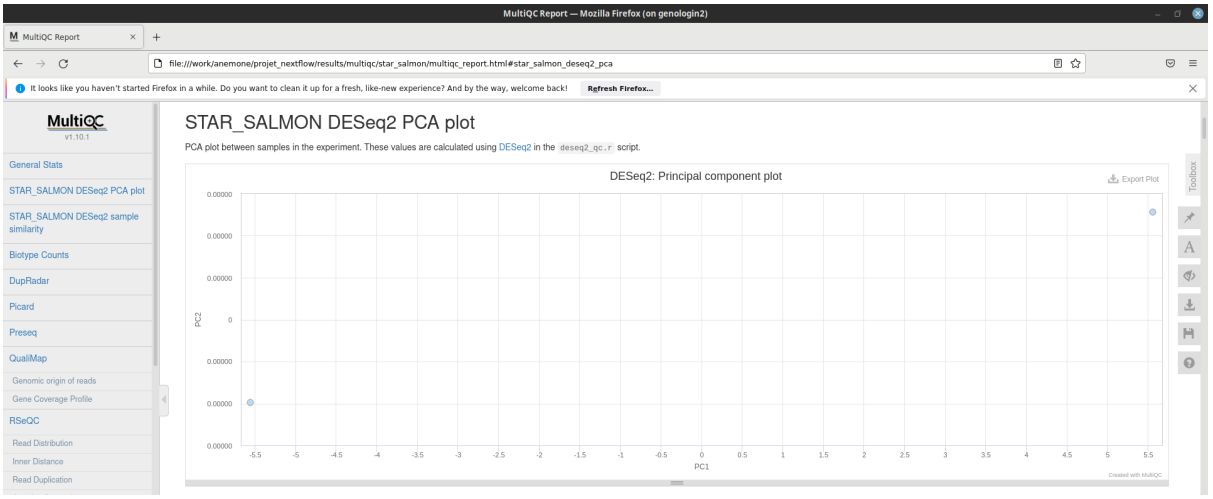
```
anemone@genologin2 /work/anemone/projet_nextflow/results $ ls -l multiqc/star_salmon/
total 1377
drwxr-xr-x 2 anemone formation    4096 Sep 30 11:07 multiqc_data
-rw-r--r-- 1 anemone formation 1401378 Sep 30 11:07 multiqc_report.html
anemone@genologin2 /work/anemone/projet_nextflow/results $ ls -l multiqc/star_salmon/multiqc_data/
total 991
-rw-r--r-- 1 anemone formation    394 Sep 30 11:07 multiqc_cutadapt.txt
-rw-r--r-- 1 anemone formation 970336 Sep 30 11:07 multiqc_data.json
-rw-r--r-- 1 anemone formation   1152 Sep 30 11:07 multiqc_fastqc_1.txt
-rw-r--r-- 1 anemone formation   1089 Sep 30 11:07 multiqc_fastqc.txt
-rw-r--r-- 1 anemone formation   2668 Sep 30 11:07 multiqc_general_stats.txt
-rw-r--r-- 1 anemone formation  20484 Sep 30 11:07 multiqc.log
-rw-r--r-- 1 anemone formation    395 Sep 30 11:07 multiqc_picard_dups.txt
-rw-r--r-- 1 anemone formation    600 Sep 30 11:07 multiqc_rseqc_bam_stat.txt
-rw-r--r-- 1 anemone formation     82 Sep 30 11:07 multiqc_rseqc_infer_experiment.txt
-rw-r--r-- 1 anemone formation    710 Sep 30 11:07 multiqc_rseqc_junction_annotation.txt
-rw-r--r-- 1 anemone formation   1714 Sep 30 11:07 multiqc_rseqc_read_distribution.txt
-rw-r--r-- 1 anemone formation   1066 Sep 30 11:07 multiqc_samtools_flagstat.txt
-rw-r--r-- 1 anemone formation     78 Sep 30 11:07 multiqc_samtools_idxstats.txt
-rw-r--r-- 1 anemone formation   1774 Sep 30 11:07 multiqc_samtools_stats.txt
-rw-r--r-- 1 anemone formation   5778 Sep 30 11:07 multiqc_sources.txt
-rw-r--r-- 1 anemone formation    814 Sep 30 11:07 multiqc_star.txt
```

On a un ensemble de fichiers *.txt* qui résume les différentes analyses faites par multiqc.
En effet, multiqc est un outil de création d'un rapport d'un ensemble d'analyses faites sur des données. Le QC est pour "quality control".
Lorsque l'on ouvre le fichier *multiqc_report.html*, on tombe sur la page suivante :

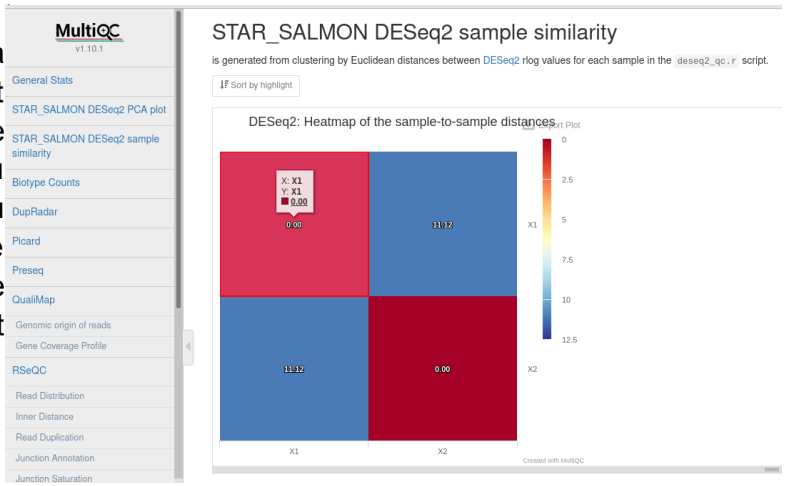


Tout d'abord, en en-tête de la page, on a la définition de l'outil MultiQC suivi de la date et de l'heure du report ainsi que le directory à partir duquel les analyses ont été faites.
A gauche, on voit un ensemble de vignettes à droite de la fenêtre qui permet de se déplacer dans la page en fonction des différentes analyses bioinformatique réalisées.

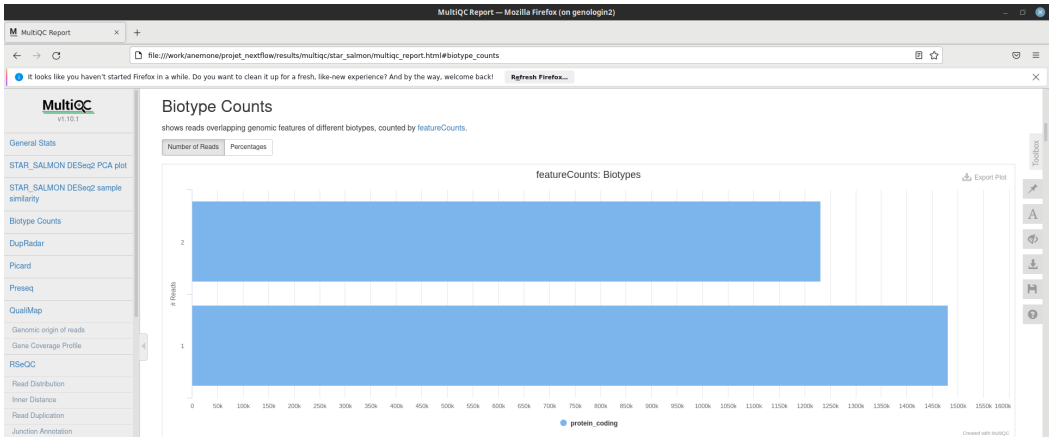
La première figure qu'on a c'est un tableau de statistiques de l'ensemble des modules utilisés lors des analyses. Cela permet de voir plus spécifiquement la différence de résultats entre les échantillons.
Ci-dessous est représenté un plot d'ACP entre les deux samples à l'aide des outils STAR et Salmon. On voit bien que les 2 points sont clairement bien séparés. Cela montre la dissimilitude entre les deux types d'échantillons. En effet les deux échantillons ne semblent pas identiques.



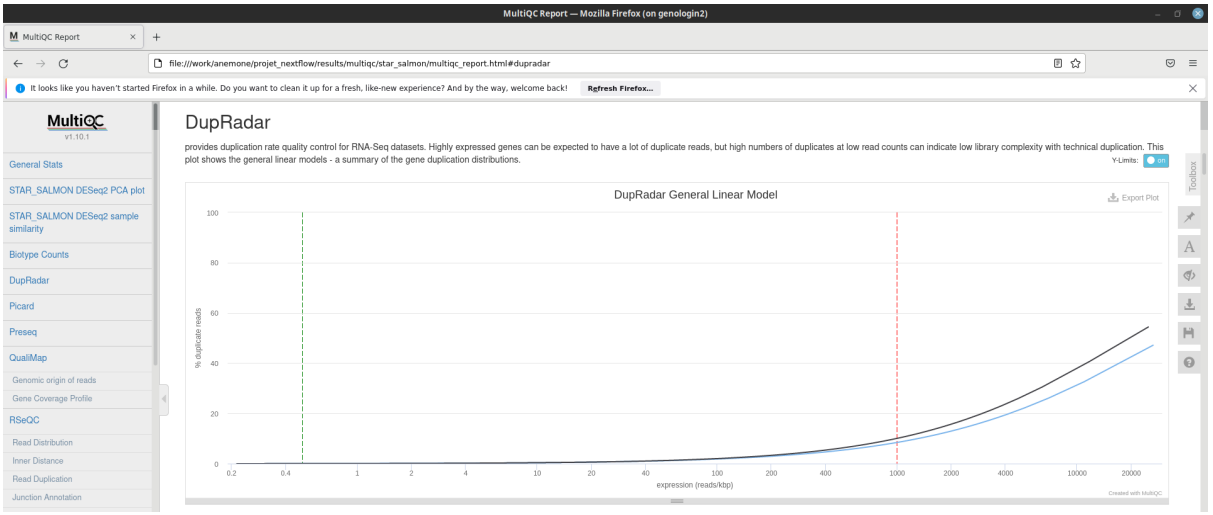
Cette heatmap, également fournie à l'aide des outils STAR et Salmon, permet de voir qu'il n'y a aucune distance (aucune différence) entre les réplicats du sample 1 ainsi qu'entre les réplicats du sample 2. Cependant on voit une distance de 11,12 entre le sample 1 et le sample 2 ce qui montre qu'ils sont différents.



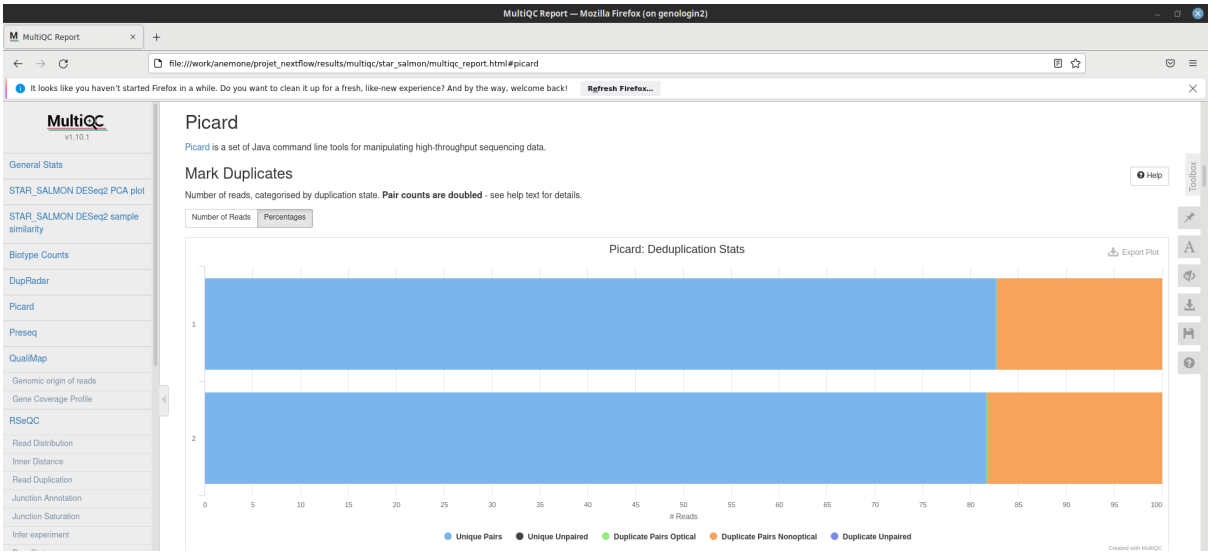
On retrouve ci-dessous le nombre de featuresCounts de chaque sample obtenu à l'aide de l'outil Biotype Counts. On remarque qu'il y a un nombre plus important de reads pour le sample 1 que le sample 2. Le nombre de reads peut parfois jouer sur les résultats de certaines analyses.



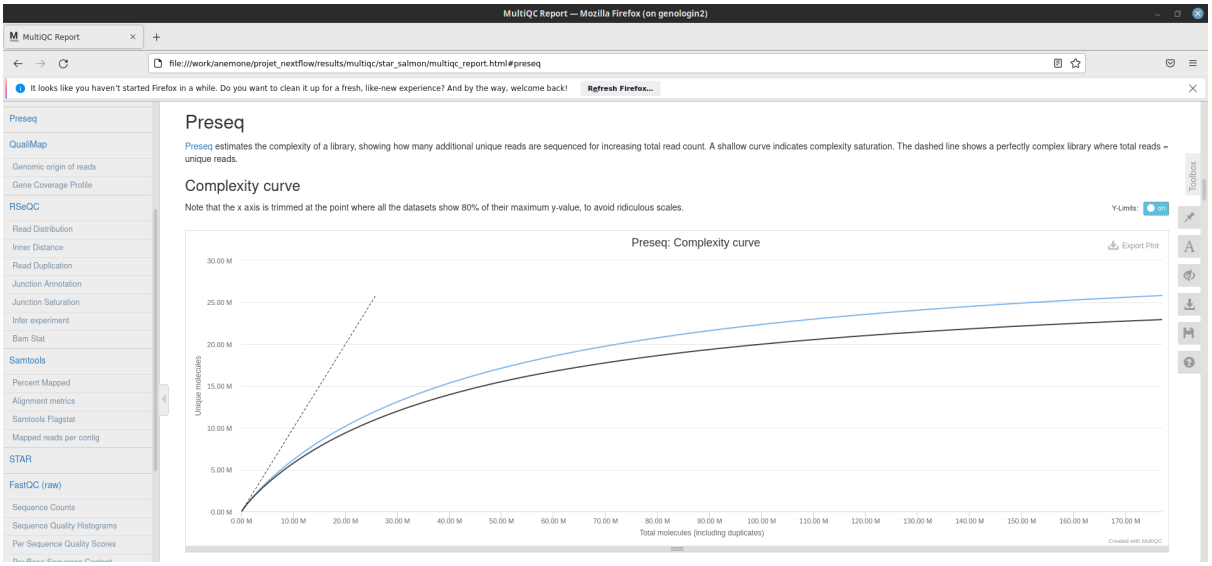
L’outil DupRadar permet de voir le pourcentage de duplicat de read. On remarque que les deux courbes des deux samples varient de la même façon avec une variation un peu plus importante pour le sample 2



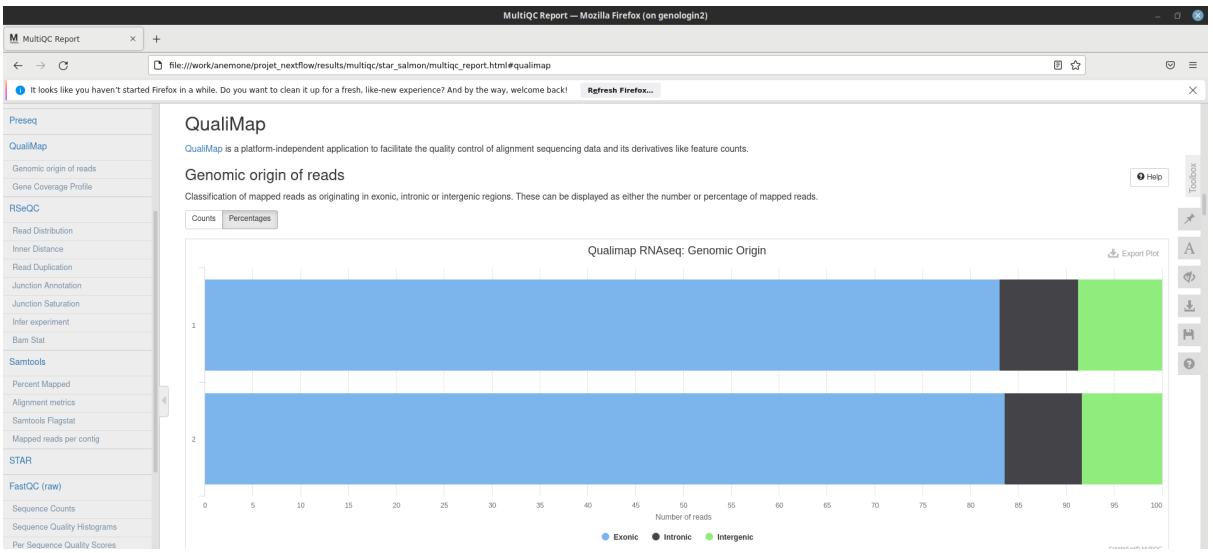
L’outil Picard permet de donner des informations sur le nombre de reads catégorisé par les états de duplication. Cela permet d'indiquer si le contenu est dupliqué ou non puisque ça arrive qu'il y est duplication lors de la préparation des librairies. On voit qu’il y a essentiellement des pairs uniques et des duplicate pairs nonoptical.



Preseq est un outil qui permet d'observer la courbe décrivant la complexité d'une librairie. La ligne en pointillé représente une bibliothèque parfaitement complexe où le nombre total de reads = nombre de reads uniques. La courbe bleue qui représente le sample 1 montre une plus grande complexité dû à son nombre de reads unique plus important.

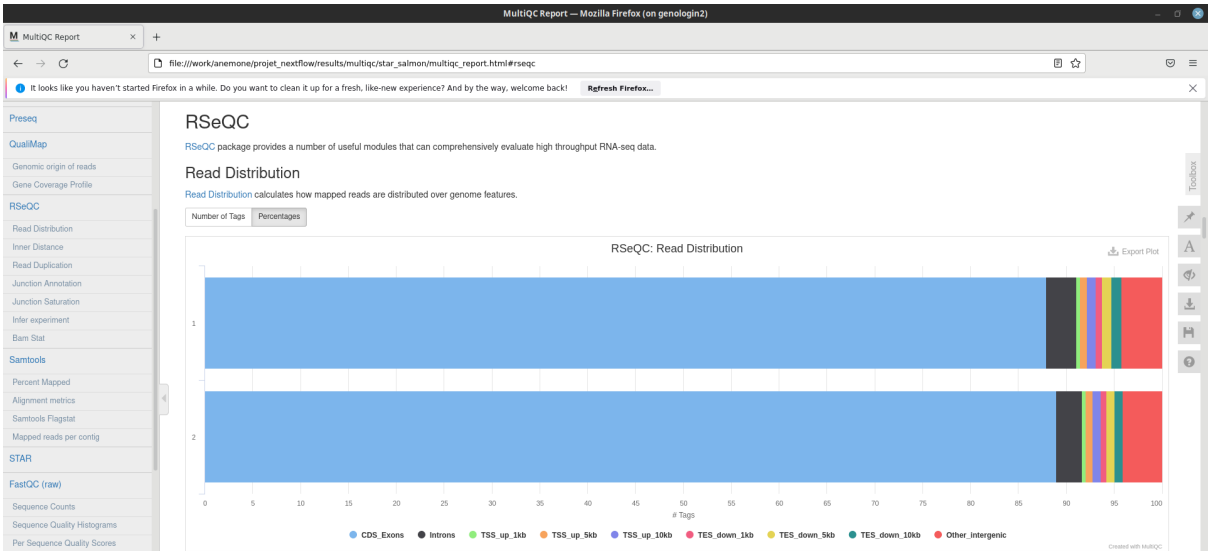


Le module QualiMap facilite le contrôle de la qualité des données de séquençage d'alignement et de ses dérivés comme le nombre de features. On ne voit pas une différence flagrante entre les deux samples. Le nombre de reads exonique, intronique et intergénique semble équivalent dans les deux échantillons.

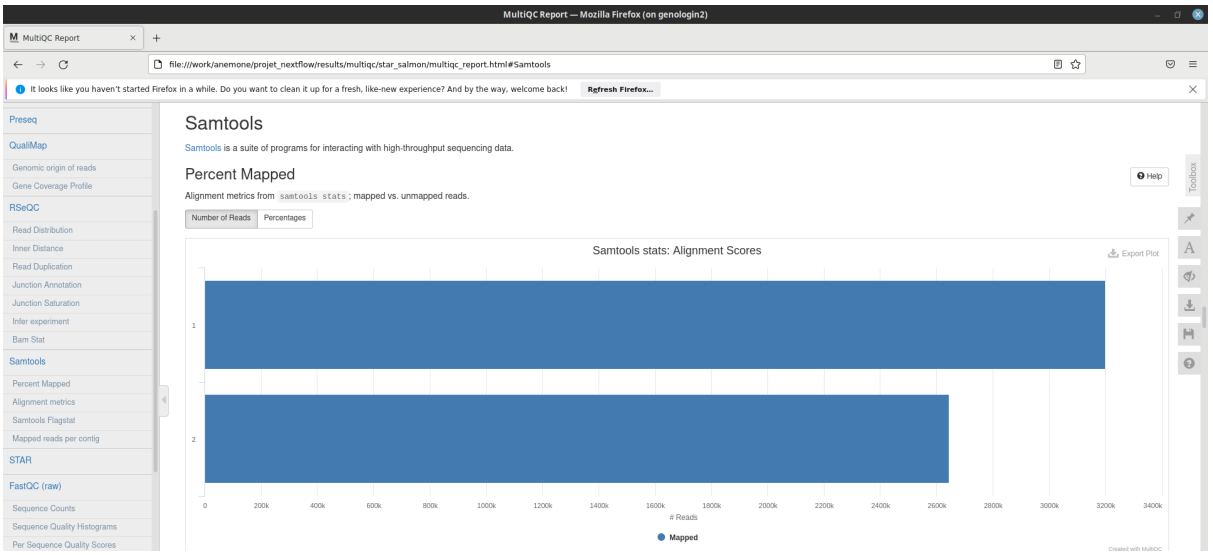


Sur la seconde figure du module QualiMap qui montre le Gene Coverage Profil, on peut voir que le sample 1 est le sample avec la plus grande couverture du génome. Cela peut également s'expliquer du fait que le sample 1 est le sample avec le plus grand nombre de reads, donc c'est le plus susceptible de couvrir le plus le génome.

RNA-Seq Quality Control est un outil qui permet de voir la répartition des différentes zones de l'ARN telles que : les régions codantes, régions intronique, les TSS, les TSE à 1kb, 5kb et 10kb, up et down et les autres régions intergénique. On voit que la distribution entre les deux semble assez similaire avec légèrement plus de régions codantes pour le sample 2.

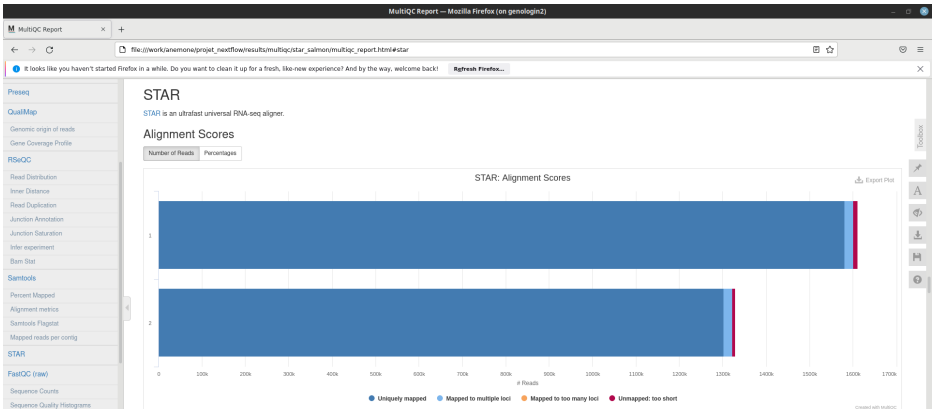


Samtools est un outil qui permet d'aligner les reads sur le génome. On voit qu'il y a moins de reads mapped pour le sample 2 ce qui s'explique d'un nombre inférieur de reads pour cet échantillon.



Quand on appuie sur "pourcentage" à côté du nombre de reads, on a 100% des reads qui sont mappés, ce qui confirme que le fait qu'on est moins de reads mapped dû à un nombre de reads plus faible pour le sample 2 .

STAR qui est aligneur de reads sur le génome de référence, montre de façon plus précise comment sont alignés les reads et si certains ne le sont pas, dû à leur taille trop faible.



Les résultats sont assez similaires pour les deux échantillons au vu du nombre de reads chacun. Une grande partie des reads sont alignés sur le génome.

FastQC rend un analyse pour chaque échantillon alors que MultiQC rend une analyse pour l'ensemble des échantillons en même temps. MultiQC permet donc de simplifier la comparaison des samples les uns aux autres. C'est pour cela qu'on va préférer comparer les résultats FastQC obtenu par le report de MultiQC plutôt que les fichiers *.html* obtenus dans le dossier FastQC de résultat.

On a deux types de résultats FastQC : (FASTQC row vs trimmed)
le FastQC avant le traitement de cutadapt (row) et le FastQC après le traitement par cutadapt (trimmed). Cependant l'ensemble des résultats paraissent très similaires. Il est quand même notable que la qualité des résultats est meilleure pour le FastQC une fois trimmed.

Sequence Counts : parmi les unique reads et les duplicate reads, on compte moins de duplicate reads lorsque les reads sont trimmed.

Sequence Quality Histograms : La moyenne du quality score pour 100 pb est d'environ 35 (score Phred) pour les reads trimmed contre environ 30 pour les reads row.

Per Sequences Quality Scores : On voit légèrement plus de reads à une valeur de score Phred de 38 (pic dans les deux conditions) lorsque les reads sont trimmed.

Per Base Sequence Content : La proportion de chaque base est très proche entre les reads trimmed et les reads row. La répartition dépend de la région où on se trouve sur la séquence.

Per Sequence GC Content : La proportion de chaque GC est très proche entre les reads trimmed et les reads row. On voit un pic vers 45% et une distribution entre 25 et 65%.

Per Base N Content : La proportion de bases N (base non défini comme A, T, G ou C) est également similaire entre les deux résultats row/trimmed. Il y a peu de bases qui sont non définies.

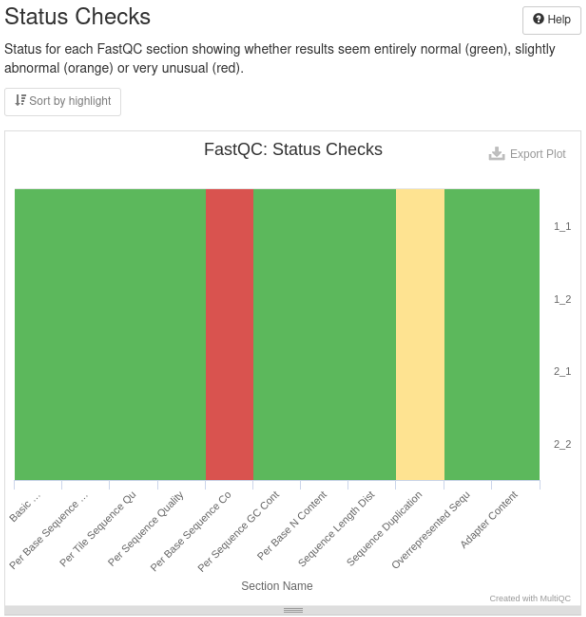
Sequence Length Distribution : La longueur de distribution des reads qui contiennent encore les adaptateurs, primers etc, est de 101pb pour chaque read. Cependant, pour les reads trimmed, on va avoir des longueurs variables, autour de 95-100 pb mais qui varient contrairement aux reads non trimmed. Cela peut s'expliquer par l'absence d'adaptateurs de primers etc après l'étape de cutadapt.

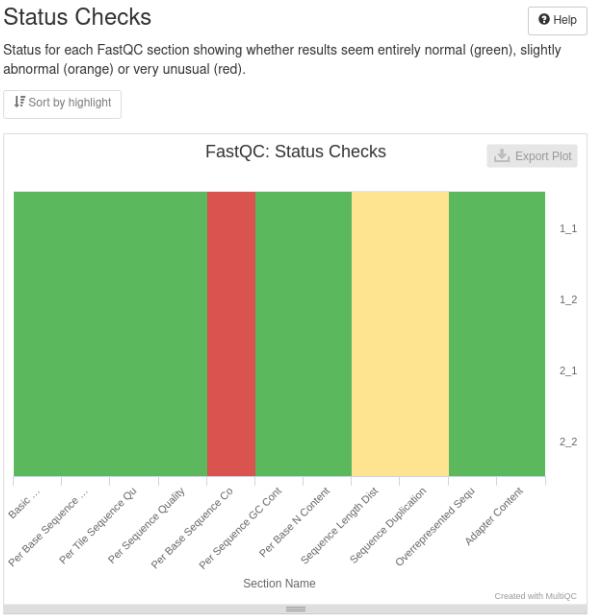
Sequence Duplication Levels : Le niveau de duplication de séquences est assez similaire entre les reads trimmed et non trimmed. On voit qu'on a un pic de 1 à 2 levels pour 20 à 40% de la librairie. Puis on va avoir un second pic de 9 à 50 levels pour moins de 20% de la librairie. Et une légère courbe en diminution de 50 à 500 levels correspondant à environ 5% de la librairie. On a donc un grand nombre de reads à faible duplication et un nombre faible de reads avec un nombre plus important de duplication.

Overrepresented sequences : On retrouve le même résultat dans les deux cas de FastQC. Les 4 échantillons ont moins de 1% de reads composant des séquences surreprésentées. Cela montre une bonne qualité des reads.

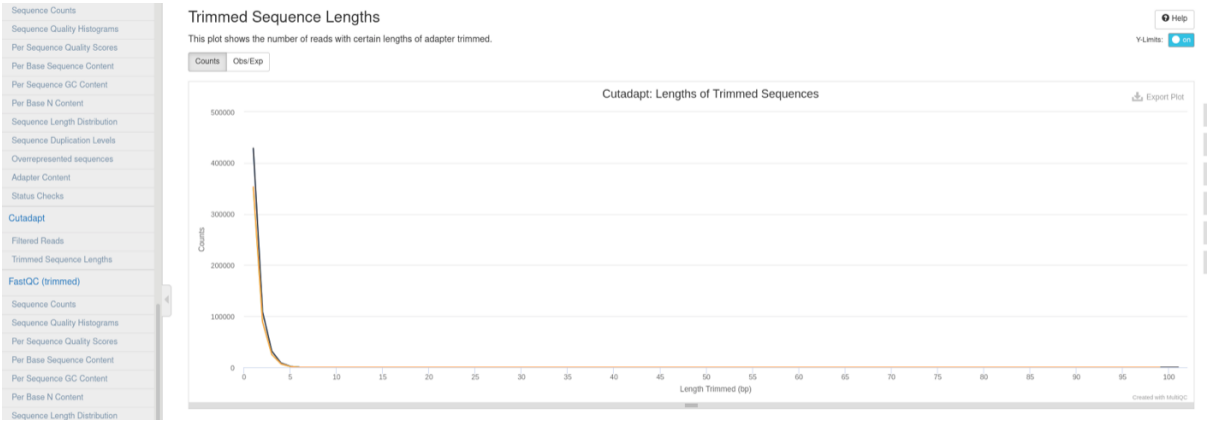
Adapter Content : On retrouve le même résultat dans les deux cas de FastQC. Aucun échantillon n'a de contamination > 0.1% dû à des adaptateurs. Cela montre une bonne qualité des reads.

Status Checks : montre le statut des différentes sections des FastQC. Lorsque c'est vert c'est que les données sont bonnes et utilisables. Lorsque c'est orange c'est que les données sont moyennement fiables et utilisables. Enfin, lorsque c'est rouge, les données ne sont pas bonnes et sont difficilement utilisables. A gauche, on trouve le status check du FastQC row et à droite le status check du FastQC trimmed. On voit que pour les reads trimmed, le module de la distribution de la longueur des reads n'est pas optimal ou précis. La variabilité de taille dûe au fait qu'on retire les séquences non génomique rend plus difficile la justesse du calcul de la taille de l'ensemble des reads.

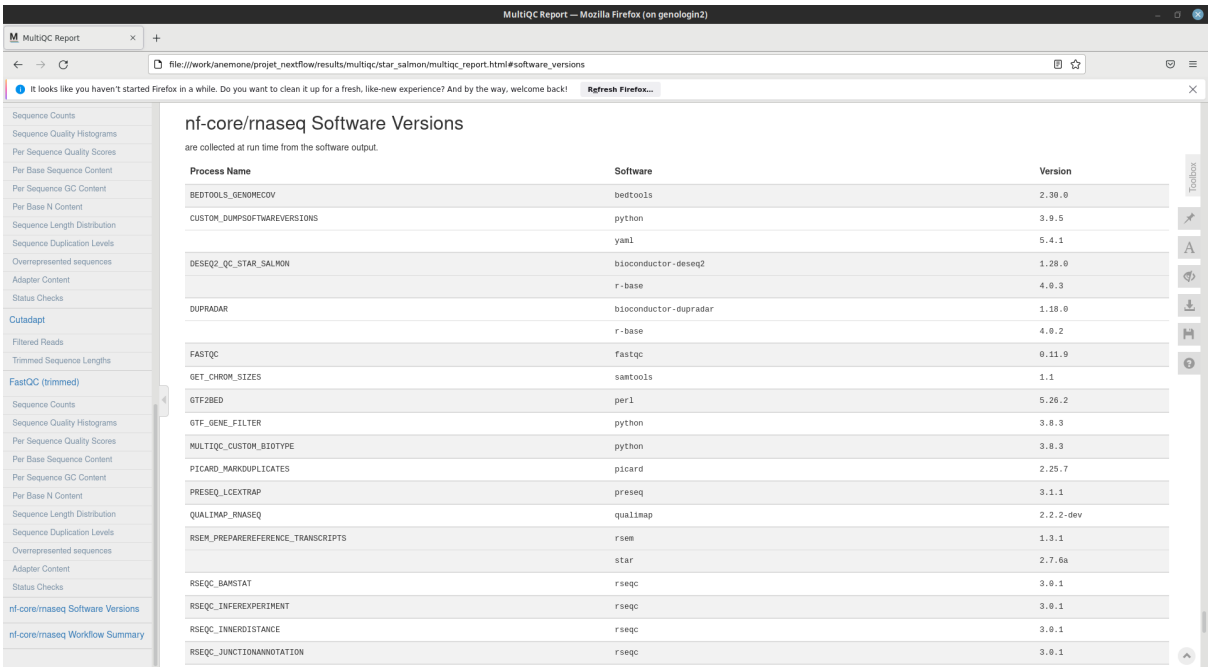




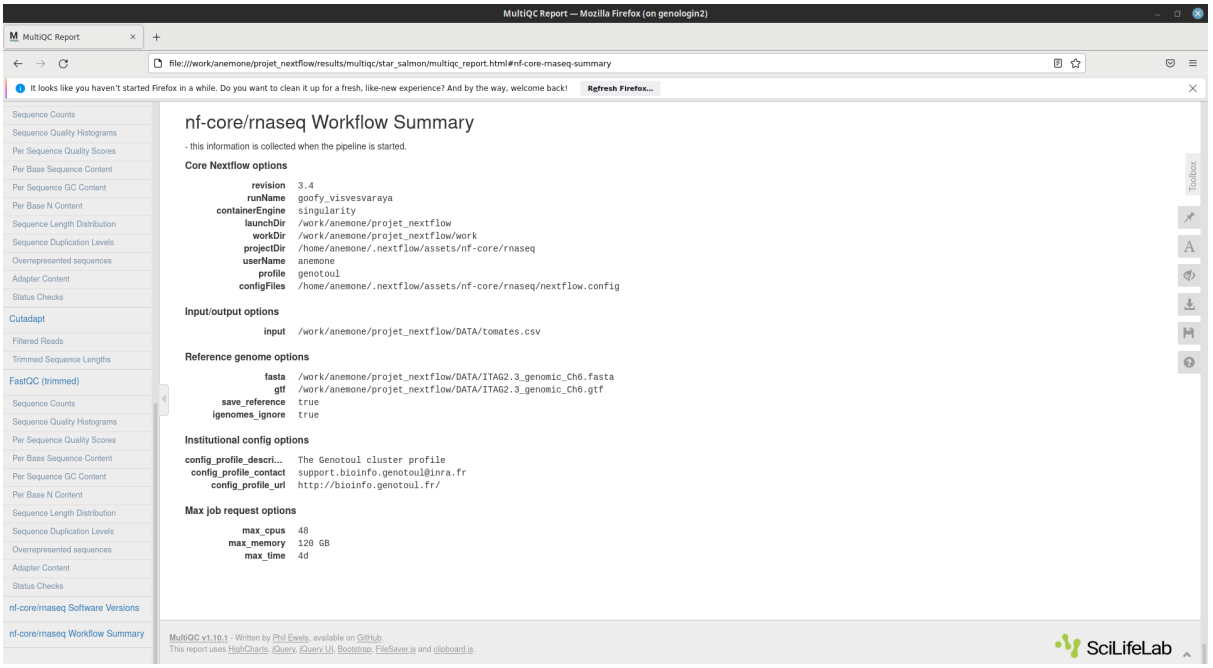
L’outil cutadapt permet de trouver et d’enlever l’ensemble des éléments non génique de nos séquences en sortie de séquenceur comme : les adaptateurs, les primers, les queue polyA etc. Afin d’avoir uniquement des reads (séquence ADN). Autrement dit, de filtrer les reads.
On remarque que de nombre reads sont trimmés sur une longueur d’environ 5pb, ce qui expliquerait la variation des 5pb lorsque l’on regarde la taille des reads trimmés avec FastQC.



Enfin, dans la partie nfcore/rnaseq, on retrouve les informations sur le summary du workflow et sur les versions des différents outils utilisés pour le report MultiQC.



Process Name	Software	Version
BEDTOOLS_GENOMECOV	bedtools	2.30.0
CUSTOM_DUMPSOFTWAREVERSIONS	python	3.9.5
	yaml	5.4.1
DESEQ2_QC_STAR_SALMON	bioconductor-deseq2	1.28.0
	r-base	4.0.3
DUPRADAR	bioconductor-dupradar	1.18.0
	r-base	4.0.2
FASTQC	fastqc	0.11.9
GET_CHROM_SIZES	samtools	1.1
GTF2BED	perl	5.26.2
GTF_GENE_FILTER	python	3.8.3
MULTIQC_CUSTOM_BIOTYPE	python	3.8.3
PICARD_MARKDUPLICATES	picard	2.25.7
PRESEQ_LCEXTRAP	preseq	3.1.1
QUALIMAP_RNASEQ	qualimap	2.2.2-dev
RSEM_PREPAREREFERENCE_TRANSCRIPTS	rsem	1.3.1
	star	2.7.6a
RSEQC_BAMSTAT	rseqc	3.0.1
RSEQC_INFEREXPERIMENT	rseqc	3.0.1
RSEQC_INNERDISTANCE	rseqc	3.0.1
RSEQC_JUNCTIONANNOTATION	rseqc	3.0.1



Conclusion :

Ce TP nous a permis de manipuler l'environnement NextFlow qui est un environnement de travail de plus en plus utilisé dans le domaine de la biologie. Il permet de manipuler des fichiers pour lancer des scripts et traiter un ensemble de données. Il existe de nombreux outils tel que nfcore/rnaseq qui permettent une analyse approfondie de données biologiques. Durant ce TP, l'outil nfcore/rnaseq a permis de comparer deux échantillons ayant 2 réplicats chacun aux génomes de référence, à l'aide d'une analyse rnaseq. Nous avons pu analyser la qualité des librairies (ensemble des reads) en sortie de séquenceurs. Cela nous a montré que lorsqu'on fait des manipulations bio comme la création de librairies, les librairies peuvent varier d'un échantillon à un autre (le nombre de reads). De plus, lorsqu'on les aligne sur les génomes, les résultats sont similaires mais diffèrent en certains points. On a également vu qu'en fonction des outils utilisés comme cutadapt, qui modifie la séquence des reads, on modifiait la taille de la séquence et donc son alignement sur le génome était réduit et cela permettait une interprétation plus juste de la localisation des reads (puisqu'on a la séquence du read directement).

J'ai trouvé ce TP très intéressant et constructif, de la connexion à un compte genotoul en passant par la création de nos fichiers .sh, du lancement de Nextflow jusqu'à l'interprétation d'une sortie d'analyse RNAseq.