HACQUIN
OCEANE
identifiant genotoul : cyclamen

COMPTE RENDU DE TP NEXTFLOW :

Initiation sur genologin

Octobre 2022

Table des matières

Table des matières 2

1- Exercice 1 : Connexion à Genologin, création d'un répertoire de travail, téléchargement de fichiers à traiter 3

2- Préparer son fichier bash de lancement Nextflow 4

a. Préparer le fichier de lancement 4

b. Suivre le job avec seff, utiliser resume si nécessaire. 5

3- Exercice 3: Interpréter le rapport MultiQC ainsi que les principaux fichiers résultats obtenus. 6

a. Généralités sur l'analyse RNAseq 6

b. Interprétation des principaux résultats: 8

1. Fastqc 8

2. Genome: 8

3. Pipeline_info: 9

4. Star_salmon 9

5. Trimgalore 11

c. Interprétation du report MultiQC 11

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

Le répertoire de travail contenant toutes les données du projet est "NEXTFLOW_projet_note" au chemin suivant :
/work/cyclamen/NEXTFLOW_projet_note

Les informations principales pour l'aide à la réalisation de ce rapport proviennent du site : https://nf-co.re/rnaseq/3.8.1/output?fbclid=IwAR0Xq-kI5-XbuiaIxRkqmOnVhJ68TVoocjsIXPLf-HGLJMrBSzu4g2hkuo4#star-and-salmon

Dans ce fichier l'architecture du dossier est la suivante :

- data/ : dossier contenant toutes les données nécessaires pour l'analyse RNAseq:
  - ITAG2.3_genomic_Ch6.fasta : génome de référence de la tomate au format fasta.
  - ITAG2.3_genomic_Ch6.gtf : fichier d'annotation du génome au format gtf.
  - MT_rep1_1_Ch6.fastq.gz : fichier fastq à la sortie du séquenceur suite à l'expérience RNAseq des cellules mutantes du run1.
  - MT_rep1_2_Ch6.fastq.gz : fichier fastq à la sortie du séquenceur suite à l'expérience RNAseq des cellules mutantes du run2.
  - WT_rep1_1_Ch6.fastq.gz : fichier fastq à la sortie du séquenceur suite à l'expérience RNAseq des cellules sauvage du run1.
  - WT_rep1_1_Ch6.fastq.gz: fichier fastq à la sortie du séquenceur suite à l'expérience RNAseq des cellules sauvage du run2.
  - input.csv : fichier csv nécessaire au lancement du job nf-core/rnaseq. Ce fichier contient les 4 fichiers fastq au format adéquat.
- results_rnaseq/ : dossier comprenant tous les résultats de sortie une fois le job exécuté.
- launch.sh : fichier bash pour le lancement de Nextflow (détaillé dans l'exercice 2).
- launch.err : fichier contenant les erreurs s'il y en a lu lors de l'exécution du job.
- slurm-37825567.out : fichier contenant toutes les exécutions effectuées lors du lancement du job nf-core/rnaseq avec les informations relatives aux exécutions (erreurs, warning, job completed, …).

2- Préparer son fichier bash de lancement Nextflow

a. Préparer le fichier de lancement

Le fichier de lancement pour le traitement RNAseq des données « tomates » se nomme : “launch.sh” :

```
#!/bin/bash
#SBATCH -J OceaneHacquin # permet de parametrer le nom du job sur le cluster
#SBATCH --mem=6G # permet de parametrer la memoire du job
#SBATCH --time=1-0  # permet de parametrer la durée maximale du job
#SBATCH -p workq # permet de lancer le job sur la workq c_est a dire placer le job dans la file d_attente pour etre pris en charge
#SBATCH -e launch.err # si il y a des erreurs elles seront contenues dans ce fichier
#SBATCH --cpus-per-task=30 #utilisation du nombre de cpu necessaire


# purge des anciens modules
module purge

# Chargement des modules
module load bioinfo/Nextflow-v21.04.1
module load system/singularity-3.5.3

# Lancement du pipeline pour effectuer l'analyse RNAseq
nextflow run nf-core/rnaseq \
-r 3.4 \
--input data/input.csv \
--fasta data/ITAG2.3_genomic_Ch6.fasta \
--gtf data/ITAG2.3_genomic_Ch6.gtf \
--outdir results_Rnaseq \
-profile genotoul \
```

Le choix des paramètres correspond à ce demandé dans l'énoncé et sont expliqués dans le fichier. Concernant la ligne de Nextflow nf-core/rnaseq, les arguments sont les suivants:

- -r : correspond à la révision du module nf-core/rnaseq utilisé.
- –input : correspond au fichier d’entrée. Celui-ci est un fichier csv contenant les fastq des différents run au format adéquat :

```
cyclamen@genologin2 /work/cyclamen/NEXTFLOW_projet_note/data $ more input.csv
sample,fastq_1,fastq_2,strandedness
1,data/MT_rep1_1_Ch6.fastq.gz,data/MT_rep1_2_Ch6.fastq.gz,unstranded
2,data/WT_rep1_1_Ch6.fastq.gz,data/WT_rep1_2_Ch6.fastq.gz,unstranded
```

- --fasta: correspond au génome de référence de la Tomate au format fasta.
- --gtf: correspond au fichier d’annotation au format gtf. Ce format de fichier est utilisé. pour décrire les gènes et d'autres éléments de séquences d'ADN, d'ARN et de protéines.
- --outdir : correspond au chemin où les résultats seront positionnés une fois le job fini.
- --profile : ce paramètre est utilisé pour choisir un profil de configuration. Les profils peuvent fournir des préréglages de configuration pour différents environnements de calcul. Ici nous avons choisi l’environnement genotoul.

Lors du premier lancement du pipeline les paramètres n'étaient pas ceux-ci dessus mais suite à différents problèmes j’ai dû adapter le fichier. En effet, premièrement lors du lancement j’ai eu des erreurs de cpu. Au départ, j’avais attribué 4 cpus mais j’ai eu une erreur m’indiquant qu’il n’y en avait pas assez. J’ai ensuite monté ce chiffre à 12 cpus et j’ai encore eu une autre erreur. J’ai donc finalement monté une dernière fois le nombre de cpu à 30.
Une fois le nombre de cpu assez conséquent une nouvelle erreur est apparue. Celle-ci notifiait que l’image singularity était inconnue. J’ai alors décidé de charger le module singulary. Après plusieurs tests de version pour que cela fonctionne, j’ai retenu la version 3.5.3. A la suite de la résolution de ce problème une dernière erreur est apparue. Les fichiers dans mon fichier csv étaient introuvables. En effet, sur la première version de mon fichier, je n’avais notifié que le nom des fichiers fastq. Or ces fichiers étant dans un dossier différent

que le fichier de lancement, celui-ci ne les trouvait pas. J'ai donc modifié mon fichier csv en ajoutant le chemin relatif pour mes fichiers fastq.
Après avoir lancé une première fois le pipeline je me suis rendue compte que le nombre de cpu était beaucoup trop grand donc je l'ai diminué à 6.
Après toutes ces modifications j'ai alors effectué la ligne de commande suivante pour le lancer le pipeline est : sbatch launch.sh.

b. Suivre le job avec seff, utiliser resume si nécessaire.

```
cyclamen@genologin2 /work/cyclamen/NEXTFLOW_projet_note $ seff 37825567
Job ID: 37825567
Cluster: genobull
User/Group: cyclamen/formation
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 6
CPU Utilized: 00:02:49
CPU Efficiency: 4.54% of 01:02:00 core-walltime
Job Wall-clock time: 00:10:20
Memory Utilized: 1.46 GB
Memory Efficiency: 24.38% of 6.00 GB
```

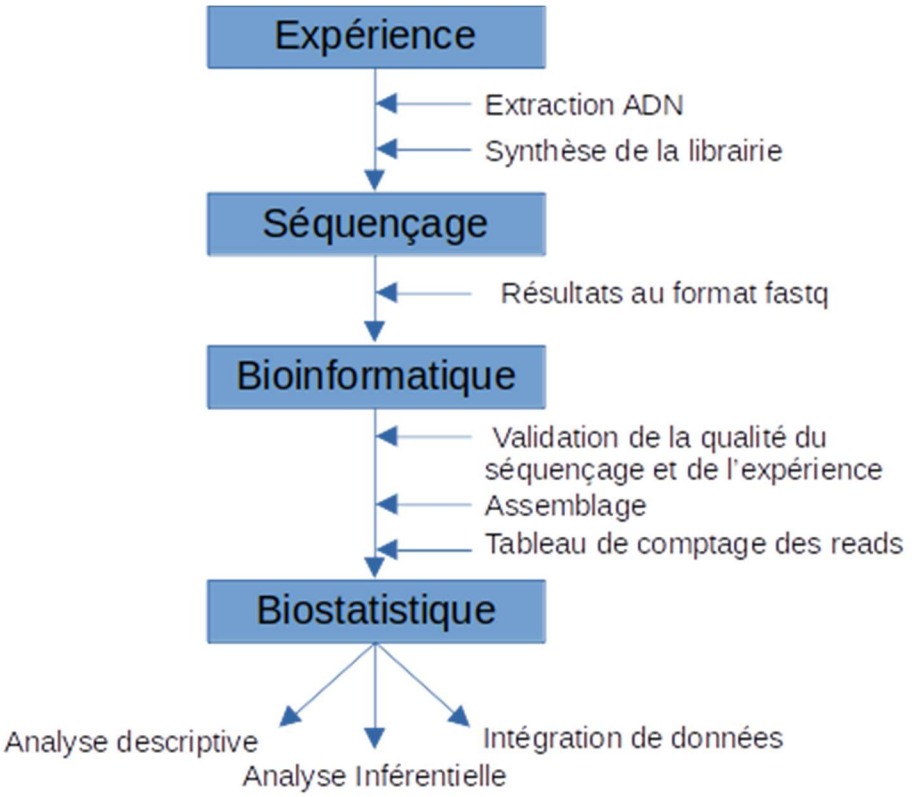
La commande seff récupère un identifiant de job et affiche l'efficacité de l'utilisation du processeur et de la mémoire de ce travail. C'est-à- dire les ressources ayant réellement été utilisées pour exécuter le travail, par rapport à la réservation des ressources initiales.
Cette commande peut donc aider à définir une réservation adéquate de ressources.
Cet affichage se décompose en 11 lignes :

- Job ID : correspond à l'identifiant de notre travail. Celui-ci est donné lors du lancement du job sur le cluster.
- Cluster : le cluster possède plusieurs nœuds, le nœud genobull est ici utilisé (c'est un admin node ayant 20 cores et 128 GB de RAM).
- State : correspond au statut du job. Il y a 3 possibilités :
    - RUNNING : en cours.
    - COMPLETED : fini.
    - FAILED: le job est arrêté car il y a une erreur.
- Nodes: indique combien de nœuds ont été utilisé pour effectuer le job.
- Cores per nodes: correspond au nombre de processus utilisés par nœuds. Ici c'est un paramètre que nous avons défini dans le fichier de lancement.
- CPU Utilized: correspond au temps réel utilisé par le job sur tous les cpus.
- CPU Efficiency : correspond à l'efficacité du cpu. Celle-ci est calculée comme le ratio du temps réel d'utilisation de tous les cœurs divisé par le nombre de cœurs. demandés divisé par le temps d'exécution. Ici, nous voyons que l'efficacité du CPU est de 4.54%, ce qui signifie que la tâche a utilisé les 6 cœurs à équivalent de 4.54% du maximum possible pendant la durée d'exécution. On voit bien donc ici que le nombre de cpu était encore beaucoup trop élevé pour le job.
- Job Wall-cock time : correspond à la durée s'étant écoulée depuis le début du job
- Memory Utilized: correspond à la mémoire réellement utilisée pour le travail.
- Memory Efficiency: correspond à l'efficacité de la mémoire. Elle est calculée comme le ratio de la mémoire la plus élevée utilisée par toutes les tâches divisée par la mémoire demandée pour le travail. La demande totale de mémoire pour cette tâche était de 6GB et seulement 1,46GB ont été utilisés. L'efficacité de la mémoire est donc de 24,38 %. On en déduit donc que l'on aura pu grandement réduire notre demande de mémoire dans le fichier de lancement.

3- Exercice 3: Interpréter le rapport MultiQC ainsi que les principaux fichiers résultats obtenus.

a. Généralités sur l'analyse RNAseq

La méthode RNAseq est une technique permettant l'étude du transcriptome d'une population de cellule, c'est -à -dire l'évaluation et la quantification de l'ensemble des transcrits à un instant T et dans un environnement précis. Le design d'une expérience RNAseq est le suivant:



L'expérience retourne des petits bouts de séquence génomique appelés "reads". Le principe général est que le nombre de reads est proportionnel à l'abondance des ARN correspondants dans la cellule, le but étant d'estimer cette abondance.

Nous allons ici nous intéresser à la partie bioinformatique et biostatistique et pour cela détaillé de manière globale les différentes étapes.
Lors de l'expérience, le séquenceur produit des fichiers au format fastQ. Ce format contient la séquence ADN (comme un fichier fasta) ainsi que la qualité du read. Ces fichiers permettent donc d'appréhender la qualité de séquençage du read.

(Image provenant du site : http://genome.jouy.inra.fr/~orue/module-5-Methodes-Outils/seance1/slides.html#/)

1 read équivaut à 4 lignes. La première ligne étant l'identifiant du read, la deuxième la séquence, la troisième le séparateur '+' et la dernière la qualité.
La qualité correspond à la probabilité que le séquenceur, lors du séquençage, ne se soit pas trompé. A chaque nucléotide est associé un score de qualité relié de façon logarithmique à la probabilité d'erreur sous forme de caractère : le phred score.

De ces fichiers fastQ, avec un outil, on obtient des fichiers fastQC. L'outil qui a été utilisé ici est FastQC. Les fichiers fastQC sont des formats de rapport sur la qualité de séquençage. Ils donnent plusieurs informations globales sur la qualité des mesures faites sur les reads. Les informations apportées sont entre autres la distribution de la qualité en fonction des bases, le score de qualité par séquences, le contenu des séquences par base, le contenu en GC par read, le pourcentage de nucléotide non trouvé, la distribution de taille des reads, le nombre de fois où une séquence apparaît, etc.
Toutes ces mesures permettent donc d'évaluer la qualité des données afin de savoir si une analyse bioinformatique peut être ensuite effectuée.

La prochaine étape consiste à enlever les adaptateurs des reads et d'enlever les reads de faible qualité. En effet, lors d'une expérience le séquenceur séquence, en début de séquence, un adaptateur puis il est possible que cela se produise aussi en fin de séquence (cela est notamment souvent le cas lors de séquençage de petit ARN en raison de leurs petites tailles). Ces fichiers "coupés" sont réalisés grâce à l'outil TrimGalore. Par la suite, il convient de refaire un FastQC sur les reads "coupés".

D'autres étapes de pre-processing sont ensuite effectuées : l'éviction des contaminants du génome (BBSplit) et de l'ARN ribosomal (SortMeRNA).

L'objectif est de pouvoir aligner les reads sur le génome de références afin de définir leur position sur celui-ci. L'étape d'alignement s'effectue en deux temps:

- la création de l'index : c'est un arbre des suffixes qui permet de donner toutes les suffixes possibles dans le génome de référence.
- l'alignement des reads

La création de l'index s'effectue sur le génome de référence au format fasta (ici avec l'outil STAR). Grâce à cet index l'alignement des reads sur le génome peut s'effectuer. Il peut y avoir plusieurs étapes et outils lors de l'alignement ici les outils sont STAR, Salmon, RSEM et HISATS. L'alignement produit des fichiers au format SAM. Les fichiers SAM sont des fichiers donnant les informations d'alignement des reads sur la référence. Le format SAM est composé de deux parties :

- une entête où les lignes commencent par un '@' suivi de 2 lettres.
- un enchaînement d'informations respectivement : le nom de la lecture, Flag au niveau des bits, le chromosome, la position sur le chromosome en 5' de l'alignement, la qualité de l'alignement, le nombre CIGAR, des informations sur le séquençage paired-end, la qualité des bases au format ASCII (comme pour le format fastQ).

Le format SAM est lisible par l'humain mais pas par l'ordinateur. Pour cela les fichiers SAM doivent être convertis au format BAM. Le format BAM est un fichier SAM compressé. Afin d'être utilisable ces fichiers BAM doivent être indexés puis triés. Ces fichiers indexés et triés deviennent des formats BAM.bai. Toutes ces différentes étapes s'effectuent avec l'outil SAMTools.

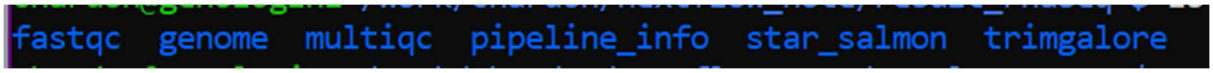
Pour comparer les alignements sur les gènes, on peut utiliser le génome de référence, les différents fichiers BAM et le fichier d'annotation au format GTF dans des logiciels tel que IGV. Les formats contenant toutes ces informations et transmis à IGV sont des formats bigWig.

Afin de pouvoir visualiser combien de reads colocalisent avec des gènes, il va falloir à partir de tous les fichiers BAM créer une seule matrice de comptage. A partir de cette matrice de comptage s'effectue un contrôle qualité via différentes mesures comme la distribution des reads, les annotations de jonction, la taille des reads , … .

b. Interprétation des principaux résultats:

Les différents dossiers regroupant les différents fichiers fournit à la suite de l'analyse sont les suivants:

1. Fastqc



Comme expliqué précédemment (cf 3.a.) les fichiers fastQC sont des rapports sur la qualité de séquençage. Ils permettent une visualisation graphique des différentes mesures d'intérêt avec un code couleur : Vert = bon, orange = attention et rouge = mauvais. Il faut faire attention aux données que l'on analyse car fastQC étant à l'origine conçu pour des données génomiques, certaines métriques peuvent être indiquées comme mauvaises alors que ce n'est pas réellement le cas. Si les données sont des données transcriptomiques, la composition en bases peut être fortement impactée par des séquences surreprésentées. La présence d'adaptateurs perturbe également la composition.

2. Genome:

Contient toutes les informations sur les fichiers de référence du génome ainsi que les fichiers en rapport avec la création de l'index expliqué au-dessus (cf 3.a.). Ce répertoire est lancé si dans le job le paramètre --save _reference est indiqué.

3. Pipeline_info:

Contient les rapports générés par Nextflow : execution_report.html, execution_timeline.html, execution_trace.txt et pipeline_dag.svg. Ces rapports permettent d'avoir aperçu de la distribution de l'utilisation des ressources pour chaque processus (cpu, mémoire, le temps du job) ainsi la chronologie d'exécution des processus.
Ce dossier contient aussi le fichier : samplesheet.valid.csv qui correspond aux données utilisées en entrée mais reformatées.
De plus, il contient le rapport généré par le pipeline: software_versions.yml qui regroupe les informations sur les versions utilisées de chaque outil.

4. Star_salmon

STAR est un outil permettant l'alignement des données de séquençage ARN alors que Salmon est un est un outil de quantification rapide des unités de transcription à partir de données RNA-seq. Il a besoin d'un ensemble de transcrits cibles (provenant d'un assemblage de référence ou *de novo)* afin de réaliser la quantification.
Les fichiers BAM généré par STAR sont fournis à Salmon pour la quantification
Ce dossier contient tous les fichiers qui ont été générés par cet ensemble d'outils :

- Les fichiers *.sorted.bam.bai sont, comme expliqués précédemment, les fichiers des reads alignés indexés et triés.
- Les dossiers *.sorted.bam contiennent les fichiers des reads alignés triés.
- Les fichiers .merged.gene_count sont des matrices de comptages des reads qui colocalisent avec les gènes sous différents formats.
- Les fichiers .merged.transcript_counts.tsv sont des matrices de comptages des unités de transcriptions (comme expliqué précédemment 3.a).
  A noter que l'on peut en plus de Salmon utilisé ensuite DESeq2 avec pour corriger les changements de la longueur moyenne des unités de transcription entre les échantillons.
- Le dossier bigwig contient les fichiers bigwig. Comme expliqué précédemment, le format bigWig est un format binaire indexé utile pour afficher des données denses et continues dans les navigateurs génomiques tels que IGV. Cela évite de devoir charger les fichiers BAM, beaucoup plus volumineux, pour visualiser les données.
- Le dossier qualimap contient des informations sur nos données traitées par cet outil. Qualimap est une application indépendante de la plate-forme, écrite en Java et R, qui fournit à la fois une interface utilisateur graphique (GUI) et une interface en ligne de commande pour faciliter le contrôle de la qualité des données de séquençage par alignement. En bref, Qualimap :
  - Examine les données d'alignement de séquençage en fonction des caractéristiques des lectures mappées et de leurs propriétés génomiques.
  - Fournit une vue d'ensemble des données qui aide à détecter les biais dans le séquençage et/ou la cartographie des données et facilite la prise de décision pour une analyse ultérieure.
- DupRadar est une bibliothèque Bioconductor écrite dans le langage de programmation R. Elle génère diverses mesures de contrôle qualité et des graphiques qui mettent en relation le taux de duplication et les niveaux d'expression des gènes afin d'identifier les expériences présentant un taux élevé de duplication technique. Ainsi le dossier de même nom contient les graphiques et informations traité par cet outil de nos données.
- L'outil Preseq vise à prédire et à estimer la complexité d'une bibliothèque de séquençage génomique, ce qui équivaut à prédire et à estimer le nombre de reads redondants à partir d'une profondeur de séquençage donnée et le nombre qui sera attendu d'un séquençage supplémentaire à partir d'une expérience de séquençage initiale. Les estimations peuvent ensuite être utilisées pour examiner l'utilité d'un

séquençage supplémentaire, optimiser la profondeur de séquençage ou cribler plusieurs bibliothèques pour éviter les échantillons de faible complexité.

- Le dossier featurecounts contient les fichiers de sortie de l'outil du même nom. Cet outil permet comme expliqué en généralité d'obtenir la matrice de comptage des gènes alignés. Il s'agit d'un contrôle de qualité supplémentaire permettant de vérifier quelles caractéristiques sont les plus abondantes dans l'échantillon et de mettre en évidence les problèmes potentiels tels que la contamination par l'ARNr.
- Desq2 est un package de R permettant d'effectuer une analyse d'expression différentielle pour les ensembles de données RNA-seq. Ainsi cela permet de donner des informations sur la reproductibilité entre échantillons. On retrouve dans ce dossier un graphique PCA et une heatmap montrant les distances euclidiennes par paire entre les échantillons de l'expérience. Cela permet de montrer la similarité entre les groupes d'échantillons et de révéler les problèmes potentiels de l'expérience.
- Dans le dossier picardmetrics se trouve les sorties de l'outil picard MarkDuplicates. Par défaut, le pipeline utilise picard MarkDuplicates pour marquer les lectures dupliquées identifiées parmi les alignements afin de permettre d'évaluer le niveau global de duplication au sein des échantillons.
- RSeQC est un ensemble de scripts conçus pour évaluer la qualité des données RNA-seq. Ce pipeline exécute plusieurs scripts RSeQC, mais pas tous. Les scripts supportés peuvent être modifié en fonction de ce que l'on souhaite exécuter en ajustant le paramètre --rseqc_modules qui, par défaut, exécutera tous les scripts suivants : bam_stat.py, inner_distance.py, infer_experiment.py, junction_annotation.py, junction_saturation.py, read_distribution.py et read_duplication.py. Il y a donc dans le dossier rseqc tous les répertoires représentant les scripts décrits au-dessus:
    - Read distribution : cet outil calcule comment les reads alignés sont distribués sur les structures génomiques. Un bon résultat pour une expérience RNA-seq standard est généralement d'avoir autant de lectures exoniques que possible (CDS_Exons). Une grande quantité de lectures introniques pourrait indiquer une contamination par l'ADN dans l'échantillon, mais peut être attendue pour une préparation d'ARN total.
    - Junction annotation compare les jonctions d'épissage détectées à un modèle de gène de référence. L'annotation de l'épissage est effectuée à deux niveaux, le niveau de l'événement d'épissage et le niveau de la jonction d'épissage.
    - inner distance : permet de calculer la distance entre deux paires de lectures. Il s'agit de la distance entre la fin de la lecture 1 et le début de la lecture 2, et elle est parfois confondue avec la taille de l'insert.
    - Junction saturation : ce script montre le nombre de sites d'épissage détectés dans les données à différents niveaux de sous-échantillonnage.
    - read duplication: permet de montrer le nombre de lectures ayant des doublons.
    - BAM stat: contient un ensemble d'informations statistiques sur les fichiers BAM alignés.
- Le dossier StringTie est le dossier de sortie de l'outil Stringtie. C'est un assembleur des alignements RNAseq qui utilise un nouvel algorithme de flux de réseau ainsi qu'une étape optionnelle d'assemblage *de novo* pour assembler et quantifier des transcrits complets représentant de multiples variantes d'épissage pour chaque locus génétique.
- Le dossier log contient :
    - *.SJ.out.tab: Fichier contenant les jonctions d'épissage filtrées détectées après le mappage des lectures.

- *.Log.final.out: Rapport d'alignement STAR contenant le résumé des résultats de cartographie.
- *.Log.outet *.Log.progress.out: fichiers STAR contenant des informations détaillées sur l'exécution. Généralement utile uniquement à des fins de débogage.

- Les dossiers 1 et 2 sont respectivement des dossiers donnants des informations sur nos 2 runs tel que des reads ambigus, la quantification des gènes, les rapports log …

5. Trimgalore

Ce dossier contient toutes les informations concernant l'utilisation de cet outil sur les fichiers fastQ donnés en entrée. Comme expliqué précédemment dans les généralités, cet outil permet de 'couper' les reads en enlevant les adaptateurs. A l'intérieur de ce dossier il y a donc les rapports contenant les informations sur ce qui a été enlevé sur les reads de chaque run. Le dossier contient en plus un autre dossier regroupant les nouveaux rapports fastQC effectués sur ces fichiers fastq "trimmés".

c. Interprétation du report MultiQC

MultiQC est un outil de visualisation qui génère un rapport HTML unique résumant tous les échantillons du projet. La plupart des résultats du contrôle qualité du pipeline sont visualisés dans le rapport et d'autres statistiques sont disponibles dans le répertoire de données du rapport.

Les résultats générés par MultiQC regroupent les contrôles de qualité des outils pris en charge, à savoir FastQC, Cutadapt, SortMeRNA, STAR, RSEM, HISAT2, Salmon, SAMtools, Picard, RSeQC, Qualimap, Preseq et featureCounts.

Dans le rapport multiQC l'échantillon noté 1 correspond aux mutants et l'échantillon noté 2 correspond aux WT.

Le premier onglet General Statistics donne des informations générales sur les expériences en regroupant les mesures de différents outils. On s'aperçoit que les deux samples sont semblables sur plein de mesures comme le pourcentage d'alignement, le taux d'erreur, le pourcentage de duplicat de reads, ... Cependant on remarque une différence au niveau du nombre de reads alignés sur le génome de référence. En effet, le sample 1 possède 3.7 millions de reads alignés alors que le deuxième seulement 2.7millions.
Via les graphiques suivant l'ACP et la heatmap provenant de DESeq2, on remarque que les deux expériences en termes de similarité entre les groupes d'échantillons sont équivalentes. En effet, les distances entre X1/X2 et X2/X1 sont égales et les distances X2/X et X1/X1 sont bien égales à 0. Cela signifie que sur les informations données par la heatmap il n'y a pas eu de problèmes potentiels au sein des expériences.
Dans le graphique Biotype Counts, on s'aperçoit que 100% des reads sont alignés avec le génome et que ces parties alignées sont des endroits codant pour des protéines. Comme vu dans les informations générales, l'échantillon 2 possède moins de reads alignés que l'échantillon 1.
Le graphique de Dupradar permet d'évaluer le pourcentage de duplication au sein des échantillons. On s'aperçoit que pour les deux échantillons, plus les gènes sont exprimés et plus le pourcentage de duplication augmente. Cette augmentation reste dans les seuils car elle commence réellement lorsque les reads ont une expression supérieure à 1000. On voit aussi que l'échantillon 2 possède plus de reads dupliqués quand les gènes sont plus exprimés.

Sur le graphique des sorties de l'outil Picard, on voit bien que les mutants présentent plus de reads alignés sur le génome mais pour les deux échantillons les pourcentages de duplications sont les mêmes.

Dans les analyses suivantes on voit que les deux échantillons possèdent les mêmes pourcentages de reads exoniques, introniques et intergéniques, que la distribution moyenne de la profondeur de couverture sur la longueur de tous les transcrits alignés est la même et que la distribution des reads est globalement la même. Plusieurs autres metriques présentent les même resultats pour les deux échantillons.
On s'aperçoit par contre que sur le graphique de la "Inner distance", il y a une réelle différence de pourcentage entre les 2 échantillons. En effet, dans l'échantillon 2 donc des WT, il y a nombre plus élevé de paires de reads qui ont une distance égale à 0. Donc que les reads alignés se trouvent à la même distance.

Concernant maintenant les analyses des rapports FastQC. Les nombres de reads entre les deux réplicats de chaque expérience sont identiques. Cela signifie que les expériences sont reproductibles.
Grâce à l'histogramme montrant la qualité des séquences, on voit bien que les expériences sont de bonnes qualités et équivalentes entre les deux réplicats (il est normal que la qualité du séquençage baisse en fin de séquençage). Les scores de qualité pour les reads sont bons, ainsi que le pourcentage en GC et la distribution des longueurs des séquences.
Cependant le contrôle qualité du contenu des séquences en base est annoncé comme mauvais. En effet, idéalement les 4 lignes correspondant à chaque base devraient se superposer car le pourcentage de chaque base est équivalent. Or ici, au début des séquences on voit une variabilité jusqu'à environ 15pb. Sauf que certains types de librairies produiront toujours une composition de séquence biaisée, notamment au début de la lecture. Ce biais n'implique pas une séquence spécifique, mais fournit plutôt un enrichissement d'un certain nombre de différents K-mers à l'extrémité 5' des lectures. Bien qu'il s'agisse d'un véritable biais technique qui ne peut pas être corrigé avec un outil de "trimming", il ne semble pas affecter négativement l'analyse. Il produit donc une erreur mais elle n'est pas prise en compte.
On voit aussi un warning dans le module du taux de séquences dupliquées dans les "raw reads". En effet, on voit un pourcentage plus élevé de séquence dupliquée plus de 10 fois. Mais quand on regarde le rapport FastQC du même module mais pour les reads "coupés", on s'aperçoit qu'ils ne sont plus présents. Ces séquences devaient être de faibles qualités et ont été retirées.
Les reads filtrés ont étés plus nombreux dans les échantillons 1 : WT que pour les mutants. Cela peut donc expliquer qu'il y ai moins de reads alignés avec le génome de référence pour les WT car plus de reads ont étés enlevés.

En conclusion, on peut dire que les analyses en RNAseq sont de bonnes qualités et bien reproductibles entre les deux réplicats de chaque échantillon. On pourrait s'attendre à ce que les échantillons WT présentent plus de reads alignés avec le génome de référence puisqu'il y a eu moins de mutations. Mais grâce à certains graphiques, on a pu se rendre compte que plus de séquences ont été enlevées sur les reads WT donc il a eu moins de reads à alignés. En regardant les différents pourcentages des autres graphiques on voit que les deux échantillons sont comparables dans beaucoup d'autres mesures.
Pour finir les warning et fail des modules ne sont pas impactant pour l'analyse car il n'atteste pas réellement d'un problème critique dans l'expérience.