Master Bioinformatique
Université de Montpellier

HAU803I : Rapport de stage de M1

Titre à choisir

Etudiant :
Mickael Coquerelle

Professeur :
Anthony Boureux

Résumé

a remplir

Liste des abréviations

Acronymes

ADN	Acide DésoxyRiboNucléique
DGE	Analyse d'Expression Différentielle
ARN	Acide RiboNucléique
API	Application Programming Interface
CVS	Concurrent Versions System
FP	Faux Positifs
FN	Faux Négatifs
KB	KiloBase
SLA	Sclérose Latérale Amyotrophique
RCS	Révision Control System
SHD	Séquençage Haut Débit
SNP	Polymorphisme nucléotidique unique
SIF	Singularity Image Format
UML	Language de modélisation unifié
VP	Vrai Positifs
VN	Vrai Négatifs

Symboles

$\mathcal{A}_{\mathrm{x}}$	Alphabet de x
Σ_{x}	Somme de x
Q_P	Score de qualité Phred
Q_A	Score de qualité Phred encodé en ASCII
S	Séquence biologique
P	Motif recherché
S_e	Sensibilité
S_p	Spécificité
$\mathcal{T}$	Texte
w	Mot
$\mathcal{O}()$	Notation de Landau
$\mathcal{SA}_x$	Table des suffixes de x

Table des matières

1 Introduction

1.1 Environnement du stage

Ce travail est la synthèse de mon stage de première année de Master, durant lequel j'ai intégré l'équipe de la professeure Thérèse Commes du laboratoire Bio2M, rattaché à l'Institut national de la santé et de la recherche médicale (INSERM). J'ai eu la chance d'être accompagné dans mon apprentissage par Anthony Boureux, enseignant-chercheur. L'équipe collabore avec des services cliniques et des plateformes hospitalières, ce qui favorise la résolution de problématiques liées au diagnostic médical. Ainsi, j'ai eu l'occasion de contribuer à un projet de recherche translationnelle que Bio2M mène en partenariat avec le CHU de Nîmes. Ce travail est en lien direct avec des enjeux diagnostiques, puisqu'il concerne une maladie neurodégénérative : la sclérose latérale amyotrophique (SLA). Les différentes missions qui m'ont été confiées s'inscrivent dans le champ de la transcriptomique, et plus particulièrement dans le cadre de l'analyse d'expression génique appliquée à la SLA, l'idée étant d'initier une stratégie permettant de détecter une expression différentielle à l'échelle de certains gènes d'intérêt dans la SLA, avec des contraintes à la fois techniques et biologiques, pour *in fine* tenter d'identifier la causalité génétique chez les patients.

1.2 Contexte biologique

Une maladie rare se définit par une prévalence[1] de 0,05 % dans la population générale. Quatre-vingts pour cent de ces maladies sont d'origine génétique [4], et la SLA en fait partie : elle touche un individu sur 20 000 en Europe [9], et sa prevalence mondiale varie de 1,57 à 11,8 pour 100 000 selon les pays, de l'Iran aux États-Unis [3]. C'est une maladie neurodégénérative causée par une atteinte du motoneurone central au niveau du cortex cérébral **(figure 1)**, conduisant à une dégénérescence progressive des fonctions musculaires. Cette pathologie est très handicapante, tant sur le plan physique que social. En raison de sa gravité et des conséquences dévastatrices pour les patients et leur entourage, elle constitue un domaine de recherche de premier plan pour les généticiens cliniques. C'est pourquoi il est pertinent d'intégrer une approche transcriptomique afin d'augmenter le rendement diagnostique des formes génétiques de la maladie et de mieux en comprendre les mécanismes. Notons que les gènes responsables de la SLA sont globalement bien documentés. À ce jour, une quarantaine de gènes ont été identifiés et associés à la maladie. Dans 90 % des cas, leur implication est directe dans les formes familiales. Dans les 10 % restants, on observe

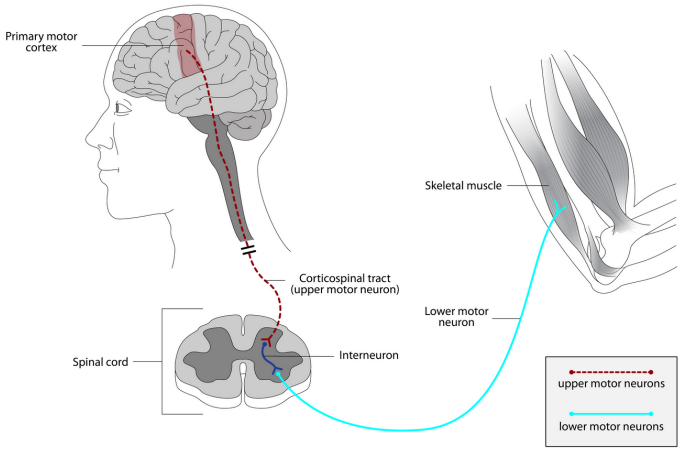

Figure 1 :
Atteinte neuronale dans la SLA(P.Wicks, 2024)

des formes dites sporadiques, où la causalité génétique est cette fois indirecte, via des perturbations de processus cellulaires clés tels que l'homéostasie[2] de l'ARN, le Transport axonal[3] ou l'autophagie[4]. On observe également une forte hétérogénéité génétique dans cette maladie, perceptible à travers l'implication de gènes aux fonctions parfois très différentes, mais qui convergent toujours vers une dégénérescence neuronale. Les gènes les plus fréquemment impliqués sont *SOD1*, *TARDBP*, *FUS* et *C9ORF72* [4]. Majoritaires dans la maladie, ils constituent le socle des recherches génétiques pour tenter la mise au point de thérapies ciblées et faire progresser la compréhension ainsi que le diagnostic de cette pathologie.

1.3 L'apport du RNA-Seq ciblé dans le périmètre de notre analyse

D'un point de vue biologique, *stricto sensu*, on sait que l'étape de transcription est fondatrice de la diversité protéomique ; elle constitue, de ce fait, une source majeure d'anomalies génétiques. À l'issue de ce mécanisme, un gène peut exprimer plusieurs isoformes, dont certaines peuvent avoir un impact pathologique. Tout l'objet de ce travail est de proposer une méthode pertinente pour détecter des différences d'expression de tel ou tel gènes, susceptibles d'aider le biologiste à établir un lien avec la maladie, notamment à travers une haploinsuffisance[5] ou, au contraire, une surexpression.

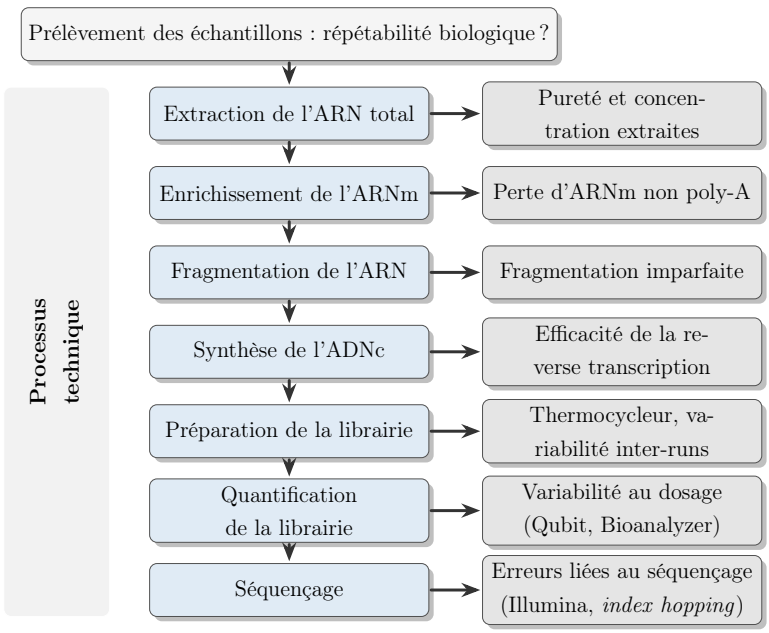

Figure 2 :
Processus expérimental du RNA-Seq et biais potentiels

Supposons maintenant que l'on cherche à établir un profil d'expression génique pour notre quarantaine de gènes. Il convient alors de s'interroger sur le support de lecture de chacun de ces gènes, c'est-à-dire le nombre de fois qu'une région de l'ADN a été lue (ou comptée) au cours du séquençage. On s'attend logiquement à observer une distribution des lectures, interprétable comme le reflet du niveau d'expression de chaque gène étudié. En pratique, si l'on se fonde sur les données brutes, cette hypothèse s'avère souvent éloignée de la réalité : un certain nombre de variables — biologiques (rythme circadien, état physiologique, etc.) ou expérimentales (moment du prélèvement, conditions de manipulation, etc.) — influencent cette distribution théorique, introduisant une dispersion non négligeable dans le jeu de données.

Par ailleurs, le traitement bioinformatique peut lui aussi fausser le profil d'expression, notamment à travers les choix opérés lors de l'étape d'alignement — comme nous le verrons. La **figure 2** illustre quelques-uns des biais, non exhaustifs, susceptibles d'affecter la quantification de l'expression génique. On comprend à travers cette illustration qu'à chaque étape technique du protocole RNA-seq,

de la préparation des échantillons à l'analyse bioinformatique, une part de variabilité est introduite, parfois de manière systématique entre les expériences, parfois de manière aléatoire.

Toute la stratégie de quantification différentielle de l'expression génique (DGE) consiste à limiter ces biais ou, le cas échéant, à les intégrer dans l'analyse — afin d'obtenir des résultats à la fois fidèles à la réalité biologique du patient, mais également reproductibles dans un contexte de routine diagnostique. Je m'efforce ici d'effectuer un certain nombre de vérifications sur les données expérimentales, notamment à l'aide d'analyses statistiques descriptives, et *in fine*, de suggérer de corrections appropriées (par la phase de normalisation).

À cet égard, j'ai mené un travail bibliographique visant à établir un état des lieux des stratégies conventionnelles , dont je propose une synthèse dans la **figure 3**. Ces investigations m'ont permis d'approfondir ma compréhension de l'analyse quantitative en RNA-Seq.

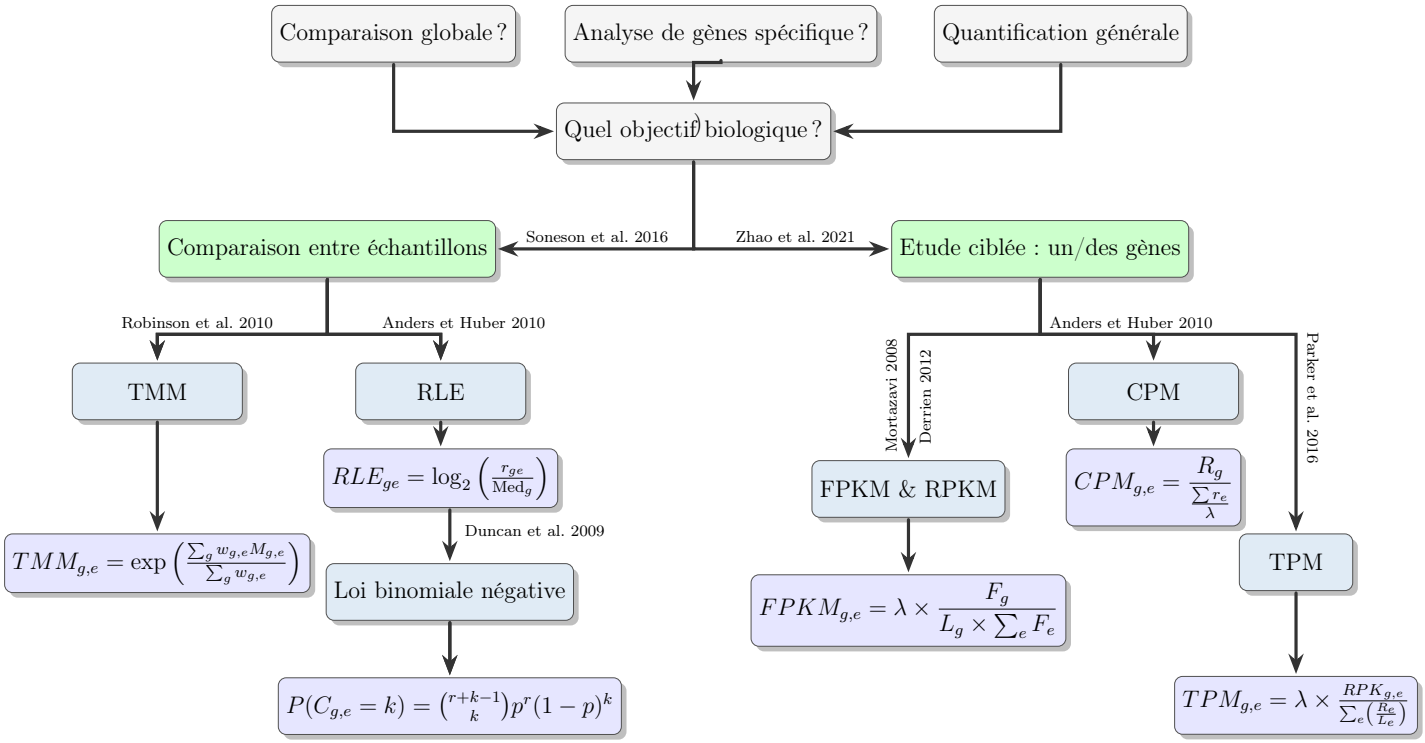

Figure 3 : Schéma décisionnel pour normaliser les données RNASeq

1.4 Problématique et réflexion sur les choix expérimentaux

Au regard des éléments mentionnés précédemment, ce travail à un double objectifs en avançant sur les deux questions suivantes :

— Les méthodes conventionnelles de normalisation sont-elles adaptées à l'analyse quantitative de panels de gènes ciblés ?

— Quelle crédibilité accorder à cette même analyse quantitative en RNA-Seq ciblé dans notre contexte expérimental peu reproductible ?

En effet, à la différence d'analyse du transcriptome global, le RNA-Seq ciblé produit des données peu bruyante — et pour cause : l'étape expérimentale d'enrichissement conduit à une surreprésentation des gènes d'intérêt. Cette absence de « consistance » dans les données de comptage complexifie l'interprétation statistique et appelle, disons, à une adaptation soit de l'approche analytique.

Une dernière question mérite d'être posée : pourquoi avoir opté pour une approche de RNA-Seq ciblé plutôt qu'une analyse transcriptomique globale ? On peut dégager plusieurs arguments à cette question. À première vue, en se concentrant sur un panel restreint de gènes d'intérêt (ceux impliqués dans la SLA), j'améliore la sensibilité de détection tout en réduisant les coûts expérimentaux, d'analyse bioinformatiques et tertiaire. C'est d'autant plus pertinent dans une perspective diagnostique, puisque les résultats sont obtenus plus rapidement et sont plus rapidement exploitables, on peut raisonnablement penser que l'analyse biologique d'un transcriptome globale et bien plus chronophage.

A l'inverse, cette approche limite l'analyse aux seuls gènes ciblés, excluant par là même occasion, la détection d'événements potentiellement pertinents, tels que des isoformes rares ou des altérations affectant d'autres régions du transcriptome. Il s'agit donc d'un compromis, le RNA-Seq global offre une vision d'ensemble, mais au prix d'une complexité analytique accrue, d'un coût plus élevé, et d'un bruit de fond plus important qui nous le verrons peut être utile à certaines étapes du *pipeline*.

Tout est discutable, il n'existe pas, à proprement parler, de stratégie idéale : tout dépend du contexte médical, de la pertinence clinique et des contraintes techniques et financière. Pour autant, dans le cadre des maladies rares comme la SLA, où les gènes impliqués sont généralement bien caractérisés, le RNA-Seq ciblé constitue une option pragmatique. Ainsi, les sections suivantes décrivent la méthodologie adoptée pour contrôler la qualité des données, évaluer l'impact de l'alignement, et amorcer une réflexion sur les approches de quantification adaptées à l'étude que j'ai en charge de mener, et je propose ensuite une méthode de quantification fondée sur une approche par $k$mers, qui pourrait être explorée pour contourner toutes les contraintes évoquées et les limites inhérentes aux méthodes classiques (voir **figure 3**), en conservant une résolution suffisante pour la détection de l'haploinsuffisance.

2 Matériels & Méthodes

2.1 L'étape d'alignement

2.1.1 Présentation conceptuelle de STAR et CRAC

Ajouter explication BWT[1], SA [2], et approche par kmer ...

Ainsi, j'ai cherché à évaluer l'impact de cette étape en générant les fichiers `BAM` pour l'intégralité des patients concernés à l'aide de deux outils distincts : `STAR` et `CRAC`, fondés sur des stratégies algorithmiques différentes. La majeure partie des commandes, ainsi que leurs paramètres, sont consul-

tables dans un Makefile disponible sur le Git. Comme évoqué dans la partie introductive, nous travaillons ici sur un jeu de données limité à 72 patients. La génération des fichiers BAM [6] s'effectue à partir de deux fichiers FASTQ [5], obtenus via la technologie Illumina™, puisque les données ont été produites en mode séquençage pairé[6].

2.1.2 Génération des fichiers d'alignement au format BAM

Dans l'absolu, ce qui nous intéresse, c'est le fichier consignant les métriques d'alignement générées durant l'exécution de l'outil, et non directement le fichier BAM lui-même. Pour autant, ce fichier, est produit pendant la génération du BAM. Il s'agit d'un fichier « log » pour STAR et d'un fichier dénommé « summary » pour CRAC. Pour automatiser ces opérations, j'ai conçu un *pipeline* en Bash à l'aide de l'utilitaire Make, de manière à traiter séquentiellement les paires de fichiers FASTQ.

```
SAMPLES = $(shell \
   for R1 in $(FASTQ_DIR)/*_1.fastq.gz; do \
      R2= echo $$R1 | sed 's/_1.fastq.gz/_2.fastq.gz/'; \
      if [ -f $$R2 ]; then \
         basename $$R1 _1.fastq.gz; \
      fi; \
   done)
```

Code 1 : Construction de la variable SAMPLES pour traiter séquentiellement les FASTQ

Les cibles que je vous présente ci-dessous sont conçues pour rendre l'exécution la plus générique possible. En effet, le répertoire de stockage des dépendances (comme l'index, par exemple) ainsi que celui des fichiers de sortie dépendent de l'organisation de l'utilisateur sur sa machine d'exécution. Par ailleurs, la documentation de STAR référence une quantité substantielle d'options mais l'objectif ici n'est pas de réaliser une comparaison formelle de l'outil, mais plutôt d'évaluer dans quelle mesure une approche alternative de *mapping* influence la manière dont les lectures sont classées.C'est une première étape dans les investigations visant à caractériser le jeu de données et à évaluer la viabilité d'une analyse DGE dans le cadre du RNA-seq ciblé et ses contraintes présentées en introduction. Le code ci-dessous montre la construction de la règle pour lancer STAR :

```
# Détection des échantillons par présence des fichiers FASTQ _1 et _2
BAMS_STAR = $(addprefix $(OUTBAM_STAR)/, $(addsuffix .bam, $(SAMPLES)))

Star_Paire: $(BAMS_STAR)
$(OUTBAM_STAR)/%.bam:
  @mkdir -p $(OUTBAM_STAR) $(OUTLOG_STAR) $(TMPDIR)/$*;
  @echo ">> Lancement de l'alignement de l'échantillon $* avec STAR";
  STAR --runThreadN $(THREADS)
    --genomeDir $(REF_STAR)
    --readFilesIn $(FASTQ_DIR)/$*_1.fastq.gz $(FASTQ_DIR)/$*_2.fastq.gz
    --readFilesCommand zcat --outSAMtype BAM SortedByCoordinate
    --outFileNamePrefix $(TMPDIR)/$*/;
  @mv $(TMPDIR)/$*/Aligned.sortedByCoord.out.bam $(OUTBAM_STAR)/$*.bam;
```

```
14  @mv $(TMPDIR)/$*/Log.final.out $(OUTLOG_STAR)/$*.Log.final.out;
```

Code 2 : Cible `Star_Paire` pour générer les `BAM` avec STAR

Concernant l'utilisation de `Crac`, le paramétrage a également été laissé par défaut, à l'exception du choix de la taille du *kmer*, indispensable puisque l'outil repose sur une approche de type *kmer matching*. Evidemment, la même référence génomique a été utilisée pour `STAR`, afin de garantir la comparabilité des métriques. Notons que ce choix de taille du *kmer* est fondamental dans la mesure ou il conditionne la sensibilité et la spécificité de l'aligneur. Si il est trop court on augmentera la sensibilité (autrement dit sa capacité à détecter des correspondances même en présence d'erreurs ou de mutations), mais au prix d'une plus grande ambiguïté et d'un risque accru d'alignement non spécifique [8] . À l'inverse, un *kmer* plus long favorise la spécificité, mais peut perdre en sensibilité, notamment dans des régions génomiques complexes ou peu couvertes.

Dans notre cas, un compromis a été établi avec une taille de **31 bases**. Ce choix n'est pas anodin : il permet notamment de représenter efficacement les *kmers* en mémoire en utilisant un encodage binaire compact, où chaque base (A, C, G, T) est encodée sur 2 bits. Ainsi, un *kmer* de 31 bases tient sur 62 bits, ce qui permet de l'encoder dans un entier de 64 bits, sans dépassement de capacité, tout en optimisant les performances en termes de vitesse d'accès et d'empreinte mémoire. Ce type de représentation est couramment utilisé dans les structures de données bioinformatiques comme les tables de hachage. Ce réglage technique conditionne donc les performances algorithmiques de `CRAC` :

```
1  CRAC_SAMS = $(addprefix output/crac/bam/, $(addsuffix .sam, $(SAMPLES)))
2  Crac_Paire: $(CRAC_SAMS)
3
4  # Règle pour chaque .sam
5  output/crac/bam/%.sam: $(FASTQ_DIR)/%_1.fastq.gz $(FASTQ_DIR)/%_2.fastq.gz
6    @echo "Vérification de l'échantillon : $*"
7    @if [ ! -f output/crac/bam/$*.bam ]; then
8       echo "Lancement de l'alignement de l'échantillon : $* avec CRAC";
9       mkdir -p output/crac/summary output/crac/log output/crac/bam; \
10      gunzip -c $(FASTQ_DIR)/$*_1.fastq.gz > $(FASTQ_DIR)/$*_1.fastq; \
11      gunzip -c $(FASTQ_DIR)/$*_2.fastq.gz > $(FASTQ_DIR)/$*_2.fastq; \
12      crac --nb-tags-info-stored 10000 --bam --stranded -i $(REF_CRAC) -k
            $(KMER_CRAC) \
13          --summary output/crac/summary/$*.summary \
14          --nb-threads $(THREADS) -r $(FASTQ_DIR)/$*_1.fastq
               $(FASTQ_DIR)/$*_2.fastq -o output/crac/bam/$*.sam \
15          2> output/crac/log/$*_crac.log;
16   else
17      echo "Fichier déjà aligné : $@";
18   fi
```

Code 3 : Cible `Crac_Paire` pour générer les `BAM` avec CRAC

Notons qu'une étape supplémentaire est nécessaire, car l'aligneur génère nativement une sortie

au format SAM [10]. J'ai donc rédigé une cible supplémentaire pour convertir les fichiers SAM en BAM grâce à l'utilitaire Samtools :

```
# Génère une liste des fichiers SAM sans extension
SAM_FILES := $(basename $(notdir $(wildcard output/crac/bam/*.sam)))

# Règle principale pour convertir tous les SAM en BAM+BAI
Convert_bam_sam: $(addprefix output/crac/bam/, $(addsuffix .bam, $(SAM_FILES)))

output/crac/bam/%.bam: output/crac/bam/%.sam
  @echo "Conversion SAM -> BAM pour léchantillon : $*"
  #Conversion (view), en binaire (-b), avec une entrée SAM (-S) ET et trie du
     BAM en position génomique (sort) :
  samtools view -@ $(THREADS) -bS $< | samtools sort -@ $(THREADS) -o $@
  samtools index $@
  @rm -f $<
```

Code 4 : Cible Crac_Paire pour générer les BAM avec CRAC

2.1.3 Traitement des fichiers de sortie

Ces opérations effectuées, on obtient des formats assez différents (cf. annexe **5.1**), pour lesquels il est nécessaire d'extraire les métriques pertinentes afin d'évaluer s'il existe une différence significative dans la capacité des outils à aligner de manière unique les lectures. Si tel est le cas, il convient ensuite d'examiner si cette différence peut impacter les analyses quantitatives réalisées *a posteriori*. Pour ce faire, j'ai développé un petit script Bash dont le code est consultable sur GitHub. Dans les grandes lignes, le script démarre par la définition des répertoires sources, puis effectue les vérifications d'usage, (présence des fichiers par exemples). Pour chaque log associé au BAM, j'extrais les métadonnées du patient (*Run*, *Patient*, *Type*) contenues dans le nom du fichier, tout ces indicateurs sont récupérés à l'aide de commandes shell standard (grep, cut, tr, sed), en tenant compte des spécificités de chaque catégorie de log présenté en annexe **5.1**.

Au dela du développement de ce script Bash j'ai eu à compléter la règle du *pipeline* Snakemake[7] au laboratoire. J'ai intégré la logique « d'extraction de ces métriques, ce qui n'était pas prévu au départ dans la règle historique. Cette contribution est disponible sur le Git et en annexe **5.2**. Mon fichier tabulé effectivement généré, j'ai compléter l'analyse en m'intéressant au flag NH (*Number of Hits*) et CIGAR (*Compact Idiosyncratic Gapped Alignment Report*), il s'agit de champs par défaut du fichier SAM [10]

Dans les grandes lignes, la stratégie consiste à extraire toutes les lectures alignées de manière unique dans CRAC, à récupérer leur valeur NH qui logiquement est égale à 1, ainsi que les informations de qualité CIGAR (*Compact Idiosyncratic Gapped Alignment Report*), puis à les comparer avec la métrique NH associée dans STAR pour un même read. L'objectif est d'identifier des pistes pour expliquer une éventuelle différences de classification des *reads* entre les outils.

À ce stade, je recommande d'intégrer une comparaison de méthode plus approfondie des deux outils d'alignement, car ce travail n'est pas suffisant pour mesurer l'influence de la stratégie d'alignement sur le pourcentage de lectures mappées uniques. Pour autant, il serait intéressant d'aller plus loin, afin d'explorer l'efficacité intrinsèque des outils et chercher à optimiser cette phase du pipeline, bien que cela ne soit pas l'objet du présent travail.

Statistiques descriptives sur la profondeur de lecture, par run et par échantillon. Mise en évidence de la non-reproductibilité : inter-run vs. intra-run. Visualisations pour illustrer la disparité de couverture.

Présentation de STAR et CRAC : Méthodologie, hypothèses sous-jacentes, différences de traitement.

Alignement des mêmes échantillons avec les deux outils.

Comparaison des métriques de mapping : taux d'alignement unique/multiple/non-aligné.

Conclusion : validation que l'étape d'alignement n'est pas responsable de la variabilité observée.

2.1.4 Vers une approche alternative de quantification

Justification du besoin de s'affranchir des biais d'alignement.

Introduction à la quantification par kmer : promesse de neutralité méthodologique.

Transition vers une stratégie plus robuste de détection différentielle.

2.1.5 Qualité et spécificité de l'alignement

2.2 Génération de la table de comptage

FeaturesCount ....

https ://rdrr.io/bioc/Rsubread/man/featureCounts.html regle de génération :

2.3 Recherche de la bonne approche pour normaliser les données

L'objectif de cette section, est de présenter les trois approches de normalisation que j'ai étudiées durant le stage, TPM, TMM et RLE, avec l'objectif de choisir la plus adéquate pour normaliser les données pour l'analyse DGE. Quelques éléments de formalise s'imposent concernant les notations. On supposera que chaque expérience de SHD a permis d'obtenir des *reads* alignés (R_g), des comptages (C_g) ou des fragments (F_g) sur un gène d'intérêt (g) de longueur (L_g), en kilobases, pour un échantillon (e) aligné sur un génome de référence — ici GRCh37, afin de conserver une référence cohérente avec les expériences menées sur l'ADN au laboratoire sur la SLA. Les comptages de *reads* (en ordonnée y) sont étiquetés avec l'identifiant de l'échantillon ou le *run* (en abscisse x) pour la majorité des graphiques qui seront présentés. Enfin, comme je l'ai évoqué en introduction, on admet l'hypothèse selon laquelle l'abondance des $C_{g,e}$ est proportionnelle à l'expression du gène en question. Ces remarques effectuées, passons aux aspects purement méthodologiques.

2.3.1 TPM : Transcrit par million

Cette méthode a été introduite par Wagner et al. en 2012 [14]. L'objectif des TPM est de comparer les comptages entre les échantillons. Cela est possible grâce à l'ordre des opérations mathématiques. On va chercher à normaliser d'abord sur la longueur du gène, puis dans un second temps sur la profondeur échantillonnale. Cette technique est relativement sommaire, pour autant j'ai choisi d'intégrer les TPM dans ma comparaison de méthodes car la somme des $TPM_{g,e}$ est identique, ce qui facilite la comparaison des différents échantillons pour un même gène et il est facile de s'en convaincre :

Formule de calcul des TPM

$$\mathrm{TPM}_{g,e} = \lambda \cdot \frac{\mathrm{RPK}_{g,e}}{\sum_e \left(\frac{R_e}{L_e}\right)}$$

avec

$$\mathrm{RPK}_{g,e} = \frac{R_{g,e}}{L_{g,e}}$$

```
all_tpm |> group_by(Sample) |> summarise(tot_tpm = sum(tpm)) |> arrange(tot_tpm)

   Sample     tot_tpm
   <chr>          <dbl>
 1 2307121892   1000000
 2 C01P016      1000000
 3 C01P027      1000000
...
```

Code 5 : Vérification de l'égalité des TPM par échantillon

Par ailleurs, pour la partie normalisation sous `R`, j'ai travaillé avec la librairie `EdgeR` [13]. Pour autant, il n'existe pas de fonction intégrée pour le calcul des TPM ; ses modalités de calcul étant simples, je les ai donc codées manuellement. L'intégralité du script est disponible sur le `Git`. La stratégie TPM ne tient pas compte de l'hétérogénéité des librairies par exemple. Pour la suite, j'introduis les méthodes TMM et RLE, qui, en théorie, permettent de pallier à ces limitations. Du reste, nous verrons dans la section résultat si ces approches se comportent en pratique comme attendu en théorie.

2.3.2 Limiter l'impact des valeurs aberrantes : TMM

Il s'agit ici d'une approche fondamentalement différente de celle abordée précédemment, avec une formulation mathématique plus complexe. La compréhension du TMM (Trimmed Mean of M-values) repose sur trois notions fondamentales : la M-value, l'A-value, et les coefficients de pondération. La M-value $M_{g,e}$ est la première étape pour calculer le facteur TMM_e d'un échantillon e, par rapport à un échantillon de référence k, pour chaque gène g. Elle se définit comme le log-ratio de l'expression normalisée du gène g entre e et k, après correction des biais liés à la profondeur de lecture.

La M-value (TMM)

$$M_{g,e} = \log_2\left(\frac{R_{g,e}}{R_{g,k}}\right)$$

Pour obtenir une estimation fiable de $M_{g,e}$, deux critères sont généralement requis. Premièrement, la condition de référence k doit être biologiquement pertinente — par exemple, un échantillon non traité dans une étude évaluant la SLA (cela parait logique). Deuxièmement, cette référence doit être aussi homogène que possible vis-à-vis des variables confondantes (âge, sexe, etc.), afin de limiter l'introduction de biais dans les comparaisons.

Toutefois, ces conditions sont difficiles à satisfaire dans le cadre de ce travail, pour deux raisons principales : d'une part, j'ai rencontré des difficultés pour obtenir les métadonnées techniques associées aux échantillons controles ; d'autre part, les critères d'inclusion des patients dans le protocole n'ont pas nécessairement été définis en tenant compte des contraintes spécifiques aux analyses bioinformatiques en aval ...

La seconde notion, l'A-value ($A_{g,e}$), quantifie l'abondance moyenne du gène g dans les échantillons e et k. Il s'agit d'une moyenne géométrique exprimée en base logarithmique, qui se veut représenter l'expression globale du gène dans la comparaison. Bien qu'elle n'intervienne pas directement dans le calcul du facteur TMM, l'A-value permet de visualiser l'intensité moyenne du signal et d'explorer la stabilité des gènes en fonction de leur abondance. Il existe une relation mathématique simple entre M-value et A-value[robinson_scaling_2010]

Formule de calcul de la A-value

$$A_{g,e} = \frac{1}{2}\log_2\left(R_{g,e} \times R_{g,k}\right)$$

Enfin, les M-values sont pondérées par des coefficients $w_{g,e}$, qui reflètent leur précision. Ces poids sont définis comme l'inverse de la variance estimée des M-values, de manière à réduire l'impact des gènes dont l'expression est trop variable.

A-value (TMM)

$$M_{g,e} = \log_2(R_{g,e}) - \log_2(R_{g,k}) \quad \Longleftrightarrow \quad M_{g,e} =$$

Il est clair que le TMM est plus complexe à calculer que les normalisations linéaires précédentes. De plus, les détails de certains paramètres (filtrage, choix de k, pondérations exactes) sont difficilement accessibles directement dans la fonction `calcNormFactors()` d'`edgeR`. Nous proposons donc ci-dessous le code R utilisé pour la normalisation TMM, accompagné de fonctions simples pour illustrer les calculs de M-value et d'A-value :

Pondération des M-Value

$$TMM_e = \exp\left(\frac{\sum_g w_{g,e} M_{g,e}}{\sum_g w_{g,e}}\right), \quad \text{avec } w_{g,e} = \frac{1}{\mathrm{Var}(M_{g,e})}$$

En conclusion, le TMM constitue une méthode robuste de normalisation, en intégrant à la fois la stabilité relative des gènes (via la M-value) et leur abondance (via l'A-value), tout en éliminant les gènes extrêmes. Moins sensible à la profondeur de séquençage que CPM ou FPKM, elle constitue une alternative pertinente pour les jeux de données complexes ou peu reproductibles. Nous présenterons à présent une méthode complémentaire : la normalisation basée sur la médiane logarithmique, mieux adaptée à certaines contraintes biologiques.

3 Resultats

3.1 Analyse de la variabilité expérimental et levée d'ambiguité sur l'alignement

Il est intéressant de vérifier si les deux outils d'alignement diffèrent significativement dans leur capacité à aligner de manière unique les lectures. Dans un premier temps j'ai construit à partir

de mon fichier tabulé issue du script Bash Pour ce faire, j'ai réalisé une analyse statistique sur les proportions relatives des lectures uniques par patient. J'ai commencé par effectuer un test de normalité de Shapiro-Wilk [11] sur les données de proportions de lectures uniques pour chaque outil, afin de vérifier l'hypothèse de normalité nécessaire à l'utilisation d'un test paramétrique. [12]

Les résultats du test ne m'ont permis de rejeté cette hypothèse ; j'ai donc opté pour une approche non paramétrique plus robuste : le test de Wilcoxon [15] pour échantillons appariés.

Il permet de comparer les distributions des proportions de lectures uniques entre `STAR` et `CRAC`, tout en tenant compte de la structure appariée des données (chaque patient servant de contrôle pour lui-même en quelque sorte).

Mon objectif est ici de déterminer si les différences observées dans les taux d'alignement unique relèvent de fluctuations aléatoires ou d'une divergence systématique entre les deux méthodes d'alignement. Pour autant il ne s'agit pas d'une comparaison d'outil en bon et due forme, mais de regarder grossièrement dans qu'elle mesure deux stratégies différentes pourraient influer sur la profondeur de lectures alignés. Subsidiairement, si il existe une significativité, est ce que cela peut influer en tout ou partie sur la variabilité des unités de comptages (nombres de lectures alignés à une position).

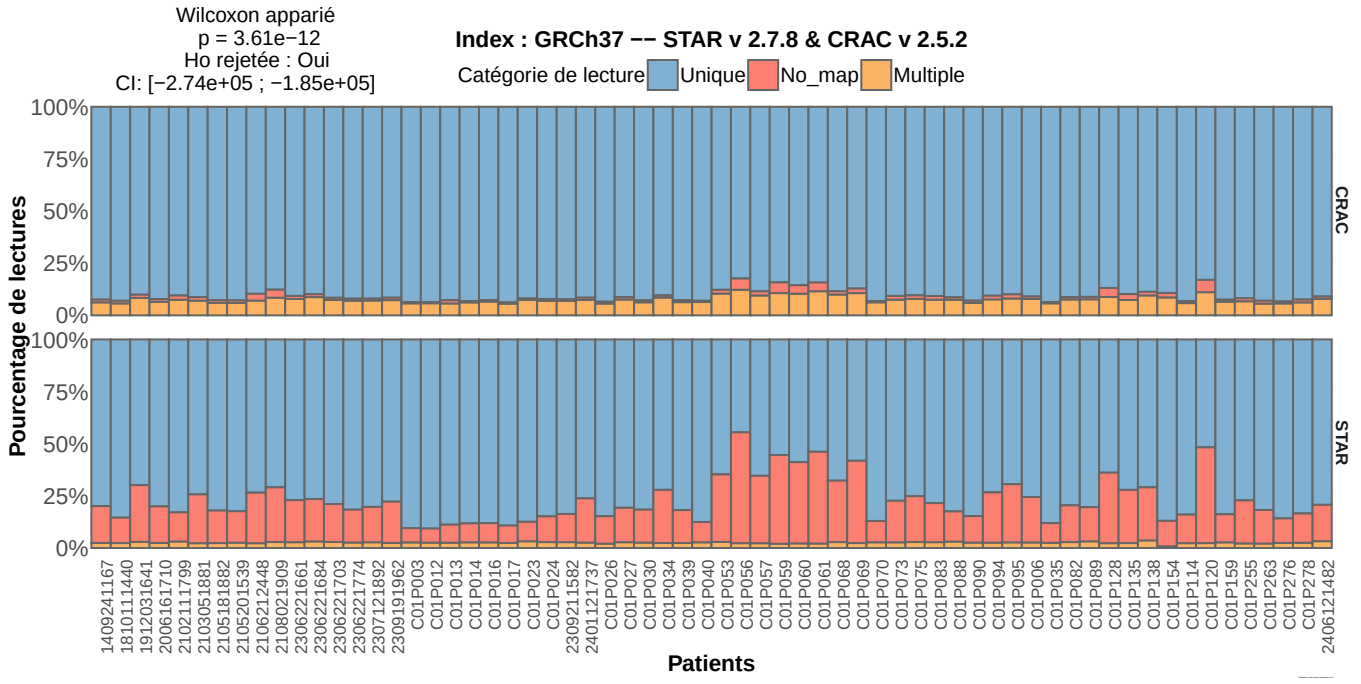

Figure 4 : Comparaison des profils d'alignement par patient entre STAR et CRAC

Soulignons ces résultats ne reflètent pas nécessairement une différence de qualité dans l'alignement entre `STAR` et `CRAC`. Comme évoqué précédemment, les stratégies algorithmiques diffèrent, notamment dans la manière de classifier les lectures uniques, multiples, et enfin celles non mappées. Ces divergences peuvent influencer la sensibilité de détection à certains types de variants.

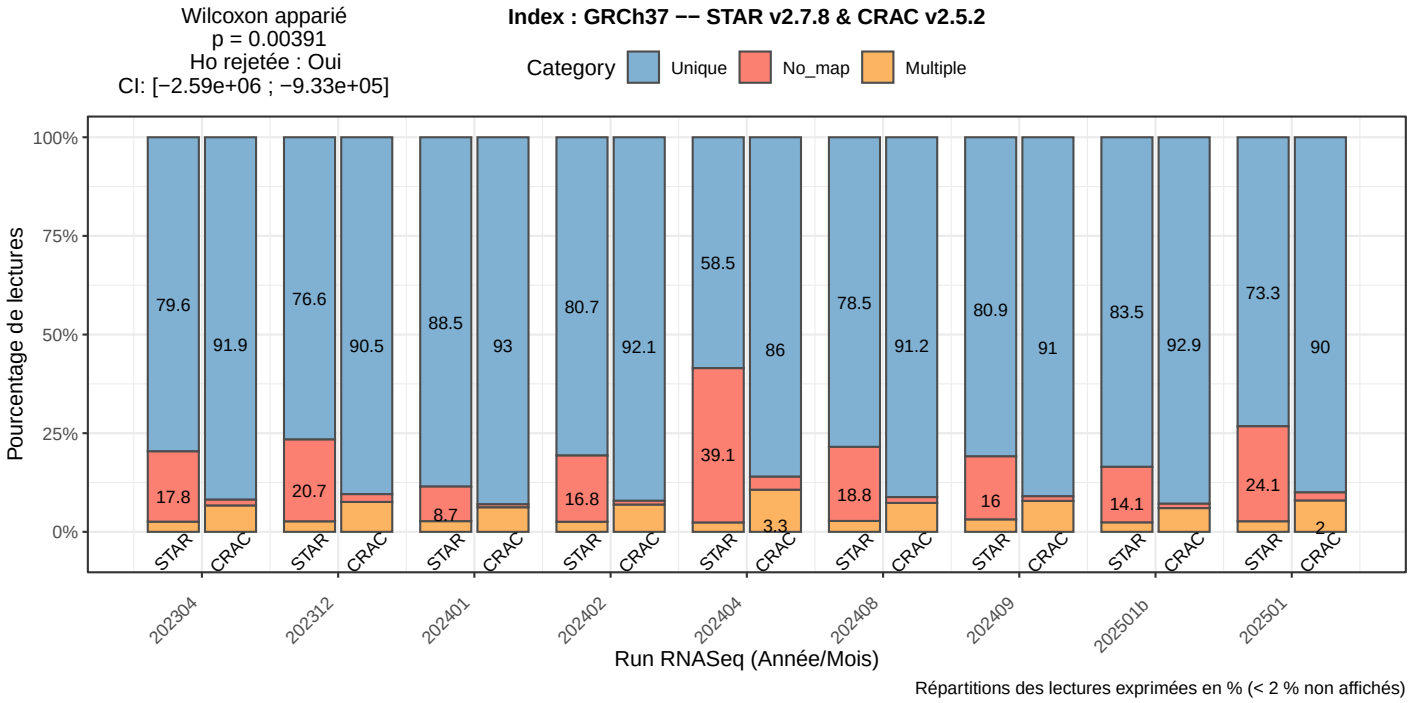

Figure 5 : Profils d'alignements par run entre STAR et CRAC

L'analyse statistique par test de Wilcoxon apparié révèle une différence significative entre le nombre de lectures uniques alignées par STAR et CRAC ($p = 0{,}00391$). L'intervalle de confiance négatif $[-2{,}59 \times 10^6; -9{,}33 \times 10^5]$ nous indique que `CRAC` (dans le paramétrage par défaut et pour notre cas d'étude précis) génère de manière systématique un nombre de lectures alignées de manière unique significativement plus élevé que `STAR` pour un même échantillon. Ce résultat suggère soit une sensibilité accrue, soit une stratégie d'alignement plus permissive. Par conséquent, même si la proportion de lectures uniques est globalement comparable entre les deux outils (conclusion que l'on peut admettre autant à l'échelle du patient (**figure 4**) qu'a l'échelle du *run* RNASeq **figure 5**. Avant d'éffectuer des analyses de quantification, j'ai mené des explorations complémentaires en extrayant la métrique nommée « NH » (*Number of Hits*), cette information est un champ dans le fichier de sortie au format `SAM`. Pour ce faire, j'ai développé un script Bash (cf. **Annexe à ajouter**). L'intérêt ici est d'essayer de comprendre la différence de lectures uniques entre les deux outils.

3.2 Analyse des données de comptages avec normalisation TPM

3.3 Analyse des données de comptages avec normalisation TMM

Le test évalue si, pour un gène donné, l'expression normalisée (CPM_TMM) est stitistiquement différente entre les conditions (ex : patients vs contrôles). Une p-value faible (<0.05 généralement) indique que la différence observée est peu probable due au hasard, donc potentiellement biologiquement significative. Ce test est robuste face aux distributions non normales, ce qui est fréquent en RNA-seq. Cependant, ce test ne donne pas d'indication sur la taille de l'effet (seulement s'il y a une différence significative). Sur 56 tests (un par gène), il faut ajuster la p-value pour réduire les faux

positifs.

Modalités de calcul du test Wilcoxon (Mann-Whitney) Classe toutes les valeurs des deux groupes combinés (conditions) par ordre croissant. Attribue un rang à chaque valeur. Calcule la somme des rangs pour un groupe. Compare cette somme avec la distribution attendue si les deux groupes venaient de la même population. La p-value correspond à la probabilité d'obtenir une somme de rangs aussi extrême (ou plus) sous l'hypothèse nulle (pas de différence entre conditions). Graphique j'affiche la distribution des CPM normalisés par gène, par condition, avec des points colorés. L'annotation avec label permet d'afficher la p-value pour chaque gène, informant visuellement sur la significativité des différences observées. Cela m'aide à relier visuellement l'intensité d'expression avec la signification statistique.

L'absence de différences statistiquement significatives ($p > 0.05$) dans les niveaux d'expression normalisés (CPM TMM) des 56 gènes d'intérêt entre les conditions étudiées suggère que, au sein des échantillons analysés, ces gènes ne présentent pas de variations d'expression marquées ou robustes en lien avec le facteur expérimental considéré. Cela peut indiquer que, dans ce contexte biologique précis, la modulation transcriptionnelle de ces gènes n'est pas significativement impactée par la condition testée, ou que les effets sont trop faibles pour être détectés avec la taille d'échantillon actuelle. Il est aussi possible que la variabilité inter-individuelle ou technique masque de potentielles différences subtiles. Enfin, ces résultats invitent à considérer d'autres niveaux de régulation (post-transcriptionnel, épigénétique) ou des approches complémentaires (ex. analyses multi-omics, échantillonnages plus larges) pour mieux comprendre les mécanismes biologiques sous-jacents dans cette pathologie ciblée.

4 Discussion

5 Annexes

5.1 Structures des fichiers de log pour **STAR** et **Crac**

```
mickael@zazie:~/M1_Stage/M1_ALS_RnaSeq/4.LOG_STAR_HG37$ cat *P059*
                                 Started job on |       May 01 22:46:43
                             Started mapping on |       May 01 22:46:54
                                    Finished on |       May 01 22:47:45
       Mapping speed, Million of reads per hour |       125.49

                           Number of input reads |       1777828
                       Average input read length |       301
                                    UNIQUE READS:
                   Uniquely mapped reads number |       985128
                        Uniquely mapped reads % |       55.41%
                          Average mapped length |       266.09
                       Number of splices: Total |       593665
            Number of splices: Annotated (sjdb) |       0
                       Number of splices: GT/AG |       574949
                       Number of splices: GC/AG |       16805
                       Number of splices: AT/AC |       63
               Number of splices: Non-canonical |       1848
                      Mismatch rate per base, % |       0.55%
                         Deletion rate per base |       0.01%
                        Deletion average length |       1.72
                        Insertion rate per base |       0.02%
                       Insertion average length |       1.60
                             MULTI-MAPPING READS:
         Number of reads mapped to multiple loci |       36282
              % of reads mapped to multiple loci |       2.04%
         Number of reads mapped to too many loci |       220
              % of reads mapped to too many loci |       0.01%
                                  UNMAPPED READS:
  Number of reads unmapped: too many mismatches |       0
       % of reads unmapped: too many mismatches |       0.00%
            Number of reads unmapped: too short |       755734
                 % of reads unmapped: too short |       42.51%
                Number of reads unmapped: other |       464
                     % of reads unmapped: other |       0.03%
                                  CHIMERIC READS:
                       Number of chimeric reads |       0
                            % of chimeric reads |       0.00%
```

log d'alignement avec STAR

```
for a strand specific library.

Time to index all reads: 0 s
Time to classify all reads: 172.027 s

---------------------------------
           Some STATISTICS

---------------------------------
Total number of reads analyzed: 3555656

Single: 2994747 (84.2249%)
Multiple: 125264 (3.52295%)
None: 181345 (5.10018%)
Duplication: 254300 (7.15199%)

---------------------------------
Explainable: 3288037 (92.4734%)

Repetition: 116475 (3.27577%)
Normal: 1601 (0.0450269%)
Almost-Normal: 0 (0%)
Sequence-Errors: 939992 (26.4365%)
SNV: 524161 (14.7416%)
Short-Indel: 17841 (0.501764%)
Splice: 500810 (14.0849%)
Weak-Splice: 1369 (0.038502%)
Chimera: 49298 (1.38647%)
Paired-end Chimera: 129569 (3.64403%)
Bio-Undetermined: 2872021 (80.7733%)
Undetermined: 207549 (5.83715%)
```

summary généré par CRAC

Figure 6 : Comparaison visuelle des statistiques d'alignement STAR et CRAC.

5.2 Modification du pipeline Snakemake

```
rule aggregateStar:
   input:
      logs = expand(OUTPUT + "/STAR/results/{sample}/Log.final.out",
         sample=SAMPLES)
   output:
      csv = OUTPUT + "/STAR/Stats_Log_star.csv"
   shell:
   r"""
   # Headers generation :
   echo "Run,Patient,Type,STAR_Date_Mapping,STAR_Total_reads,\
   STAR_Unique_reads,STAR_Unique_pct,\
   STAR_Multi_reads,STAR_Multi_pct,\
   STAR_No_map_reads,STAR_No_map_pct_sum,STAR_No_map_pct_mismatch,\
   STAR_No_map_pct_tooshort,\
   STAR_No_map_pct_other,\
   STAR_Avg_read_len" > {output.csv}

   # Metrics extraction
   for logfile in {input.logs}; do
      basename=$(basename "$logfile")
      sample=$(echo "$basename" | sed -E 's/.+\/?([^\/]+)_Log.final.out/\1/')
      Run=$(echo "$sample" | cut -d '-' -f1)
      Patient=$(echo "$sample" | cut -d '-' -f2)
      Type=$(echo "$sample" | cut -d '-' -f3 | cut -d '.' -f1)

      Ext() {
         grep "$1" "$logfile" | cut -d '|' -f2 | tr -d ' ' | tr -d '%' || echo
            "0"
      }
      STAR_Date_Mapping=$(Ext "Finished on")
      STAR_Total_reads=$(Ext "Number of input reads")
      STAR_Unique_reads=$(Ext "Uniquely mapped reads number")
      STAR_Unique_pct=$(Ext "Uniquely mapped reads %")
      ...
      echo "$Run,$Patient,$Type,$STAR_Date_Mapping,$STAR_Total_reads,\
      $STAR_Unique_reads,$STAR_Unique_pct,\
      $STAR_Multi_reads,$STAR_Multi_pct,\
      $STAR_No_map_reads,$STAR_No_map_pct_sum,$STAR_No_map_pct_mismatch,\
      $STAR_No_map_pct_tooshort,$STAR_No_map_pct_other,\
      $STAR_Avg_read_len" >> {output.csv}
   done
   """
```

Code 6 : Règle Snakemake `aggregateStar` pour l'extraction des statistiques STAR

Glossaire

autophagie L'autophagie est un processus cellulaire de recyclage qui dégrade et élimine les composants cellulaires endommagés, contribuant à l'homéostasie et à la protection contre le stress cellulaire. p. 5

haploinsuffisance Incapacité d'une seule copie fonctionnelle d'un gène à produire une quantité suffisante de produit génique (ARN ou protéine) pour assurer une fonction biologique normale, entraînant ainsi un phénotype pathologique. p. 5

homéostasie Ensemble des mécanismes qui régulent la production, la maturation, le transport et la dégradation pour maintenir un équilibre fonctionnel dans la cellule. p. 5

prévalence Proportion d'individus dans une population donnée présentant une caractéristique (généralement une maladie) à un instant donné ou sur une période donnée. p. 4

séquençage pairé Méthode de séquençage où les deux extrémités d'un fragment d'ADN sont séquencées indépendamment, permettant d'obtenir deux lectures (lectures appariées) qui facilitent l'alignement et la détection des variants, notamment dans les régions complexes ou répétées. p. 8

Transport axonal Le transport axonal permet le déplacement des organites, protéines et ARN le long de l'axone en assurant la communication et la survie neuronale sur de longue distance. p. 5

6 Bibliographie compilée

[1] Sèverine Bérard. *La Transformée de Burrows-Wheeler (BWT)*. French. Présentation pédagogique de l'algorithme BWT. 2025. URL : https://moodle.umontpellier.fr/pluginfile.php/932659/mod_resource/content/3/BWT.pdf (visité le 28/04/2025).

[2] Sèverine Bérard. *Table des Suffixes — Suffix Array*. French. Support de cours sur les tableaux de suffixes. 2025. URL : https://moodle.umontpellier.fr/pluginfile.php/636401/mod_resource/content/1/TableSuffixes.pdf (visité le 02/04/2025).

[3] C. Wolfson et al. "Global Prevalence and Incidence of Amyotrophic Lateral Sclerosis : A Systematic Review". In : *Neurology* 101.6 (août 2023). Revue systématique des données mondiales sur la SLA. DOI : 10.1212/WNL.0000000000207474. URL : https://www.neurology.org/doi/10.1212/WNL.0000000000207474 (visité le 13/11/2024).

[4] Centre Constitutif SLA de Tours. *Protocole National de Diagnostic et de Soins (PNDS) – Filière FILSLAN*. Argumentaire scientifique. Document officiel du PNDS pour la SLA, version génétique. Centre de Référence SLA, TOURS, nov. 2020, p. 6. URL : https://www.has-sante.fr/upload/docs/application/pdf/2021-12/pnds_argumentaire_sla_genetique_2020.final.pdf (visité le 28/03/2024).

[5] Peter J.A. Cock et al. "The Sanger FASTQ File Format for Sequences with Quality Scores, and the Solexa/Illumina FASTQ Variants". In : *Nucleic Acids Research* 38.6 (déc. 2009). Définition du format FASTQ Sanger et variantes, encodage qualité Phred en ASCII, p. 1767-1771. DOI : 10.1093/nar/gkp1137.

[6] Illumina, Inc. *BAM File Format*. Description du format binaire compressé (BGZF), indexation, et usage en RNA-seq. Illumina. 2025. URL : https://support.illumina.com/help/BS_App_RNASeq_Alignment_OLH_1000000006112/Content/Source/Informatics/BAM-Format.htm (visité le 11/06/2025).

[7] Johannes Köster. *Snakemake Tutorial (Official Documentation)*. Tutoriel pour apprendre à utiliser Snakemake (workflow RNA-seq, règles, dépendances, environnement). 2025. URL : https://snakemake.readthedocs.io/en/stable/tutorial/tutorial.html (visité le 17/06/2025).

[8] Alban Mancheron. "Extraction de motifs communs dans un ensemble de séquences : Application à l'identification de sites de liaison aux protéines dans les séquences primaires d'ADN". Thèse de doctorat. Nantes, France : Université de Nantes, École Doctorale STIM, juin 2006, p. 162. URL : https://theses.fr/2006NANT2060 (visité le 14/01/2025).

[9] Orphanet. *Prévalence des Maladies Rares : Données Bibliographiques Classées par Fréquence Décroissante*. Argumentaire scientifique. Synthèse statistique de la prévalence des maladies rares. Orphanet – INSERM, US14, nov. 2023, p. 14. URL : https://www.orpha.net/

pdfs/orphacom/cahiers/docs/FR/Prevalence_des_maladies_rares_par_prevalence_decroissante_ou_cas.pdf (visité le 28/03/2024).

[10] SAM/BAM Format Specification Working Group. *SAM/BAM Format Specification (v1.6)*. Technical Report. Définit la structure du format SAM/BAM v1.6, champs obligatoires et options. Global Alliance for Genomics et Health / Samtools, nov. 2024. URL : https://samtools.github.io/hts-specs/SAMv1.pdf (visité le 11/06/2025).

[11] Samuel S. Shapiro et Martin B. Wilk. "An Analysis of Variance Test for Normality (Complete Samples)". In : *Biometrika* 52.3–4 (1965). Test de normalité basé sur l'analyse de la variance, p. 591-611.

[12] Antoine Soetewey. *What Statistical Test Should I Do ?* Blog post on Stats and R. Guide pratique d'aide au choix du test statistique. Déc. 2021. URL : https://statsandr.com/blog/what-statistical-test-should-i-do/ (visité le 11/06/2025).

[13] Posit Team. *DGEList Function — RDocumentation*. Documentation de la fonction `DGEList()` dans edgeR (v3.14.0). 2024. URL : https://www.rdocumentation.org/packages/edgeR/versions/3.14.0/topics/DGEList (visité le 30/08/2024).

[14] Günter P. Wagner, Koryu Kin et Vincent J. Lynch. "Measurement of mRNA Abundance Using RNA-seq Data : RPKM Measure Is Inconsistent Among Samples". In : *Theory in Biosciences* 131.4 (déc. 2012). Critique de la méthode RPKM ; justification de l'usage du TPM pour la comparabilité, p. 281-285. DOI : 10.1007/s12064-012-0162-3. URL : https://doi.org/10.1007/s12064-012-0162-3 (visité le 30/08/2024).

[15] Frank Wilcoxon. "Individual Comparisons by Ranking Methods". In : *Biometrics Bulletin* 1.6 (déc. 1945). Introduction du test des rangs signés pour échantillons appariés, p. 80-83.