# Epissage alternatif et homéostasie des métaux chez les plantes : analyse de données RNA-Seq

**Auteur :** Lolos, Colin
**Promoteur(s) :** Hanikenne, Marc
**Faculté :** Faculté des Sciences
**Diplôme :** Master en bioinformatique et modélisation, à finalité approfondie
**Année académique :** 2020-2021
**URI/URL :** http://hdl.handle.net/2268.2/12555

Université de Liège
Département des Sciences de la Vie
Faculté des Sciences
Août 2021

# Epissage alternatif et homéostasie des métaux chez les plantes : analyse de données RNA-Seq

**Colin Lolos**

Mémoire réalisé en vue de l'obtention du grade de
Master en Bioinformatique et Modélisation

Promoteur : **Marc Hanikenne**
Co-promoteur : **Patrick Motte**
Encadrant : **Steven Fanara**

Laboratoire de Génomique Fonctionnelle et Imagerie Moléculaire Végétale, UR InBioS

# Remerciements

Je tiens à sincèrement remercier tous les intervenants qui ont contribué à l'élaboration de ce travail de fin d'études.

En premier lieu, je remercie Monsieur Hanikenne, le promoteur de ce mémoire, pour l'aide et la direction qu'il m'a fournies tout au long du processus, sans oublier sa disponibilité et sa bonne humeur. Ses remarques m'ont permis d'aborder et de décomposer les résultats sous divers aspects tels que la pertinence et la représentation biologique, tout en encourageant mon propre cheminement scientifique.

Je remercie également Steven Fanara, mon encadrant, pour le partage généreux de ses connaissances, ses justifications précises et pointilleuses, le temps qu'il m'a volontairement consacré et pour la bienveillance dont il a fait preuve à mon égard au cours de cette année.

Je suis également reconnaissant envers Roxane Gilson et Pauline Stévenne, membres du laboratoire, pour leur aide en termes de code et de rédaction, de justification et pour les pistes de réflexion qu'elles m'ont apportées.

Je souhaite remercier le laboratoire de Génomique Fonctionnelle et Imagerie Moléculaire Végétale de façon générale, pour leur accueil, l'atmosphère positive qui y règne et le temps alloué aux présentations continues de mes avancées.

Je remercie l'ensemble des professeurs du Master en Bioinformatique et Modélisation pour leurs apports respectifs dans les différentes matières qu'ils dispensent ; ces enseignements ont permis la réalisation de ce travail tel qu'il est présenté aujourd'hui.

Finalement, et pas des moindres, je remercie chaleureusement ma famille pour leur soutien moral et leur disposition sans quoi la complétion de ce travail n'aurait pas été possible. Au même titre, je remercie mes amis pour leurs encouragements et pour les fenêtres d'évasion qu'ils ont constituées.

Sincèrement, merci à tous.

Colin Lolos
Promoteur : **Marc Hanikenne**
Co-promoteur : **Patrick Motte**
Encadrant : **Steven Fanara**

Laboratoire de Génomique Fonctionnelle et
Imagerie Moléculaire Végétale
Août 2021

# Résumé

Des études récentes menées sur les mécanismes d'épissage alternatif chez *Arabidopsis thaliana* suggèrent un lien entre un facteur d'épissage appelé SR45 et certains gènes impliqués dans l'homéostasie du fer [1],[2]. En se liant à des séquences particulières de l'ARN pré-messager, SR45 reconnaît ses cibles et permet le recrutement d'une machinerie d'épissage, le spliceosome, afin d'assurer le retrait des introns et la jonction des exons, des évènements qui caractérisent le phénomène d'épissage.

Sur base de la relation putative entre ce facteur et les voies de régulation du fer, une étude de l'épissage alternatif a été conduite sur des individus sauvages et mutants *sr45-1* pour différentes conditions en fer.

Parallèlement à l'avènement des séquenceurs à haut débit, les outils d'analyse ont évolué et les méthodologies se sont multipliées. Aujourd'hui, les protocoles d'étude de l'épissage alternatif de données RNA-seq sont légion - ceci s'explique notamment par l'abondance de programmes qui peuvent intervenir à une étape donnée des analyses, leur intercompatibilité ou encore la question biologique adressée.

L'objectif de ce mémoire comprend la conception d'un pipeline adapté à l'étude de l'épissage alternatif différentiel chez *Arabidopsis thaliana* et son application aux données expérimentales.

La première étape consiste à détecter les nouveaux variants et transcrits issus des génotypes et conditions étudiés et d'enrichir les annotations existantes. Les données relatives à seize individus *A. thaliana* pour des génotypes sauvages ou mutants (*sr45-1*) et des conditions contrôle ou carence en fer, avec quatre réplicats biologiques, ont été séquencées et alignées sur un génome de référence. Les transcriptomes des différents échantillons ont été reconstruits et assemblés par groupe de réplicats.
Les transcrits extraits ont été comparés avec des annotations de référence. Les nouvelles annotations ont été filtrées et les nouveaux isoformes ou transcrits potentiels ont été ajoutés aux annotations. Un nouvel alignement des échantillons a été réalisé avec les annotations enrichies. Sur base de ces résultats, les transcrits ont été quantifiés et les analyses des gènes et transcrits différentiellement exprimés ont été menées.
Conjointement, l'analyse de l'épissage alternatif différentiel a été conduite entre les différents génotypes et conditions sur base des alignements des échantillons.

# Table des matières

# 1 Introduction

## 1.1 L'épissage alternatif

Les plantes ont conservé au cours de l'évolution une plasticité remarquable en termes d'adaptation aux conditions de leur milieu. La pression de conservation autour de cette faculté s'explique entre autres par leur caractère sessile et leur besoin de pouvoir fournir une réponse fine et modulable aux stress environnementaux (carences ou excès nutritifs, seuil critique de chaleur, ... ) [3].

D'abord considéré comme contributeur mineur à cette capacité, l'épissage est aujourd'hui reconnu comme un processus essentiel impliqué dans de nombreuses voies métaboliques et ce, chez tous les eucaryotes.
A l'origine, l'information génétique est encodée au niveau des génomes. Ces séquences de bases nucléotidiques appelées ADN sont transcrites en ARN messagers pré-matures (pré-ARNm) constitués d'introns et d'exons. Ces molécules médient l'information génétique. Les introns sont par définition non codants et sont excisés du reste de l'information dite codante. L'épissage correspond au processus d'excision des introns et de ligature des exons qui aboutit à la formation des ARN messagers (ARNm) matures. Ces derniers sont lus et traduits en chaînes d'acides aminés, les protéines, par le biais des ribosomes. L'information initiale nucléotidique est donc à terme convertie en chaînes d'acides aminés, et la composition de ces chaînes conditionne le rôle métabolique des protéines qu'elles constituent au sein de l'organisme [4].

Cependant, l'épissage n'est pas un phénomène constitutif - les introns ne sont pas systématiquement clivés, conduisant à des évènements d'épissage dits alternatifs. De plus, ces derniers ne se limitent pas à la rétention d'intron mais englobent différentes classes d'évènements. Il existe cinq classes majeures d'évènements d'épissage alternatif : le saut d'exon, l'exclusion mutuelle d'exons, l'épissage alternatif en 5' ou en 3' et la rétention d'intron. Ce dernier évènement est prépondérant chez les plantes supérieures - il est estimé qu'entre 60% et 70% des gènes qui contiennent des introns sont sujets à l'épissage alternatif [5],[6], ce qui affirme le rôle clé de ce processus au niveau de la régulation de l'expression des gènes.

Le champ d'action de l'épissage alternatif s'étend à la localisation des ARNm, leur stabilité et traduction ainsi que la modification de leur cadre de lecture. Au cours de ce processus post-transcriptionnel, l'action de facteurs d'épissage au niveau de leurs cibles pré-ARNm peut ainsi conduire à la modification de la séquence des ARN matures, engendrant par épissage alternatif la genèse de plusieurs séquences à partir d'un même gène. En conséquence, grâce à l'épissage alternatif, un même gène peut conduire à la production de différentes protéines ce qui permet la régulation de mécanismes qui régissent leur métabolisme. Parmi ces voies métaboliques, celles des métaux incluant le fer, le cuivre, le manganèse ou encore le zinc sont essentielles à la prospérité des végétaux puisqu'elles interviennent dans des processus tels que la respiration, la photosynthèse ou la synthèse d'ADN [7].

Globalement, environ 95% et 60% des gènes composés de plusieurs exons chez l'Homme et *A. thaliana* , respectivement, sont sujets à l'épissage alternatif, ce qui affirme le rôle clé de ce processus au niveau de la régulation de l'expression des gènes [8].

## 1.2 Le facteur d'épissage SR45 et l'homéostasie des métaux

SR45 appartient à la grande famille des facteurs d'épissage riches en résidus sérines/arginines (resp. S,R), hautement conservée chez les eucaryotes multicellulaires [9]. Ces facteurs participent à l'assemblage du splicéosome via des interactions protéines-protéines et protéines-ARN.
La composition des facteurs SR est modulable, mais se caractérise toujours par la présence d'au moins un domaine RRM (*RNA Recognition Motif*) qui conditionne la spécificité de reconnaissance aux ARN et d'un domaine RS qui intervient dans les interactions en tant que telles.
La mutation perte-de-fonction pour le facteur SR45 chez *A. thaliana* entraîne les observations phénotypiques suivantes : des retards de développement racinaire, une floraison tardive, des siliques plus courtes contenant moins de graines, des feuilles et pétales plus étroits et un nombre inhabituel d'organes floraux [10]. De plus, les phénotypes mutants *sr45-1* sont responsables d'altérations au niveau de la mobilisation des métaux tels que le fer, ainsi que de leur localisation et transport [2]. Fonctionnellement, SR45 est un régulateur d'épissage qui impacte notamment les voies de signalisation de l'ABA (Acide Abscissique) et de la défense des plantes [1],[11]. De plus, plusieurs protéines SR, qui interagissent notamment avec SR45, sont connues pour intervenir dans la biogenèse de miRNAs [2],[12],[13]. Dans cette mesure, des précautions particulières ont été prises dans le but d'inclure les miRNAs détectés dans les analyses.

## 1.3 Les objectifs

Ce mémoire a pour but d'établir un protocole d'analyse RNA-Seq qui permet l'étude de l'épissage alternatif différentiel (*Differential Alternative Splicing*, DAS) dans des individus *A. thaliana* sauvages et mutants pour SR45, pour des conditions en fer contrôles ou carencées.

Le protocole à établir doit respecter plusieurs contraintes afin de conforter sa pertinence : (i) il doit se composer d'outils récents afin de garantir une intégration durable, efficaces en termes de significativité et de mesure d'incertitude mais également en termes de vitesse computationnelle ; (ii) les programmes utilisés doivent être supportés favorablement par la littérature ; et finalement (iii) le protocole dans son ensemble doit être flexible et applicable à d'autres données expérimentales.

## 1.4 Les approches bioinformatiques / L'analyse RNA-seq

La méthodologie employée repose en premier lieu sur le séquençage des ARN messagers ou RNA-Seq des individus étudiés. Au cours de ce processus, les échantillons sont soumis à plusieurs étapes : les ARNm sont purifiés, fragmentés et des morceaux de leur séquence en bases sont déterminés, les reads. Ces derniers sont enregistrés dans des fichiers qui permettront leur analyse numérique, au format fastq.
Ensuite, le protocole est constitué de grandes étapes typiques des analyses RNA-seq : l'alignement des reads

séquencés sur un génome de référence, la quantification des reads alignés et les analyses d'expression [14]. L'analyse DAS, quant à elle, est plus spécifique : elle consiste à comparer les évènements d'épissage survenus entre différents groupes de test. Ce cadre d'analyse d'un mutant pour un facteur d'épissage implique une étape préliminaire additionnelle - l'enrichissement des annotations pour les nouveaux variants.
En effet, la mutation perte de fonction *sr45-1* ainsi que la condition de carence en fer sont toutes deux sujettes à perturber les processus d'épissage alternatif tels que rencontrés pour des organismes sauvages exposés à des conditions "normales". Dans l'affirmative, ces perturbations se répercutent au niveau des transcrits exprimés pour lesquels les évènements d'épissage deviennent extraordinaires, ce qui conduit à la production de nouveaux transcrits non répertoriés dans les bases de données d'annotation d'*A. thaliana* .
Les conclusions biologiques qui pourront être formulées au terme de cette étude sont donc tributaires de la détection de ces nouveaux variants.

### 1.4.1 Séquençage Illumina

La technologie Illumina est la méthode de séquençage de nouvelle génération la plus populaire.
Elle repose sur l'amplification clonale, sous forme de clusters, et sur la détection de synthèse par fluorescence. De plus, divers kits existent qui permettent le séquençage d'ADN, d'ARN ou de sous-espèces caractéristiques telles que les ARNm.

La première étape est la préparation des bibliothèques - dans le cadre d'un séquençage d'ARNm, les sous-classes d'ARN sont discriminées suivant la présence d'une queue poly-adénines, typique des messagers. Des billes magnétiques poly-thymines complémentent les ARNm des échantillons au niveau de la queue poly(A) et permettent leur purification. Le matériel est ensuite fragmenté chimiquement ou via sonication en morceaux de l'ordre de 200 pb appelés inserts. Les inserts sont convertis en ADNc, leur équivalent complémentaire en ADN, le second brin est synthétisé et des adaptateurs sont ligaturés aux extrémités 5' et 3'. Cette dernière structure se dénomme fragment.
Certains kits offrent la possibilité de conserver l'information relative à l'orientation du brin. En effet, l'incorporation du dUTP, un agent naturellement absent de l'ADN et éliminé avant le séquençage, joue le rôle de témoin afin de cibler le brin originel.

Les fragments sont ensuite attachés à une *flowcell* par l'intermédiaire d'oligonucléotides qui recouvrent le support et qui permettent leur maintient. L'amplification par ponts consiste à singulariser les brins, favoriser leur liaison aux oligonucléotides vacants sur la *flowcell* et de synthétiser le brin complémentaire. Ces réactions sont réalisées en cycle et aboutissent à l'édification de clusters qui regroupent des clones de fragments d'ADNc.

Au terme de l'amplification clonale, les brins complémentaires sont évacués et le séquençage prend place. Les polymérases s'apparient aux primers des adaptateurs et incorporent des bases à fluorescence spécifique qui permet leur identification. Chaque étape d'incorporation de base est suivie de son identification, libération de son extrémité 3' et d'une nouvelle incorporation en aval. Le nombre de cycles détermine la longueur des fragments séquencés [51],[15]. Le séquençage est dit *single-end* lorsque les fragments sont séquencés par une seule de leurs extrémités

et *paired-end* lorsque les deux extrémités sont lues par les polymérases.

### 1.4.2 Alignement des reads

Au terme du séquençage, les reads sont numérisés sous forme de fichiers FASTQ et prêts à être manipulés. La collection de séquences générée est dépourvue d'information quant aux positions respectives des reads sur le génome, si bien que déterminer des mesures d'expression pour des gènes spécifiques est impossible - l'information doit être extraite. Si l'organisme étudié dispose d'un génome ou d'un transcriptome de référence publié, les reads peuvent être superposés sur celui-ci afin de déterminer à quelle portion du génome ils correspondent : c'est l'alignement. Dans le cas où une telle référence n'est pas disponible, typiquement en présence d'un organisme peu étudié, le génome ou transcriptome peut être reconstruit sur base des simples reads séquencés : c'est la méthode d'assemblage *de novo*.

L'alignement peut se réaliser sur une référence, ou sur base d'un assemblage. En termes de différences, l'alignement requiert moins de temps et de ressources à l'exécution et ses résultats sont relativement moins dépendants de la taille des reads séquencés. Les outils d'alignement sur base d'un génome de référence se divisent selon le critère de sensiblité aux jonctions d'épissage (SJ). Ces jonctions se situent aux extrémités des introns et sont identifiées par les quelques nucléotides qui composent leurs extrémités.

Les aligneurs sensibles aux jonctions d'épissage (*splice-aware*) réalisent l'alignement tout en identifiant ces sites, qu'ils soient préalablement annotés ou nouveaux. En présence de ces jonctions, les reads sont scindés et alignés au génome de façon interrompue laissant place à de potentiels introns, afin de déterminer si le read est issu d'une modification par épissage. Une telle stratégie d'alignement permet la détection de nouveaux transcrits et de nouvelles isoformes, absents des fichiers d'annotations. Cependant, cette approche n'est adaptée qu'aux organismes pour lesquels un génome de référence de qualité suffisante est disponible [51].

Dans une optique de détection des nouveaux sites et variants d'épissage, certains aligneurs proposent une stratégie d'alignement en deux temps, le *2-pass*. Un premier alignement est réalisé de façon classique. Les jonctions d'épissage détectées lors de cette première étape sont renseignées à l'aligneur en tant qu'annotations lors d'un deuxième alignement.

De nombreux aligneurs sensibles aux jonctions d'épissage ont été développés par la communauté scientifique (par ex. TopHat, HiSat2). Parmi eux, STAR est un aligneur splice-aware populaire. D'après des études comparatives, sa vitesse d'exécution est réduite d'un facteur 50 vis-à-vis des autres outils d'alignement [16] tout en présentant une haute sensibilité de détection avec un FDR (*False Discovery Rate*) minimal pour des génomes humains [17] mais également chez *A. thaliana* [18]. De plus, STAR offre la possibilité de réaliser des alignements doubles, par échantillon ou pour plusieurs échantillons, et incorpore une option de quantification des gènes propre. Ces deux premières options sont particulièrement intéressantes pour la recherche et la détection de nouveaux variants, en augmentant respectivement le nombre de reads qui supportent les jonctions d'épissage [16] et le nombre de jonctions détectées. Le bénéfice est d'autant plus conséquent pour les échantillons dont les reads sont de tailles réduites, permettant leur alignement sur les jonctions en impliquant moins de nucléotides [17].

L'algorithme d'alignement de STAR fonctionne en deux étapes : la détermination d'une graine (*seed*) suivie d'une phase de regroupement et d'attribution de scores pour ces seeds.

Lors de la première étape, STAR détermine le Préfixe Maximal Mappable (MMP), soit la longueur maximale d'une sous-séquence d'un read donné qui s'aligne sur la séquence génomique de référence et conserve l'information sous la forme de tableau de suffixes non compressés (*Suffix Array*, SA). Si le read considéré correspond à une séquence incluant une jonction d'épissage, l'alignement ne peut être contigu et le MMP est calculé pour chaque extrémité de la jonction en question, produisant jusqu'à plusieurs seeds par read. STAR calcule alors chaque seed en ne considérant que les portions non-alignées, ce qui réduit considérablement le temps de recherche. Cette méthodologie a l'avantage de capturer tous les alignements potentiels pour chacun des reads sans induire de coût computationnel conséquent, facilitant la détection d'alignements multiples. Les valeurs de MMP sont calculées pour les deux orientations des reads [16].

La deuxième étape traite la collection de seeds obtenue et reconstruit les alignements des reads qui comportent plusieurs seeds. La jointure est d'abord réalisée par proximité inter-reads, puis selon un système de scores. Ces derniers sont calculés comme la somme des bases effectivement alignées pénalisés par les discordances (indels, espaces). Les portions alignées sont rassemblées sur base d'un modèle qui suppose une transcription linéaire des unités, soit que les différents bloques continus qui composent un read doivent se suivre dans le génome et ne peuvent pas se superposer [16].

### 1.4.3 Détection et sélection des nouveaux variants

La détection de nouvelles isoformes ou variants d'épissage se construit généralement en trois étapes. La méthodologie adoptée dépend des dispositions initiales telles que l'accès à un génome de référence. Dans le cas contraire, le protocole nécessite l'utilisation d'un transcriptome ou encore un assemblage préliminaire [19]. La première étape consiste à aligner les reads sur une référence ou le résultat d'un assemblage tel que précédemment présenté.

Sur base de l'alignement, les groupes de reads qui désignent de potentiels transcrits sont déduits et rapportés ce qui constitue l'étape d'assemblage en transcriptome [20].

La troisième étape, la quantification, peut être réalisée en parallèle ou successivement à l'étape d'assemblage et permet la discrimination des candidats à filtrer sur base de leur valeur d'expression. Les nouvelles isoformes peuvent être detectées par intersection avec des annotations de référence.

A l'image de l'alignement, il existe également deux approches principales à l'assemblage en transcriptome - l'assemblage guidé par référence ou *de novo*. Une reconstruction guidée assemble les reads alignés en transcrits ou fragments en accord avec les annotations qui renseignent les positions, tailles et variants pour les transcrits déjà détectés. La stratégie *de novo* appuie son assemblage sur les seuls reads qui ont été séquencés et ne bénéficie donc pas d'informations complémentaires issues d'annotations, ce qui rend son application en pratique plus compliquée et imprécise notamment dû aux problèmes posés par la reconstruction de transcrits peu exprimés ou dont l'épissage est complexe [21],[22].

La quantification des transcrits est une caractéristique généralement proposée par les outils d'assemblage et s'exprime à travers les mêmes unités que l'expression des gènes ; FPKM (*Fragments Per Kilobase of exon model per Million mapped fragments*), TPM (*Transcripts Per million*) ou encore RPKM (*Reads Per Kilobase of transcript per Million reads mapped*) [23]. Ces valeurs traduisent les comptes des reads normalisés selon la longueur et la couverture du transcrit auquel ils se rapportent [24].

Cufflinks et StringTie sont les assembleurs de transcrits les plus rencontrés dans la littérature. Le dernier, plus récent, présente des résultats de reconstruction de plus haute sensibilité et précision [20].

StringTie offre la possibilité de réaliser un assemblage guidé par référence ou *de novo*, il permet la quantification des expressions relatives aux transcrits assemblés et dispose également d'une option de fusion de plusieurs transcriptomes, StringTie-merge. Cette étape est typiquement nécessaire en présence de réplicats biologiques afin de consolider le transcriptome commun.

TACO est un outil spécialement destiné à l'assemblage de plusieurs échantillons issus de reconstructions en transcriptome consensus. Similaire aux options de fusion proposées par les deux assembleurs Cufflinks et StringTie, les performances d'assemblage de Cuffmerge et StringTie-merge tendent à se détériorer à mesure que le nombre d'échantillons étudiés augmente et prédisent des transcrits issus de rétention d'introns aberrantes ou d'extrémités 5' et 3' anormalement longues, des résultats pouvant fausser les détections de nouveaux variants d'épissage [25].

L'algorithme d'assemblage TACO fusionne les collections de transcrits fournies et les trie sur base des coordonnées génomiques, les éléments qui se chevauchent étant groupés en loci. Les transcrits sont discriminés suivant leur brin d'origine et sont traités individuellement. Les expressions des transcrits sont calculées comme la somme des expressions relatives à chaque base contenue. Pour chaque cluster de transcrits, un graphe d'épissage est construit - les noeuds représentent les régions transcrites de façon contiguës exemptes de jonctions d'épissage, les exons typiquement. L'expression des différents noeuds est calculée comme la somme des expressions des transcrits qui le contiennent. Ces valeurs sont utilisées par un algorithme de détection de changement de profil d'expressions qui itère sur le graphe, aboutissant à la production d'un graphe de chemins qui encapsule les structures issues d'épissage. Finalement, un algorithme de programmation dynamique traverse ces chemins et extrait les isoformes les plus exprimées pour chaque gène [25]. Les nouveaux transcrits et isoformes peuvent ensuite enrichir les annotations de référence dans les analyses en aval.

### 1.4.4 Quantification des transcrits

Une fois l'alignement des reads réalisé, l'étape suivante du schéma classique des études RNA-Seq est la quantification. Cette procédure consiste à assigner les reads à des gènes ou transcrits afin d'en déterminer les comptes avant de procéder aux analyses d'expressions différentielles [26].

De façon générale, les algorithmes de quantification (*quantifiers*) nécessitent un fichier d'annotations de format GTF (*Gene Transfer Format*) ou GFF (*General Feature Format*) et les résultats d'alignement de format BAM (*Binary Alignment/Map*) ou SAM (*Sequence Alignment/Map*) [27],[28]. Les annotations inventorient les coor-

données précises des gènes, transcrits et autres éléments présents dans le génome sous la forme typique d'un fichier texte délimité par des tabulations. La structure du fichier se construit en neuf colonnes qui répertorient respectivement le nom de la séquence d'appartenance (chromosome), la source ou le programme qui a produit l'annotation, la nature de l'élément, la position de départ, de fin, le score qualité, l'orientation du brin, le cadre de lecture et une liste d'attributs [27].
SAM est l'extension générique des fichiers issus d'alignement qui conservent les positions de chaque read aligné sur le génome et des informations telles que les scores qualités associés et la méthode de séquençage utilisée. L'extension BAM est la forme compressée sous forme binaire [28],[29].

La quantification consiste à compter les séquences alignées sur les gènes ou transcrits recensées dans les annotations. Les quantifiers se divisent en deux catégories : basés sur l'alignement et sans alignement, également appelé pseudo-alignement [30].
La première alternative est la plus classique. Elle consiste à aligner chaque read sur un génome ou transcriptome de référence et d'estimer les comptes. Le pseudo-alignement est une approche plus récente qui est construite sur l'idée qu'un alignement exact n'est pas nécessaire pour déterminer l'origine des reads au bénéfice de la vitesse d'exécution.

RSEM et Salmon sont respectivement des représentants de ces deux catégories qui présentent des taux de précision et sensibilité supérieurs [14] et qui permettent la quantification au niveau des transcrits.
RSEM a la caractéristique qu'il n'exige pas de génome de référence mais peut à la place extraire les transcrits de référence sur base d'un génome. Le modèle statistique tient compte de la distribution des reads, des erreurs de séquençage et estime les maximums de vraisemblance des niveaux d'expression suivant un algorithme d'espérance - maximisation (*Expectation - Maximization*, EM) adapté à la gestion des reads à alignements multiples [31]. RSEM est naturellement compatible avec plusieurs aligneurs (tel que STAR) qui peuvent être combinés à sa procédure de quantification via ses commandes [32].

La quantification via Salmon se construit en trois étapes : un pseudo-alignement, une première phase qui détermine les niveaux d'expressions initiaux et les paramètres du modèle et une deuxième phase qui, comme RSEM, affine ces estimations par EM. Le modèle considère les biais spécifiques aux séquences, au contenu en GC (Guanine, Cytosine) et aux positions dans le génome. Les alignements des reads sont représentés par des k-mers, des séquences de tailles k. Les reads à alignements multiples sont traités en même temps à travers les classes d'équivalence qui regroupent tous les k-mers qui apparaissent pour les mêmes transcrits avec des fréquences similaires [33].
La quantification peut être dissociée de l'alignement si des fichiers au format BAM/SAM sont fournis, permettant son incorporation ponctuelle dans d'autres protocoles.

### 1.4.5 Analyse d'expressions différentielles des gènes

L'analyse des DEGs (*Differentially Expressed Genes*) requiert une matrice qui rapporte les comptes pour chacun des gènes détectés. DESeq2 est un des outils les plus populaires et démontre un bon équilibre en terme d'exactitude,

précision et sensibilité [34]. Une normalisation est effectuée pour tenir compte de la profondeur de séquençage des échantillons. Un modèle linéaire généralisé suivant une loi de distribution binomiale négative est ajustée pour chaque gène ; le modèle statistique estime la variabilité entre réplicats ainsi que la dispersion des expressions entre les conditions étudiées, et rapporte les résultats sous la forme d'un taux de log2FoldChange, le logarithme du ratio des expressions des différents gènes entre les conditions, avec comme hypothèse nulle l'absence d'expression différentielle [35].

### 1.4.6 Analyse de l'épissage alternatif différentiel

A travers la bioinformatique, l'épissage alternatif différentiel (DAS) est étudié par deux grandes approches basées respectivement sur les isoformes, exploitée par des outils tels que Cuffdiff2 ou DiffSplice, ou sur les comptes de reads. Cette dernière se ramifie en deux sous-catégories, l'analyse au niveau des exons ou des évènements. Les outils les plus populaires pour ces deux catégories respectives sont DEXSeq, EdgeR, Limma et rMATs, MAJIQ ou encore SUPPA2. rMATs est particulièrement représenté dans la recherche d'épissage différentiel [36].

L'étude basée sur les isoformes inclut une étape préliminaire de reconstruction des transcrits sur base des reads et de leur estimation pour chaque échantillon, après quoi les analyses statistiques sont réalisées afin d'identifier l'expression différentielle des isoformes entre les échantillons. Il existe deux méthodes d'évaluation : l'expression différentielle des transcrits (*Differential Transcript Expression*, DTE) et l'usage différentiel des transcrits (*Differential Transcript*, DTU) [36].

Un transcrit différentiellement exprimé implique une modification de son expression sans prendre en compte son gène d'appartenance et ses isoformes. Cette approche présente une limitation : par exemple, l'augmentation d'une isoforme précise peut s'expliquer soit par l'EA de l'ARNm ou par une altération de la transcription de son gène (augmentation de toutes les isoformes de celui-ci). L'efficacité et la pertinence de cette approche dépendent de l'étape de quantification des transcrits et détecte typiquement moins d'évènements que la seconde.
De son côté, l'usage différentiel des transcrits consiste à déterminer les ratios d'expression relative des différentes isoformes au sein de leur gène, pour différentes conditions. Un transcrit est considéré comme différentiellement utilisé si sa contribution dans l'expression totale des isoformes du gène est significativement perturbée.

Les méthodes basées sur les comptes suivent une stratégie différente et considèrent les exons qui, décomposés, peuvent correspondre à une fraction d'un gène.

L'approche basée sur l'exon considère chaque exon un à un et détermine de façon indépendante l'expression différentielle de ce dernier par rapport à l'ensemble des exons du gène (*Exon Usage*, EU). Cette approche limite l'interprétation notamment au niveau des évènements d'exclusions mutuelles (information perdue). Seule la caractéristique différentiellement exprimée est détectée. De plus, l'évènement d'épissage impliqué n'est pas identifié.

La dernière approche basée sur l'évènement permet une plus grande détection que les précédentes en termes de nombres d'évènements capturés. L'essence de ce type d'analyse consiste à examiner chaque événement d'épissage et évaluer leur pourcentage d'inclusion, dénoté $\psi$ (*PSI, Percent Spliced In*). $\psi$ correspond au ratio entre le

nombre de reads qui supportent l'inclusion de l'événement et le nombre total de reads qui supportent l'événement (inclusion et exclusion). Les moyennes des taux de PSI à travers les réplicats pour chaque évènements sont alors confrontées entre les conditions étudiées afin de déterminer la significativité de l'évènement. Cette approche permet une interprétation plus aisée de l'épissage en lui-même puisqu'elle rapporte l'information en évènements [47],[37]. A l'inverse, l'interprétation biologique est moins évidente car les événements sont analysés de manière disjointe, ce qui ne permet pas de tirer des conclusions sur les changements fonctionnels comme les domaines protéiques.

# 2 Matériels

## 2.1 Données

### 2.1.1 Echantillons *A. thaliana* et séquençage

Avant ce travail, des plantes d'*Arabidopsis thaliana* ont été cultivées en milieu hydroponique pendant neuf semaines. Les individus se divisent en deux génotypes, les sujets sauvages (wt) de souche Col-0 et des mutants pour le facteur d'épissage SR45 de souche *sr45-1*, et ont été exposés à deux conditions, contrôle (Ctrl) et carence en fer (Fe-) qui correspondent à des concentrations de cultures respectives de 10 et 0 µM de fer (pendant 3 semaines avant récolte), avec quatre réplicats pour chaque groupe (génotype et condition), soit 16 échantillons au total.

Des extraits de racines (1 µg d'ARN totaux) ont été prélevés et séquencés par Illumina NextSeq500 à quatre voies, sous forme de reads non-pairés. Le kit utilisé pour la préparation des librairies est le TruSeq Stranded mRNA Library Prep Kit et le séquençage de chaque échantillon a produit quatre fichiers, pour un total de 64 [2]. Ces séquences, sous forme de fichiers fastq, étaient disponibles à l'entame de ce travail.

### 2.1.2 Génome de référence et annotations

Le génome de référence d'*Arabidopsis thaliana* (version 10) ainsi que son fichier d'annotation GTF (version 11) a été téléchargé (le 17/07/2021) sur le site officiel du projet TAIR (The Arabidopsis Information Resource) accessible via ce lien : https ://www.arabidopsis.org/.

## 2.2 FastQC

FastQC est un logiciel qui propose différentes analyses de contrôle qualité pour identifier les éventuels problèmes de données de séquences brutes issues de séquençages à haut débit.

FastQC génère un rapport qui comporte les résumés des contrôles sous forme de graphiques et de tableaux et rend compte notamment de la qualité par base/séquence, du contenu en GC ou encore de la distribution des tailles des séquences.

L'outil peut être exécuté selon deux modes, interactif et non interactif. Le mode interactif permet de conduire les analyses dans l'application pour un nombre réduit de fichiers. Le mode non interactif permet l'intégration des étapes de contrôle qualité dans un pipeline plus large ou pour un nombre de fichiers à analyser plus conséquent [48].

Le programme est téléchargeable à cette adresse :
https ://www.bioinformatics.babraham.ac.uk/projects/fastqc/

## 2.3 Trimmomatic

Trimmomatic est un programme qui permet la filtration et le rognage de fichiers FASTQ générés par Illumina. Il permet d'exclure les reads de faible qualité ou les séquences des adaptateurs issus de la préparation des librairies, ce qui diminue les biais et artéfacts dans les analyses en aval. Trimmomatic accepte les reads sous forme pairée ou non pairée.

Le programme est téléchargeable à cette adresse :
http ://www.usadellab.org/cms/ ?page=trimmomatic

## 2.4 STAR

STAR est un outil d'alignement sensible aux jonctions d'épissage qui incorpore la détection de reads de fusion. Son algorithme est basé sur la création de tableaux de suffixes.

L'aligneur propose également des options précieuses pour la détection de nouveaux variants et peut générer les comptes des gènes en parallèle de l'étape d'alignement [38].

Le programme est téléchargeable au lien suivant :
https ://github.com/alexdobin/STAR

## 2.5 StringTie

StringTie permet l'assemblage d'alignements RNA-Seq en potentiels transcrits, la quantification de leur expression ainsi que la fusion de plusieurs transcriptomes [50].

L'outil est accessible à l'adresse suivante :
https ://ccb.jhu.edu/software/stringtie/

## 2.6 TACO

"*Transcriptome Assemblies Combined into One*", à l'instar de la fonctionnalité de StringTie-merge, est un assembleur de transcriptomes récent qui repose sur un algorithme de programmation dynamique et l'élaboration de graphes d'épissage pour accorder les différents échantillons.

Le logiciel est téléchargeable à l'adresse suivante :
https ://tacorna.github.io/

## 2.7 GFFCompare

GffCompare propose diverses fonctionnalités dans le but de traiter des fichiers d'annotations au format GFF.

L'outil fonctionne en ligne de commande et permet de comparer des assemblages, les fusionner, les classifier, détecter les dissimilitudes avec un fichier d'annotations de référence ou encore de générer des annotations.

Les comparaisons produites sont classifiées en codes qui décrivent chacun un évènement d'altération défini du transcrit originel adapté à la détection de variants ou de nouveaux transcrits [27].

GffCompare est accessible au lien suivant :
https ://ccb.jhu.edu/software/stringtie/gffcompare.shtml

## 2.8 Gff3sort.pl

Ce script Perl est conçu pour trier des annotations en considérant le chromosome et la position des caractéristiques rencontrées. Pour les fichiers au format GFF possédant un attribut *Parent*, le paramètre **precis** assure le respect de la hiérarchie avec leurs attributs *Enfants* respectifs dans le résultat de tri final [39].

## 2.9 Salmon

Salmon requiert un transcriptome de référence au format FASTA ou des résultats d'alignement BAM/SAM et des données de séquençage brutes FASTA/Q. L'outil réalise l'alignement des reads sur le transcriptome de référence et quantifie en parallèle les expressions des transcrits représentés.

Salmon est conçu pour détecter les erreurs spécifiques aux échantillons grâce à des modèles construits sur base d'observations d'authentiques données RNA-Seq [49].

L'outil est téléchargeable à l'adresse suivante :
https ://combine-lab.github.io/salmon/

## 2.10 RSEM

RSEM est un outil d'alignement et de quantification. Ses performances sont comparables voire supérieures à celles de ses concurrents basés sur l'alignement sur un génome de référence, et ses méthodes permettent de résoudre le problème d'ambiguïté d'alignement des reads [32].

L'outil est téléchargeable à l'adresse suivante :
https ://github.com/deweylab/RSEM

## 2.11 DESeq2

DESeq2 est un package R populaire qui présente une série de fonctions destinées à l'analyse de gènes différentiellement exprimés.

Cet outil nécessite une matrice d'expression des gènes ou des transcrits et repose sur un modèle binomial négatif pour modéliser les distributions.

DESeq2 peut être installé sous R avec la commande suivante :

```
if (!requireNamespace("BiocManager", quietly = TRUE))
install.packages("BiocManager")

BiocManager::install("DESeq2")
```

## 2.12 rMATs

Cet outil est dédié à l'étude de l'épissage alternatif sur base de l'évènement. rMATs peut être exécuté sur le produit d'alignement sous forme de fichiers BAM ou sur les fichiers FASTA/Q issus de séquençage et tient compte des réplicats biologiques.

rMATs est téléchargeable au lien suivant :
http ://rnaseq-mats.sourceforge.net/

## 2.13 Maser

Maser est un package R développé de manière à faire suite aux résultats générés par rMATs et faciliter leur visualisation et analyse.

Le package peut être téléchargé avec la commande suivante :

```
if (!requireNamespace("BiocManager", quietly = TRUE))
install.packages("BiocManager")

BiocManager::install("maser")
```

# 3 Méthodes

Les scripts permettant la mise en oeuvre des méthodes décrites dans cette section sont intégralement fournis au sein des annexes A.1-A.14.

## 3.1 Contrôle qualité et filtration des séquences

La qualité des données issues du séquençage Illumina est évaluée avec FastQC. Le séquenceur disposant de quatre voies, le séquençage de chaque échantillon a produit quatre fichiers fastq. Le contrôle qualité et la filtration ont été réalisés sur chaque fichier individuellement avant que ceux-ci soient fusionnés par échantillon respectif (Annexe A.1,3).
Sur base des informations recueillies par les rapports FastQC, les séquences contenues dans les fichiers sont filtrées et rognées par Trimmomatic. Les paramètres leading, trailing, slidingwindow, crop et minlen ont été respectivement définis avec les valeurs suivantes : 26, 26, 10 :26, 74 et 70 (Annexe A.3.1,2)
Le nombre de séquences moyennes avant et après filtration ainsi que le pourcentage de filtration moyen ont été calculés (Annexe A.3.4).

## 3.2 Premier alignement STAR

### 3.2.1 Indexation du génome

Le génération de l'index du génome est indiquée par l'option genomeGenerate du paramètre **runMode**.
**genomeDir** reçoit le chemin du dossier où les résultats sont enregistrés. Les chemin vers le génome de référence d'*A. thaliana* et le fichier d'annotations sont respectivement passés aux paramètres **genomeFastaFiles** et **sjdbGTFfile**.
La valeur 73 a été renseignée pour le paramètre **sjdbOverhang** et 12 pour **genomeSAindexNbases** (Annexe A.4.1).

### 3.2.2 Alignements

Les alignements ont été conduits pour chaque groupe de réplicats (Annexe A.4.2) en renseignant les membres au paramètre **readFilesIn**.
STAR est exécuté en mode d'alignement. Le paramètre **genomeDir** renseigne le dossier vers l'index du génome, le nombre de processus utilisés a été posé à six avec le paramètre **runThreadN**.

## 3.3 Filtration des jonctions

Les jonctions détectées ont été filtrées avec le script Perl SJ_filter.pl codé dans le cadre de ce travail (Annexe A.5).

Le script Perl ne prend qu'un seul argument, le chemin vers le fichier généré par STAR au terme du premier alignement.

## 3.4 Second alignement STAR

Les seconds alignements STAR ont été réalisé pour chacun des échantillons. Les jonctions d'épissage filtrées pour les différents groupes de réplicats enrichissent les annotations des échantillons correspondants via le paramètre **sjdbFileChrStartEnd**.
Les valeurs BAM et SortedByCoordinate ont été passées au paramètre **ouSAMtype** (Annexe A.6).

## 3.5 Construction des transcriptomes et assemblage

### 3.5.1 Modification des annotations de référence

Le fichier d'annotation a été traité par un oneliner Perl conçu pour cette étude dont le code est présenté en Annexe A.7.1.

### 3.5.2 Reconstruction des transcriptomes avec StringTie

Les assemblages de transcriptomes avec StringTie ont été réalisés pour chaque échantillon avec les paramètres **rf** et **G** suivi du chemin vers le fichier d'annotations.

### 3.5.3 Assemblages avec TACO

Les transcriptomes reconstruits ont été assemblés selon leur groupe de réplicat.
Les paramètres utilisés sont **p** et **filter-min-length N** avec les valeurs respectives *200* et *0* en précisant le fichier qui liste les chemins vers les transcriptomes reconstruits par StringTie et un répertoire pour les résultats avec **o** (Annexe A.8).

## 3.6 Comparaison et sélection des nouveaux transcrits

### 3.6.1 Comparaison avec GffCompare

La commande GffCompare a été exécutée avec les arguments **R**, **r** et le chemin vers le fichier d'annotations de référence en renseignant les quatre fichiers issus des assemblages de TACO. Le nombre de transcrits au total et ceux relatifs aux miRNAs ont été calculés (Annexe A.9.1).

### 3.6.2 Filtration et sélection des nouveaux transcrits

Le script Perl selection.pl, dont le code est présenté en annexe (Annexe A.9.4), a été codé lors de cette étude afin de traiter les résultats produits par GffCompare et filtrer les transcrits pertinents pour l'enrichissement des annotations.
Les codes de classe retenus pour la sélection des nouveaux variants listés dans le fichier classCodes.txt sont les suivants : **k**, **m**, **n**, **j**, **e**, **o**, **i** et **u**. Les combinaisons d'échantillons considérées pour l'établissement des listes de transcripts exprimés respectives sont les suivantes : **1**,**2**,**3**,**4**,**12**,**13**,**14**,**23**,**24**,**34**,**123**,**124**,**234**,**134**,**1234** et figurent dans le fichier comb.txt.

Un diagramme de Venn a été construit sur base des listes des transcrits par échantillon (*_lists*) (Annexe A.9.5).

#### 3.6.3 Filtration des miRNAs de code *k*

Parmi les nouveaux variants, les transcrits correspondants à des miRNAs avec un codes de classe $k$ attribué par GffCompare ont été filtrés. Cette étape a été réalisée par le script Perl miRNAs_filter.pl (Annexe A.9.6).

#### 3.6.4 Ajustement des annotations

Les annotations relatives aux nouveaux transcrits ont été formatées par le script Perl formatGtf.pl afin d'assurer l'association consistante des isoformes avec leur gène dans les analyses en aval (Annexe A.9.7).

### 3.7 Enrichissement

Les annotations de référence et des nouveaux variants ont été concaténées dans un fichier unique et triées grâce au script Perl gff3sort.pl avec ses paramètres par défaut (Annexe A.10).

### 3.8 Alignement STAR avec les annotations enrichies

#### 3.8.1 Re-indexation du génome avec les annotations enrichies

Une nouvelle indexation du génome est réalisée comme précédemment avec le fichier d'annotations enrichi (Annexe A.11.1).

#### 3.8.2 Alignement et quantification avec STAR

Les reads relatifs aux échantillons ont été alignés individuellement sur le génome de référence avec le nouveau fichier d'annotations enrichi et le paramètre **quantMode** avec l'argument *TranscriptomeSAM* (Annexe A.11.2-3).

### 3.9 Quantification

#### 3.9.1 RSEM

##### 3.9.1.1 Préparation

La préparation des index a été réalisée sur base du fichier d'annotations enrichi avec le paramètre **gtf** et du génome de référence par l'indication de son chemin (Annexe A.12.1).

##### 3.9.1.2 Calcul des expressions

La commande de calcul des expressions de transcrits a été exécutée avec les arguments **bam**, **no-bam-output**, **strandedness** et **p** avec respectivement les valeurs *reverse* et *16*, pour chaque résultat issus de l'alignement et de la quantification STAR (Annexe A.12.2).

## 3.10 DGE - DTE

L'analyse d'expressions différentielles de gènes a été menée avec les fonctions fournies par le package R DESeq2. Les données de comptage produites par RSEM ont été importées via le package R *TxImport* et les critères de significativité ont été définis pour un FoldChange de 2 et un FDR $< 0.05$. Les scripts sont disponibles en annexe (Annexe A.14).

La méthodologie d'analyse des DTEs est la même que celle suivie pour les gènes, hormis que les données de comptages importées correspondent aux expressions des isoformes.

## 3.11 DAS

### 3.11.1 rMATs

L'analyse de l'épissage alternatif différentiel a été réalisée avec rMATs. Les fichiers qui recensent les chemins vers les résultats d'alignements bam et qui discriminent les groupes de test ont été générés par des commandes bash (Annexe A.13).
Les paramètres utilisés pour l'analyse rMATs sont **gtf** et le chemin vers le fichier d'annotations enrichi, **t** *single*, **variable-read-length**, **allow-clipping**, **nthread** *16* et **libType** *fr-firstrand*.

### 3.11.2 Maser

Le script est disponible en annexe, A.14 . Les gènes alternativement épissés (ASG) ont été sélectionnés après filtration des évènements supportés par moins d'un read. Les DAS ont été sélectionnés pour une valeur de PSI $>$ 0.1 pour le même seuil de filtration.

# 4 Résultats

Une figure intégrative qui résume le protocole mis en oeuvre dans sa globalité est présentée en annexe (**Figures - A.0.1-2**).

## 4.1 Contrôle qualité et filtration des séquences

Un exemple de rapport d'analyse de la qualité des reads réalisée par FastQC avant et après filtration par trimmomatic est présenté à l'annexe xx.
L'analyse de données brutes reflète une bonne qualité générale jusqu'aux deux dernières positions (Annexe - **Fig. A.1.1)** où le score de qualité chute. Le contenu en base à travers les positions est constant pour l'intervalle 18 à 72 pb (Annexe - **Fig. A.1.2)**.

Les bases aux extrémités des reads dont la qualité n'excédait pas un score de 26 et les séquences de taille inférieure à 70 paires de base ont été filtrées par Trimmomatic. Les bases des séquences au-delà de la position 74 ont été retirées pour contourner la perte de qualité observée en fin de séquence.

Les anomalies du contenu en base au niveau de l'extrémité 5' des reads n'ont pas été sujettes à un traitement particulier. Elles sont issues des amorces utilisées lors de la préparation des librairies et implique un biais étendu à tous les reads séquencés. Les deux dernières positions filtrées en 3' sont issues des adaptateurs et ont été retirées.

La filtration par Trimmomatic a généré des reads de meilleure qualité (Annexe - **Fig. A.2.1**) et diminué les biais présents aux extrémités 3' (Annexe - **Fig. A.2.2**) en conservant plus de 93% des reads initiaux (**Table 1**).

TABLE 1 – Nombre de reads par échantillons pré- et post-filtre qualité

| Echantillon | Pré-filtration | Post-filtration | Filtrés (%) |
|---|---|---|---|
| 1 | 24,975,478 | 23,404,684 | 6.29 |
| 2 | 25,253,135 | 23,633,624 | 6.41 |
| 3 | 25,585,396 | 23,986,484 | 6.25 |
| 4 | 22,940,384 | 21,460,474 | 6.45 |
| 5 | 24,157,241 | 22,557,639 | 6.62 |
| 6 | 25,410,000 | 23,747,904 | 6.54 |
| 7 | 25,931,226 | 24,168,585 | 6.80 |
| 8 | 24,388,231 | 22,728,386 | 6.81 |
| 9 | 23,948,364 | 22,443,248 | 6.28 |
| 10 | 23,084,281 | 21,567,783 | 6.57 |
| 11 | 25,656,557 | 24,024,068 | 6.36 |
| 12 | 22,308,139 | 20,786,170 | 6.82 |
| 13 | 22,489,813 | 20,975,533 | 6.73 |
| 14 | 50,034,043 | 46,616,112 | 6.83 |
| 15 | 24,336,557 | 22,641,927 | 6.96 |
| 16 | 25,747,258 | 24,038,210 | 6.64 |

Les colonnes renseignent respectivement les échantillons considérés, le nombre de reads présents avant et après filtration et le pourcentage total de reads non conservés

## 4.2 Alignements STAR

### 4.2.1 Premiers alignements STAR

L'étape d'alignement nécessite la construction d'un index préalable du génome de référence. L'utilisation d'un fichier d'annotations permet d'améliorer la qualité des alignements compte tenu des jonctions d'épissage. Le paramètre **sjdbOverhang** spécifie la taille de la séquence génomique à utiliser autour des annotations lors de la collection des jonctions d'épissage par l'algorithme. La valeur idéale correspond à la taille des reads - 1. Comme présenté dans les rapports qualité, les tailles des reads après filtration sont comprises entre 70 et 74 pb, indiquant une valeur de 73 à utiliser [53].
La valeur de **genomeSAindexNbases** doit être ajustée suivant la taille du génome de l'organisme étudié. Il spécifie la longueur des chaînes de caractères à utiliser pour l'indexation des SA, et se calcule de la façon suivante : min(14, log2(GenomeLength)/2 - 1) [53]. La taille totale du génome d'*Arabidopsis thaliana* est d'approximativement 119,146,348 pb [52], soit un résultat de 12 selon l'équation précédante, arrondi à la baisse. La génération des indices produit plusieurs fichiers dont l'utilité est exclusivement réservée au processus d'alignement conduit par STAR.

Un premier alignement des reads sur le génome de référence avec le fichier d'annotations est réalisé. Ce premier passage de STAR vise à détecter de nouvelles jonctions d'épissage afin de les réutiliser dans un deuxième alignement consécutif pour augmenter les reads qui les supportent.

STAR propose naturellement une option pour conduire un double alignement automatisé (**Figure 1**). Dans ce cas de figure, l'option **twopassMode** *Basic* récolte les nouvelles jonctions d'épissage et enrichit les annotations avant d'amorcer un deuxième alignement. Si cette première approche est simple d'utilisation, elle offre moins de contrôle sur les informations réinjectées dans le deuxième passage, telles que les jonctions d'épissage relatives au chromosome mitochondrial qui peuvent interférer comme faux positifs lors des analyses en aval et rallonger le temps d'exécution d'analyses pour de larges échantillons [54]. Une alternative consiste à réaliser manuellement les alignements laissant la possibilité de poser les filtres souhaités entre les deux processus.

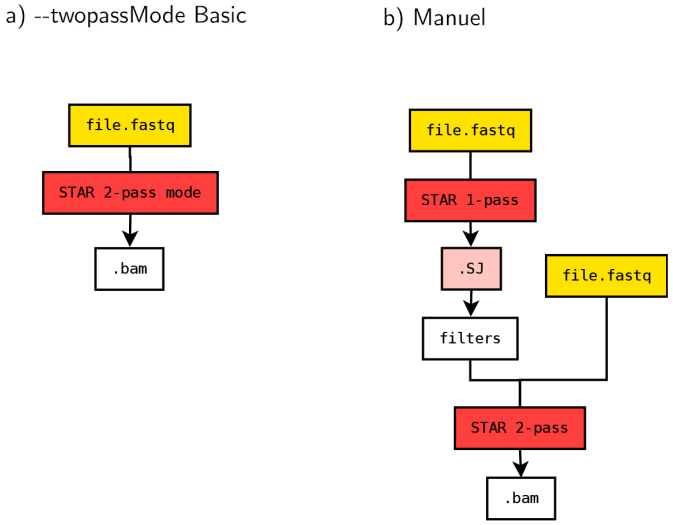


**Figure 1 :** Alternatives du mode d'alignement *2-pass* de STAR
Le panneau a) représente l'alignement 2-pass automatique réalisé en une seule commande. Le panneau b) illustre l'alternative en deux alignements. Dans cette version, les jonctions détectées à la suite du premier alignement sont stockées dans un fichier .SJ et peuvent être exposées à des filtres (filters), sous la forme d'un script Perl par exemple, avant de procéder au deuxième alignement.

STAR propose également deux stratégies d'alignement (**Figure 2**). La première option est classique et consiste à aligner chaque échantillon à tour de rôle. La seconde consiste à renseigner plusieurs échantillons. Les jonctions d'épissage détectées sont alors partagées et supportées à travers les alignements, ce qui permet d'augmenter le nombre de nouvelles jonctions détectées.

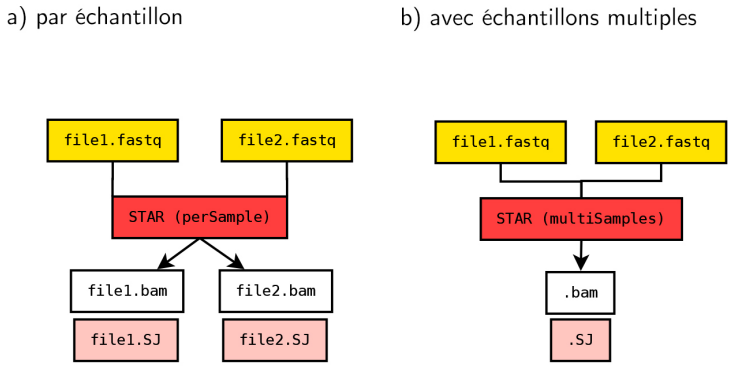


**Figure 2 :** Alignement unique ou multiple avec STAR. Les fichiers .bam contiennent les résultats d'alignement des reads sur le génome, les fichiers .SJ recensent les jonctions d'épissage détectées lors de cette procédure.

Dans une volonté de généralisation du pipeline, le premier alignement a été réalisé par groupes d'échantillons constituant des réplicats, les jonctions d'épissage obtenues ont été filtrées à l'aide d'un script Perl spécialement codé à cet effet au cours de ce travail (*filtering.pl*). Le script Perl sonde chaque ligne du fichier SJ passé en argument, chacune correspondant à une jonction d'épissage (**Table 3**), et filtre les informations relatives au chromosome mitochondrial, indiqué par la valeur *ChrM* dans la première colonne. Le deuxième alignement a ensuite été réalisé pour chaque échantillon, comme présenté à la **Figure 3**.

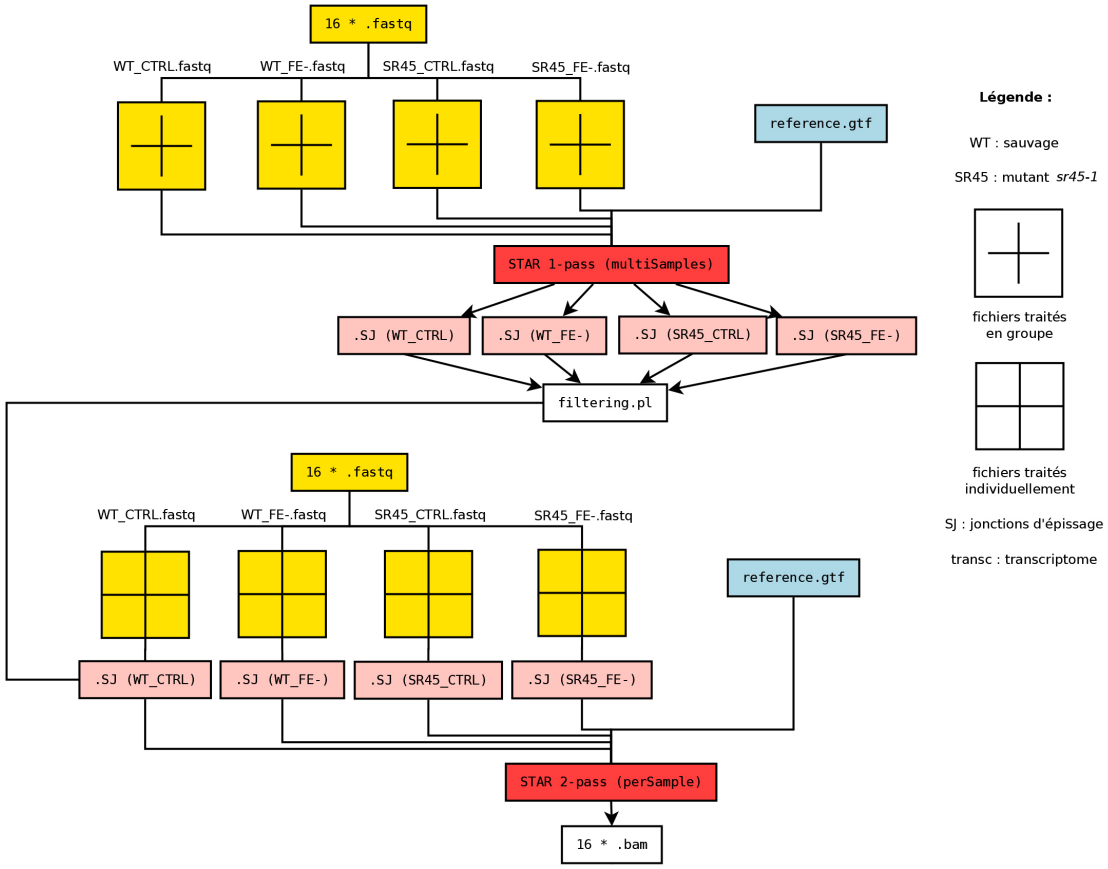


**Figure 3 :** Alignements STAR et filtration des jonctions.

Les statistiques obtenues pour les premiers alignements des différents groupes sont présentées à la **Table 2**. Le pourcentage de reads dont l'alignement est unique s'élève à approximativement 95% à travers les groupes, ce qui traduit un nombre réduit d'alignements ambigus. Le paramètre **outFilterMultimapNmax** qui détermine le nombre de loci maximum sur lesquels les reads peuvent s'aligner est défini à 10. D'après le rapport, entre 830 et 1347 reads répondent à ce critère et ont été filtrés, pour seulement 1.72% de reads à alignements multiples conservés. Finalement, entre 2.11 et 3.43% des reads n'ont pas été alignés et tombent dans la catégorie *too short*. Ce critère englobe deux phénomènes : la taille des séquences a été rognée de façon conséquente lors de la filtration ou l'alignement en tant que tel n'est pas signicatif. Puisque tous les reads présentent une taille comprise en 72 et 74 pb, la raison de cette exclusion est expliquée par un alignement peu résolu. Ce tableau illustre des statistiques d'alignement similaires entre les groupes considérés.

Table 2 – Statistiques des premiers alignements STAR

| | Groupes | | | |
|---|---|---|---|---|
| | WT_Ctrl | WT_-Fe | sr45_Ctrl | sr45_-Fe |
| Number of input reads | 93,202,514 | 92,485,266 | 114,271,782 | 88,821,269 |
| Average input read length | 73 | 73 | 73 | 73 |
| **UNIQUE READS :** | | | | |
| Uniquely mapped reads number | 89,913,602 | 87,709,373 | 109,436,578 | 84,304,108 |
| Uniquely mapped reads % | 96.47% | 94.84% | 95.77% | 94.91% |
| Average mapped length | 73.81 | 73.81 | 73.81 | 73.81 |
| Number of splices : Total | 20,540,151 | 20,984,805 | 25,117,459 | 19,732,877 |
| Number of splices : Annotated (sjdb) | 20,376,557 | 20,772,765 | 24,900,247 | 19,550,616 |
| Number of splices : GT/AG | 20,329,505 | 20,752,560 | 24,846,278 | 19,516,684 |
| Number of splices : GC/AG | 163,385 | 185,020 | 202,869 | 170,769 |
| Number of splices : AT/AC | 36,630 | 36,574 | 54,582 | 35,180 |
| **MULTI-MAPPING READS** | | | | |
| Number of reads mapped to multiple loci | 1,322,227 | 1,604,109 | 1,964,462 | 1,562,381 |
| % of reads mapped to multiple loci | 1.42% | 1.73% | 1.72% | 1.76% |
| Number of reads mapped to too many loci | 830 | 799 | 1,347 | 853 |
| % of reads mapped to too many loci | 0.00% | 0.00% | 0.00% | 0.00% |
| **UNMAPPED READS :** | | | | |
| Number of reads unmapped : too many mismatches | 0 | 0 | 0 | 0 |
| % of reads unmapped : too many mismatches | 0.00% | 0.00% | 0.00% | 0.00% |
| Number of reads unmapped : too short | 1,965,834 | 3,170,961 | 2,869,372 | 2,953,896 |
| % of reads unmapped : too short | 2.11% | 3.43% | 2.51% | 3.33% |
| Number of reads unmapped : other | 21 | 24 | 23 | 31 |
| % of reads unmapped : other | 0.00% | 0.00% | 0.00% | 0.00% |

### 4.2.2 Filtration des jonctions d'épissage

Les jonctions détectées par STAR sont listées dans un fichier avec l'extension SJ.out.tab construit en neuf colonnes, illustré à la **Table 3** ci-dessous.

TABLE 3 – Structure du fichier de jonctions (SJ.out.tab) produit par STAR, pour le groupe *wt - Ctrl*

| Chromosome | Première base | Dernière base | Brin | Canonicité | Annotation | Align. Uniques | Align. Multiples | Taille max. |
|---|---|---|---|---|---|---|---|---|
| Chr1 | 3,914 | 3,995 | 1 | 1 | 1 | 212 | 0 | 37 |
| Chr1 | 4,277 | 4,485 | 1 | 1 | 1 | 179 | 0 | 37 |
| Chr1 | 4,277 | 4,505 | 1 | 1 | 0 | 2 | 0 | 28 |
| Chr1 | 4,606 | 4,705 | 1 | 1 | 1 | 236 | 0 | 37 |
| Chr1 | 5,096 | 5,173 | 1 | 1 | 1 | 204 | 0 | 37 |
| Chr1 | 5,327 | 5,438 | 1 | 1 | 1 | 243 | 0 | 36 |
| Chr1 | 6,925 | 7,001 | 2 | 2 | 0 | 1 | 0 | 26 |
| Chr1 | 7,070 | 7,156 | 2 | 2 | 1 | 206 | 0 | 37 |
| Chr1 | 7,233 | 7,383 | 2 | 2 | 1 | 162 | 1 | 36 |

Le tableau présente les 9 premières lignes du fichier SJ.out.tab. Les colonnes indiquent respectivement le chromosome d'appartenance de la jonction, les positions de la première et dernière base de l'intron, l'orientation du brin (0 pour indéfini, 1 ou 2 pour respectivement brin sens et antisens), un code indiquant la canonicité ( 0 pour non canonique, les codes 1 à 6 correspondant aux motifs GT/AG, CT/AC, GC/AG, CT/GC, AT/AC ou GT/AT), une colonne binaire qui indique la présence de la jonction dans les annotations (0 pour absente et 1 pour représentée), le nombre de reads alignés de façon unique ou multiple et la taille maximale du fragment qui surplombe la jonction.

Le script Perl *SJ_filter.pl* a été conçu dans le but de filtrer les jonctions d'épissage qui seront réinjectées dans le second alignement STAR. Dans le cadre de cette étude, seules les jonctions relatives au chromosome mitochondrial ont été écartées, motivé par les raisons décrites plus haut. La stratégie du code peut être aisément étendue afin d'implémenter des critères supplémentaires.

Après filtration, les statistiques de détection des jonctions sont présentées à la **Table 4**. Pour chaque groupe d'échantillons, entre un quart et un tiers des jonctions détectées n'étaient pas annotées dans le fichier de référence d'*Arabidopsis thaliana* . Très peu de jonctions mitochondriales sont recensées, ce qui s'explique par le nombre réduit de gènes attachés aux mitochondries (60) [52] et par l'instabilité des rares transcrits mitochondriaux polyadénylés [40].

TABLE 4 – Statistiques de détection des jonctions d'épissage à l'issue du premier passage avec STAR

| Groupes | SJ totales | Nouvelles SJ (%) | SJ mitochondriales |
|---|---|---|---|
| WT_-Fe | 150,117 | 39,902 (26.58) | 9 |
| WT_Ctrl | 147,319 | 36,164 (24.55) | 4 |
| sr45_-Fe | 149,791 | 39,426 (26.32) | 9 |
| sr45_Ctrl | 154,764 | 43,392 (28.04) | 4 |

Les nouvelles jonctions d'épissage (*Splice Junctions*, SJ) ont été sélectionnées pour des valeurs de 0 dans la colonne *Annotations* des fichiers SJ.out.tab, voir **Table 3**.

## 4.3 Second alignement STAR

Un second alignement est réalisé pour chaque échantillon individuellement. Les annotations des échantillons sont enrichies par les jonctions détectées sur base de leur génotype et leur condition de culture correspondante. De ce fait, les SJ (*Splice Junctions*) ne sont partagées qu'entre réplicats (**Figure 3**), et cette scission empêche l'augmentation de reads à alignements multiples.
L'indexation du génome étant construite sur base du génome de référence et des annotations fournies, l'enrichissement de ces dernières se conduit de deux façons : en cas d'ajout de nouvelles annotations, un nouveau processus d'indexation est réalisé avec le fichier d'annotations enrichi. Dans le cas d'un second alignement, les nouvelles jonctions d'épissage détectées sont passées au paramètre **sjdbFileChrStartEnd** sans nécessiter une nouvelle indexation manuelle [58].
Le paramètre **ouSAMtype** définit le format sous lequel l'alignement est produit. La valeur par défaut est le SAM. Ce paramètre accepte deux champs, *BAM* lorsque la conversion automatique vers le SAM n'est pas désirée, et *Unsorted* ou *SortedByCoordinate* selon la volonté de trier les résultats de l'alignement. Cette dernière option est requise pour assurer la compatibilité des fichiers avec l'assembleur StringTie [50],[54].

Les résultats des seconds alignements sont fournis à la **Table 5 ci-dessous**. De manière générale, les taux calculés entre les deux procédures d'alignements sont similaires - les pourcentages de reads alignés de façon unique varient entre 94.17 et 97.33%, les comptes des jonctions correspondent aux nombres calculés précédemment à un facteur quatre près et le pourcentage de reads rapportés dans la catégorie *too short* vaut 2.79% de moyenne. Une légère augmentation est notable en termes de nombre de reads alignés sur un nombre trop important de loci (10) et des pourcentage de reads à alignements multiples. Cette observation est cohérente avec l'augmentation des jonctions d'épissage considérées qui autorise plus d'alternatives de découpes et donc, d'alignements.

Table 5 – Statistiques du deuxième alignement

| | Echantillons | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| Number of input reads | 23,404,684 | 23,633,624 | 23,986,484 | 21,460,474 | 22,557,639 | 23,747,904 | 24,168,585 | 22,728,386 | 22,443,248 | 21,567,783 | 24,024,068 | 20,786,170 | 20,975,533 | 46,616,112 | 22,641,927 | 24,038,210 |
| Average input read length | 73 | 73 | 73 | 73 | 73 | 73 | 73 | 73 | 73 | 73 | 73 | 73 | 73 | 73 | 73 | 73 |
| **UNIQUE READS :** | | | | | | | | | | | | | | | | |
| Uniquely mapped reads number | 22,211,443 | 22,419,048 | 22,587,754 | 20,367,735 | 21,869,208 | 23,114,736 | 23,080,804 | 21,642,967 | 21,259,629 | 20,368,535 | 22,785,334 | 19,748,097 | 20,201,054 | 44,234,279 | 21,460,222 | 23,224,875 |
| Uniquely mapped reads % | 94.90% | 94.86% | 94.17% | 94.91% | 96.95% | 97.33% | 95.50% | 95.22% | 94.73% | 94.44% | 94.84% | 95.01% | 96.31% | 94.89% | 94.78% | 96.62% |
| Average mapped length | 73.82 | 73.81 | 73.81 | 73.81 | 73.81 | 73.81 | 73.81 | 73.82 | 73.81 | 73.81 | 73.81 | 73.81 | 73.82 | 73.82 | 73.81 | 73.81 |
| Number of splices : Total | 5,393,524 | 5,398,701 | 5,395,592 | 4,916,168 | 4,992,341 | 5,176,923 | 5,416,053 | 5,071,124 | 5,040,142 | 4,818,040 | 5,374,919 | 4,615,134 | 4,581,719 | 10,309,048 | 4,959,280 | 5,446,378 |
| Number of splices : Annotated (sjdb) | 5,386,654 | 5,391,739 | 5,388,617 | 4,910,329 | 4,987,535 | 5,171,870 | 5,411,226 | 5,066,722 | 5,033,984 | 4,812,248 | 5,368,260 | 4,609,328 | 4,577,605 | 10,299,074 | 4,954,273 | 5,441,634 |
| Number of splices : GT/AG | 5,330,444 | 5,335,761 | 5,333,024 | 4,860,894 | 4,939,999 | 5,123,935 | 5,356,632 | 5,015,700 | 4,984,064 | 4,764,157 | 5,315,273 | 4,564,001 | 4,532,223 | 10,196,696 | 4,905,347 | 5,385,978 |
| Number of splices : GC/AG | 50,481 | 50,541 | 50,127 | 44,943 | 41,224 | 41,705 | 46,468 | 43,203 | 45,069 | 43,504 | 47,956 | 40,756 | 36,939 | 84,388 | 41,247 | 46,746 |
| Number of splices : AT/AC | 9,462 | 9,306 | 9,295 | 7,520 | 7,958 | 8,112 | 9,100 | 8,844 | 8,286 | 7,816 | 8,781 | 7,900 | 9,896 | 21,968 | 9,792 | 10,410 |
| **MULTI-MAPPING READS** | | | | | | | | | | | | | | | | |
| Number of reads mapped to multiple loci | 448,350 | 446,886 | 441,827 | 390,266 | 375,986 | 376,079 | 401,196 | 369,063 | 444,025 | 409,642 | 455,729 | 393,155 | 428,139 | 845,457 | 531,626 | 454,962 |
| % of reads mapped to multiple loci | 1.92% | 1.89% | 1.84% | 1.82% | 1.67% | 1.58% | 1.66% | 1.62% | 1.98% | 1.90% | 1.90% | 1.89% | 2.04% | 1.81% | 2.35% | 1.89% |
| Number of reads mapped to too many loci | 641 | 687 | 783 | 617 | 1,102 | 1,015 | 1,587 | 1,393 | 828 | 936 | 1,059 | 1,012 | 2,993 | 6,810 | 3,348 | 2,366 |
| % of reads mapped to too many loci | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.01% | 0.01% | 0.00% | 0.00% | 0.00% | 0.00% | 0.01% | 0.01% | 0.01% | 0.01% |
| **UNMAPPED READS :** | | | | | | | | | | | | | | | | |
| Number of reads unmapped : too many mismatches | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| % of reads unmapped : too many mismatches | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Number of reads unmapped : too short | 744,109 | 766,857 | 955,953 | 701,732 | 310,969 | 255,764 | 684,437 | 714,488 | 738,578 | 788,443 | 781,643 | 643,650 | 342,476 | 1,527,176 | 645,522 | 355,242 |
| % of reads unmapped : too short | 3.18% | 3.24% | 3.99% | 3.27% | 1.38% | 1.08% | 2.83% | 3.14% | 3.29% | 3.66% | 3.25% | 3.10% | 1.63% | 3.28% | 2.85% | 1.48% |
| Number of reads unmapped : other | 141 | 146 | 167 | 124 | 374 | 310 | 561 | 475 | 188 | 227 | 303 | 256 | 871 | 2,390 | 1,209 | 765 |
| % of reads unmapped : other | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.01% | 0.01% | 0.00% |

## 4.4 Construction des transcriptomes et assemblage

Pour chaque échantillon, le transcriptome respectif a été reconstruit à partir des alignements STAR avec StringTie. Des transcriptomes consensus par groupe d'échantillons en réplicats ont ensuite été assemblés avec TACO, et les résultats ont été comparés avec le fichier de référence via GffCompare. Les annotations concernant les nouveaux transcrits ont été filtrées et ajoutées à la référence, comme illustré à la **Figure 4** suivante.

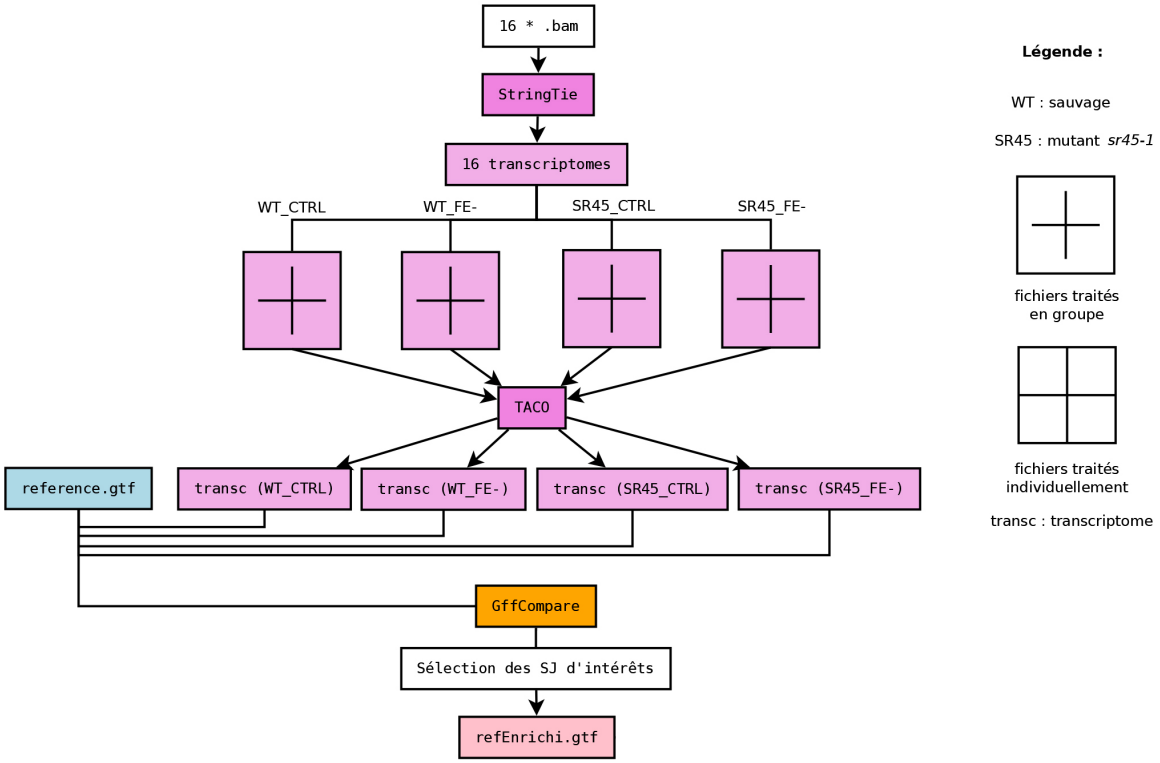


**Figure 4 :** Schéma résumé représentant les étapes de reconstruction des transcriptomes avec StringTie, les assemblages par TACO, les comparaisons des annotations de référence avec GffCompare et la sélection des nouveaux variants.

### 4.4.1 Reconstruction des transcriptomes avec StringTie

Les fichiers d'annotations au format GTF peuvent suivre différentes structures tandis que l'essence de l'information reste inchangée (attributs utilisés, ordre de recensement). Les miRNAs rapportés dans le fichier d'annotations fourni par le site du projet TAIR provoquent des avertissements à la reconstruction causés par leur identifiant d'attribut qui génèrent une ambiguïté. Le problème a été contourné en remplaçant ces attributs, *miRNA_primary_transcript* et *miRNA*, par des équivalents utilisés dans les annotations, respectivement *exon* et *CDS* (*Coding Sequence*).

L'exécution de StringTie en mode assembleur requiert le chemin vers un ou plusieurs fichiers BAM issus d'alignements. Le paramètre **G** guide la reconstruction grâce aux annotations. Le dernier paramètre **rf** indique l'utilisation d'une bibliothèque dont l'information du brin originel est conservée. L'option est à omettre en cas de bibliothèques dites *unstranded*, **rf** étant la valeur à adopter pour les kits Illumina qui impliquent une incorporation au dUTP

[55]. Les transcriptomes ont été reconstruits pour chaque échantillon. Sur base des annotations et des résultats d'alignement, StringTie extrait les transcrits détectés et calcule parallèlement les expressions relatives de ces différents transcrits. Le résultat présente une structure similaire au format GTF, les exons sont énumérés à la suite des transcrits, triés par coordonnées génomiques. Les attributs générés par StringTie sont les identifiants des gènes et transcrits, leur référence si l'élément figure dans les annotations, une mesure de la couverture en nombre de reads et les expressions rapportées en unités FPKM et TPM. Un exemple est illustré à la **Table 17** (Annexe A.7.2.1).

### 4.4.2 Assemblages avec TACO

Une fois les transcriptomes des seize échantillons reconstruits, TACO est utilisé pour assembler les résultats par groupe d'échantillons en réplicats en transcriptome unique.

Cette étape est accompagnée de filtres proposés par défaut qui peuvent être ajustés : **filter-min-length** *200* spécifie la taille minimale en pb des transcrits à conserver, **isoform-frac 0.05** exclut les isoformes dont l'expression est inférieure à 5% de l'isoforme majoritaire du gène en question et **filter-min-expr 0.5** qui ignore les transcrits dont l'expression en FPKM est inférieure à cette valeur. Les microRNAs (miRNAs) sont des transcrits impliqués dans la régulation de l'expression de certains gènes dont l'ordre de grandeur est de 22 pb. SR45 étant suspecté de réguler certains membres [7O], afin de permettre leur conservation dans les analyses en aval, le seuil de taille minimale des fragments à conserver a été défini au minimum [56]. Les fichiers à assembler sont indiqués en argument à TACO ou sous la forme d'une liste de chemins.

Les assemblages résultants sont générés au format GTF et BED (*Browser Extensible Data*) [30]. Les attributs d'identification des transcrits sont systématiques et propres à l'outil, un exemple est présenté en Annexe A.8.3 à la **Table 18**.

Des fichiers supplémentaires synthétisent les comptes de transfrags filtrés sur base des critères renseignés en paramètres (**Table 6**). Afin de vérifier l'impact de la diminution de la taille minimale de 200 à 0 pb, une comparaison des résultats pour le groupe WT_-Fe est présentée aux **Tables 6** et **7**, les tables complètes présentent des valeurs similaires entre groupes et sont disponibles en Annexe A.8.4 aux **Tables 19** et **20**. Près de 40,000 transcrits (appelés *transfrags*) sont assemblés par TACO à travers les différents échantillons. Parmi ceux-ci, approximativement 10% sont filtrés sur base d'une valeur d'expression inférieure à 0.5 FPKM. La diminution de la valeur du paramètre **filter-min-length** à 0 pb implique la conservation d'un nombre raisonnablement limité de transfrags, évalué à un peu plus d'1% du nombre total de transfrags détectés.

TABLE 6 – Filtration des transcrits avec **filter-min-length** *200* par TACO pour le groupe WT_-Fe

| sample_id | num_transfrags | filtered_length | filtered_expr | filtered_splice |
|---|---|---|---|---|
| 1 | 39,268 | 492 | 3,986 | 0 |
| 2 | 39,405 | 487 | 3,908 | 0 |
| 3 | 39,292 | 480 | 3,954 | 0 |
| 4 | 39,067 | 494 | 3,642 | 0 |

La première colonne désigne l'échantillon considéré : chaque résultat TACO est issu de l'assemblage supporté par quatre réplicats, indiqués par les indices 1 à 4. *num_ transfrags* recense le nombre total de transfrags détectés, les trois dernières colonnes contiennent les comptes des transfrags filtrés respectivement sur base du seuil de taille et d'expression minimales choisies ainsi que la canonicité (filtration de jonctions suivant leur motif).

TABLE 7 – Filtration des transcrits avec **filter-min-length** *0* par TACO

| sample_id | num_transfrags | filtered_length | filtered_expr | filtered_splice |
|---|---|---|---|---|
| 1 | 39,268 | 0 | 3,986 | 0 |
| 2 | 39,405 | 0 | 3,908 | 0 |
| 3 | 39,292 | 0 | 3,954 | 0 |
| 4 | 39,067 | 0 | 3,642 | 0 |

La première colonne désigne l'échantillon considéré : chaque résultat TACO est issu de l'assemblage supporté par quatre réplicats, indiqués par les indices 1 à 4. *num_ transfrags* recense le nombre total de transfrags détectés, les trois dernières colonnes contiennent les comptes des transfrags filtrés respectivement sur base du seuil de taille et d'expression minimales choisies ainsi que la canonicité (filtration de jonctions suivant leur motif).

## 4.5 Comparaison et sélection des nouveaux transcrits

### 4.5.1 Comparaison avec GffCompare

GffCompare prend en paramètres le fichier de référence indiqué par **r** et les fichiers à comparer avec ce dernier sous la forme d'une énumération ou d'un chemin vers une liste.

Statistiquement, la comparaison est réalisée à plusieurs niveaux ; l'exon, l'intron individuel ou l'ensemble des introns d'un transcrit, le transcrit à travers ses exons au niveau individuel et multiple et, dernièrement, au niveau des loci. Les statistiques d'alignement des caractéristiques génomiques sur la référence sont calculées à travers les taux de vrai positif (TP), fau x négatif (FN) et faux positif (FP) résumées en deux valeurs, la **sensibilité** : $TP/(TP + FN)$ et la **précision** : $TP/(TP + FP)$ [57].

Un TP est une structure dont les coordonnées correspondent aux annotations de référence, un FN est comptabilisé lorsqu'une annotation de référence n'est pas supportée par les transfrags et, finalement, un FP correspond au support d'un transfrag à travers les échantillons pour lequel aucune annotation n'est répertoriée.

Lorsque le paramètre **r** est utilisé, un code de classe est attribué pour chaque transfrag sur base des différences enregistrées pour les introns communs avec son transcrit de référence (si annoté). Une illustration des codes sélectionnés est présentée à la **Figure 5**.

L'option **R** permet d'omettre dans les statistiques les annotations qui n'ont pas trouvé de correspondance à travers les différents fichiers testés. Cette option est typiquement pertinente quand l'étude est portée sur un type tissulaire restreint, en l'occurrence les racines, où les gènes rapportés dans les annotations pour d'autres tissus tels que les parties aériennes ne sont pas représentés dans les échantillons. Les statistiques obtenues respectivement avec et sans ce paramètre sont présentées à la **Table 8** et **9**.

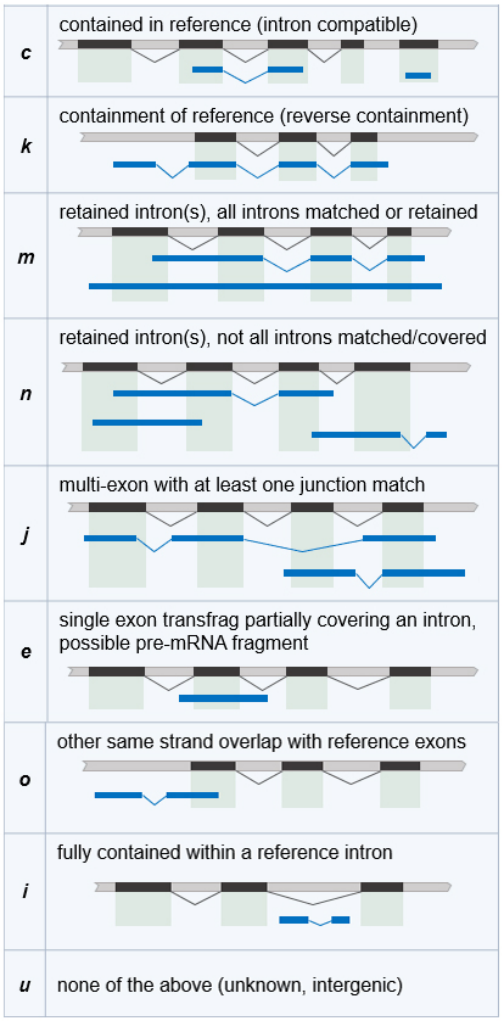


**Figure 5 :** Codes de classe GffCompare sélectionnés décrivant les différences avec le fichier d'annotation de référence
En noir, les informations propres aux annotations de référence, **en bleu**, les transfrags testés. Les codes retenus pour la sélection de nouveaux variants sont les suivants : **k**, **m**, **n**, **j**, **e**, **o**, **i** et **u**. Le code **c** correspond aux transfrags inclus dans les transcrits annotés et autorise des modifications subtiles de ses exons. Le code **k** concerne les transfrags qui contiennent une annotation de référence pouvant illustrer l'utilisation d'un promoteur alternatif ou une inclusion d'exon. Les codes **m** et **n** impliquent tous deux des évènements de rétention d'introns. Le premier nécessite que tous les introns aient été supportés tandis que le deuxième est plus laxiste. Le code **j** est rapporté quand plusieurs exons trouvent correspondance et qu'ils partagent une jonction au minimum, pouvant illustrer l'utilisation d'un promoteur alternatif ou une exclusion d'exon. Le code **e** concerne des exons uniques qui surplombent partiellement un intron et capturent les fragments d'ARNm pré-matures. Le code **o** autorise le débordement du transfrag pour peu qu'il superpose un exon de référence, le code **i** implique des éléments contenus dans des introns tels que des miRNAs introniques et, finalement, le code **u** reporte les éléments jamais annotés ou situés entre deux gènes.

TABLE 8 – Statistiques GffCompare avec l'argument **R** excluant de l'analyse les gènes non exprimés dans le jeu de données

| Echantillons | WT_-Fe | | WT_Ctrl | | sr45_-Fe | | sr45_Ctrl | |
|---|---|---|---|---|---|---|---|---|
| | Sensitivity | Precision | Sensitivity | Precision | Sensitivity | Precision | Sensitivity | Precision |
| Base level : | 94.5 | 97.8 | 94.5 | 98.3 | 94.5 | 98.0 | 94.6 | 97.9 |
| Exon level : | 81.2 | 90.5 | 81.4 | 91.4 | 81.1 | 90.8 | 81.2 | 90.8 |
| Intron level : | 89.0 | 94.8 | 89.1 | 95.4 | 88.9 | 95.1 | 89.0 | 94.9 |
| Intron chain level : | 51.4 | 67.1 | 51.6 | 69.2 | 51.1 | 67.5 | 51.0 | 67.4 |
| Transcript level : | 54.7 | 70.9 | 55.0 | 72.8 | 54.6 | 71.4 | 54.3 | 71.1 |
| Locus level : | 84.6 | 85.8 | 86.1 | 86.7 | 85.3 | 85.9 | 84.8 | 85.6 |
| Matching intron chains : | 18,636 | | 19,003 | | 18,645 | | 18,778 | |
| Matching transcripts : | 22,756 | | 23,100 | | 22,830 | | 22,803 | |
| Matching loci : | 18,268 | | 18,715 | | 18,501 | | 18,433 | |
| Missed exons : | 5,160/163,448 | ( 3.2%) | 4,985/165,353 | ( 3.0%) | 5,086/164,059 | ( 3.1%) | 4,973/165,290 | ( 3.0%) |
| Novel exons : | 1,122/136,537 | ( 0.8%) | 866/137,330 | ( 0.6%) | 1,002/136,504 | ( 0.7%) | 1,033/137,667 | ( 0.8%) |
| Missed introns : | 5,194/115,835 | ( 4.5%) | 5,145/117,246 | ( 4.4%) | 5,326/116,215 | ( 4.6%) | 5,239/117,219 | ( 4.5%) |
| Novel introns : | 1,972/108,750 | ( 1.8%) | 1,749/109,579 | ( 1.6%) | 1,704/108,605 | ( 1.6%) | 1,864/109,893 | ( 1.7%) |
| Missed loci : | 0/21,589 | ( 0.0%) | 0/21,728 | ( 0.0%) | 0/21,694 | ( 0.0%) | 0/21,739 | ( 0.0%) |
| Novel loci : | 163/21,247 | ( 0.8%) | 142/21,556 | ( 0.7%) | 171/21,502 | ( 0.8%) | 179/21,516 | ( 0.8%) |

**Sensitivity** = TP / (TP+FN), **Precision** = TP / (TP+FP).

TABLE 9 – Statistiques GffCompare par défaut, prenant en compte dans l'analyse les gènes non exprimés dans le jeu de données

| Echantillons | WT_-Fe | | WT_Ctrl | | sr45_-Fe | | sr45_Ctrl | |
|---|---|---|---|---|---|---|---|---|
| | Sensitivity | Precision | Sensitivity | Precision | Sensitivity | Precision | Sensitivity | Precision |
| Base level : | 65.4 | 97.8 | 66.2 | 98.3 | 65.7 | 98.0 | 66.2 | 97.9 |
| Exon level : | 65.7 | 90.5 | 66.7 | 91.4 | 65.9 | 90.8 | 66.4 | 90.8 |
| Intron level : | 75.6 | 94.8 | 76.6 | 95.4 | 75.7 | 95.1 | 76.5 | 94.9 |
| Intron chain level : | 42.1 | 67.1 | 42.9 | 69.2 | 42.1 | 67.5 | 42.4 | 67.4 |
| Transcript level : | 38.5 | 70.9 | 39.0 | 72.8 | 38.6 | 71.4 | 38.5 | 71.1 |
| Locus level : | 49.2 | 85.8 | 50.4 | 86.7 | 49.8 | 85.9 | 49.6 | 85.6 |
| Matching intron chains : | 18,636 | | 19,003 | | 18,645 | | 18,778 | |
| Matching transcripts : | 22,756 | | 23,100 | | 22,830 | | 22,803 | |
| Matching loci : | 18,268 | | 18,715 | | 18,501 | | 18,433 | |
| Missed exons : | 43,654/201,942 | ( 21.6%) | 41,574/201,942 | ( 20.6%) | 42,969/201,942 | ( 21.3%) | 41,625/201,942 | ( 20.6%) |
| Novel exons : | 1,122/136,537 | ( 0.8%) | 866/137,330 | ( 0.6%) | 1,002/136,504 | ( 0.7%) | 1,033/137,667 | ( 0.8%) |
| Missed introns : | 25,708/136,349 | ( 18.9%) | 24,248/136,349 | ( 17.8%) | 25,460/136,349 | ( 18.7%) | 24,369/136,349 | ( 17.9%) |
| Novel introns : | 1,972/108,750 | ( 1.8%) | 1,749/109,579 | ( 1.6%) | 1,704/108,605 | ( 1.6%) | 1,864/109,893 | ( 1.7%) |
| Missed loci : | 15,570/37,159 | ( 41.9%) | 15,431/37,159 | ( 41.5%) | 15,465/37,159 | ( 41.6%) | 15,420/37,159 | ( 41.5%) |
| Novel loci : | 163/21,247 | ( 0.8%) | 142/21,556 | ( 0.7%) | 171/21,502 | ( 0.8%) | 179/21,516 | ( 0.8%) |

Les mesures de sensibilité et de précision sont similaires à travers les groupes d'échantillons. On observe la diminution de ces taux à mesure que l'échelle diminue - la sensibilité chute de l'ordre de 40% (94.5% à 54.5%)

respectivement lorsque les mesures sont effectuées au niveau des bases et des transcrits. Cette observation peut s'expliquer par la capture de nouveaux introns et exons. Ces nouveaux éléments sont limités (approximativement 1000 et 2000 resp. par échantillon) tandis que leur arrangement à plus haut niveau permet de multiples combinaisons. Les taux de nouveaux exons, introns et loci sont inférieurs au pourcent et sont du même ordre de grandeur à travers les groupes considérés.
L'utilisation différentielle de l'argument **R** impacte uniquement la sensibilité. En effet, le taux de FN traduit les éléments des références non couverts par les transfrags. Puisque celui-ci est un terme du dénominateur dans la mesure de la sensibilité, augmenter le nombre d'exons de référence non représentés, soit inclure les annotations relatives aux parties aériennes, conduit à une diminution de la statistique [57].

Divers fichiers sont produits suite à l'exécution de GffCompare (**Figure 6**). Un fichier à l'extension tmap recense les informations relatives à chaque transfrag comparé aux annotations de référence. Si plusieurs échantillons sont renseignés, les .tmap (**Table 10**) sont fusionnés en un fichier unique non redondant au postfixe *.combined.gtf* (**Table 11**) avec leur expression relative et code de classe. Un second fichier au postfixe *.tracking* (**Table 12**) établit la correspondance entre les transfrags répertoriés dans le fichier *.combined.gtf* et les échantillons dans lesquels ils ont été détectés. Cette trace est conservée à travers plusieurs colonnes qui correspondent au nombre et indices des échantillons testés. Les échantillons qui contiennent le transfrag présentent ses identifiants définis par TACO et les échantillons qui en sont dépourvus affichent le symbole - [57].

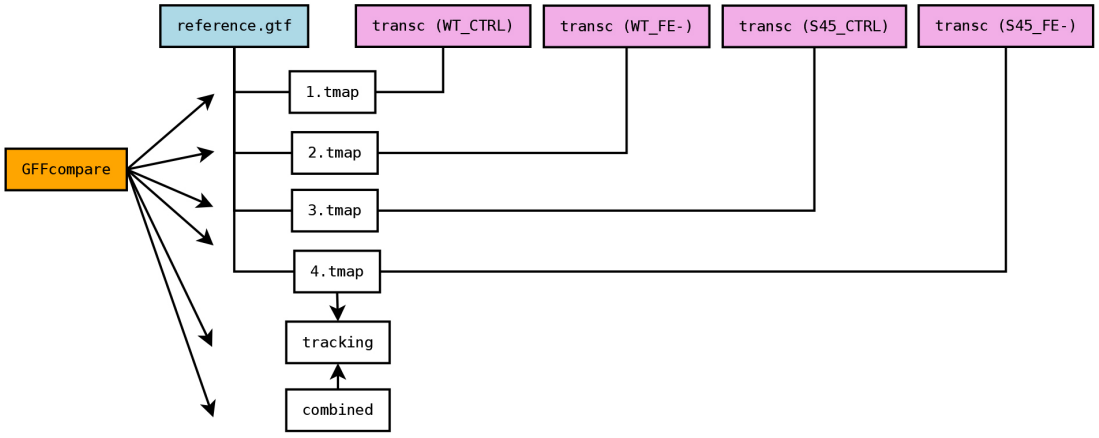


**Figure 6 :** Schéma des fichiers produits par GffCompare pour plusieurs échantillons.

Table 10 – Structure du fichier *tmap*

| ref_gene_id | ref_id | class_code | qry_gene_id | qry_id | num_exons | FPKM | TPM | cov | len | major_iso_id | ref_match_len |
|---|---|---|---|---|---|---|---|---|---|---|---|
| AT1G01010 | AT1G01010.1 | = | G1 | TU1 | 6 | 0 | 0 | 0 | 1,688 | TU1 | 1,688 |
| AT1G01020 | AT1G01020.1 | = | G3 | TU3 | 9 | 0 | 0 | 0 | 1,329 | TU7 | 1,329 |
| AT1G01020 | AT1G01020.3 | = | G3 | TU4 | 8 | 0 | 0 | 0 | 1,420 | TU7 | 1,420 |
| AT1G01020 | AT1G01020.1 | m | G3 | TU5 | 8 | 0 | 0 | 0 | 1,435 | TU7 | 1,329 |
| AT1G01020 | AT1G01020.5 | = | G3 | TU6 | 9 | 0 | 0 | 0 | 1,306 | TU7 | 1,306 |
| AT1G01020 | AT1G01020.1 | n | G3 | TU7 | 7 | 0 | 0 | 0 | 1,548 | TU7 | 1,329 |
| AT1G01020 | AT1G01020.1 | j | G3 | TU8 | 9 | 0 | 0 | 0 | 1,325 | TU7 | 1,329 |
| AT1G01030 | AT1G01030.2 | = | G2 | TU2 | 3 | 0 | 0 | 0 | 1,836 | TU2 | 1,836 |
| AT1G01040 | AT1G01040.1 | = | G5 | TU11 | 20 | 0 | 0 | 0 | 6,276 | TU12 | 6,276 |
| AT1G01046 | AT1G01046.1 | k | G5 | TU12 | 19 | 0 | 0 | 0 | 6,552 | TU12 | 207 |
| ... | | | | | | | | | | | |

*ref_gene_id* et *ref_id* associent les transfrag à leur gène et transcrit correspondant dans les annotations de référence, *class_code* indique le code de classe attribué lors de la comparaison, *qry_gene_id* et *qry_id* indiquent les identifiants des gènes et transcrits dépendant de l'assembleur utilisé (TACO), *num_exons* présente le nombre d'exon(s) contenu(s) dans les transfrags, *FPKM* et *TPM* conservent les expressions, *cov* indique leur couverture calculée par base, *len* indique la taille du transfrag, *major_iso_id* conserve l'identifiant de l'isoforme majoritaire selon le gène auquel le transfrag se rapporte et *ref_match_len* la taille de recouvrement maximum sur la référence.

Table 11 – Structure du fichier *combined*

| Chromosome | Source | Nature | Début | Fin | Score | Brin | Cadre | Attributs |
|---|---|---|---|---|---|---|---|---|
| Chr1 | taco | transcript | 3,631 | 5,930 | . | + | . | transcript_id "TCONS_00000001"; gene_id "XLOC_000001"; gene_name "AT1G01010"; oId "TU2"; cmp_ref "AT1G01010.1"; class_code "="; tss_id "TSS1"; |
| Chr1 | taco | exon | 3,631 | 3,913 | . | + | . | transcript_id "TCONS_00000001"; gene_id "XLOC_000001"; exon_number "1"; |
| Chr1 | taco | exon | 3,996 | 4,276 | . | + | . | transcript_id "TCONS_00000001"; gene_id "XLOC_000001"; exon_number "2"; |
| Chr1 | taco | exon | 4,486 | 4,605 | . | + | . | transcript_id "TCONS_00000001"; gene_id "XLOC_000001"; exon_number "3"; |
| Chr1 | taco | exon | 4,706 | 5,095 | . | + | . | transcript_id "TCONS_00000001"; gene_id "XLOC_000001"; exon_number "4"; |
| Chr1 | taco | exon | 5,174 | 5,326 | . | + | . | transcript_id "TCONS_00000001"; gene_id "XLOC_000001"; exon_number "5"; |
| Chr1 | taco | exon | 5,439 | 5,930 | . | + | . | transcript_id "TCONS_00000001"; gene_id "XLOC_000001"; exon_number "6"; |
| Chr1 | taco | transcript | 23,121 | 31,227 | . | + | . | transcript_id "TCONS_00000004"; gene_id "XLOC_000002"; gene_name "AT1G01040"; oId "TU13"; cmp_ref "AT1G01040.1"; class_code "j"; tss_id "TSS2"; |
| Chr1 | taco | exon | 23,121 | 24,451 | . | + | . | transcript_id "TCONS_00000004"; gene_id "XLOC_000002"; exon_number "1"; |
| Chr1 | taco | exon | 24,542 | 24,655 | . | + | . | transcript_id "TCONS_00000004"; gene_id "XLOC_000002"; exon_number "2"; |
| ... | | | | | | | | |

La première colonne indique le chromosome qui contient l'annotation, la colonne *Source* indique l'outil qui a généré l'information, *Nature* précise la nature de l'annotation (exon, transcrit, gène, ...), *Début* indique la position de départ de l'annotation et *Fin* celle de fin, *Score* reflète l'abondance pour un gène donné où 1000 équivaut à l'isoforme majoritaire, *Brin* révèle la direction du brin, *Cadre* indique le cadre de lecture mais n'est pas utilisé par TACO qui reporte systématiquement un point et la dernière colonne liste les attributs associés aux différentes annotations. Les attributs *gene_id* et *transcript_id* renseignent les identifiants des locus et transcrits attribués par GffCompare, *gene_name* indique le gène de référence auquel il se rapporte dans la référence, *oId* rapporte l'ancien identifiant du transcrit selon TACO, *comp_ref* associe le transfrag avec son transcrit de référence dans les annotations, *class code* indique le code de class attribué lors de la comparaison avec les annotations de référence et *tss_id* précise le site d'initiation de la transcription. Si la nature de l'annotation correspond à un exon, alors *exon_number* précise son numéro au sein du transfrag.

TABLE 12 – Structure du fichier *tracking*

| ID | Locus | Gène \| transcrit de référence | Code | Assemblage 1 (WT_-Fe) | Assemblage 2 (WT_Ctrl) | Assemblage 3 (sr45_-Fe) | Assemblage 4 (sr45_Ctrl) |
|---|---|---|---|---|---|---|---|
| TCONS_00000001 | XLOC_000001 | AT1G01010\|AT1G01010.1 | = | q1 :G1\|TU1\|6\|0\|0\|0\|1688 | q2 :G1\|TU1\|6\|0\|0\|0\|1708 | q3 :G1\|TU1\|6\|0\|0\|0\|1688 | q4 :G2\|TU2\|6\|0\|0\|0\|1719 |
| TCONS_00000002 | XLOC_000002 | AT1G01040\|AT1G01040.1 | = | q1 :G5\|TU11\|20\|0\|0\|0\|6276 | q2 :G4\|TU8\|20\|0\|0\|0\|6276 | q3 :G4\|TU8\|20\|0\|0\|0\|6276 | q4 :G6\|TU14\|20\|0\|0\|0\|6284 |
| TCONS_00000003 | XLOC_000002 | AT1G01046\|AT1G01046.1 | k | q1 :G5\|TU12\|19\|0\|0\|0\|6552 | - | - | q4 :G6\|TU15\|19\|0\|0\|0\|6560 |
| TCONS_00000004 | XLOC_000002 | AT1G01040\|AT1G01040.1 | j | q1 :G5\|TU13\|20\|0\|0\|0\|6463 | - | - | - |
| | | | | ... | | | |

A titre d'exemple, le tableau reprend les 5 premières lignes d'un fichier tracking de Gffcompare. ID correspond à l'identifiant unique du transcrit attribué par Gffcompare. Les transcrits sont rassemblés en locus, identifiés par le préfixe *XLOC* dans la colonne suivante. Si le transfrag peut être associé à un transcrit de référence, le gène et l'isoforme en question sont indiqués. Une nouvelle annotation est représentée par un tiret. La colonne *Code* rapporte le code de classe déterminé lors de la comparaison (voir **Figure 5**). Les quatre dernières colonnes ciblent les transcrits correspondants à travers les fichiers des échantillons passés en argument. Si un groupe présente le transfrag, ses identifiants TACO (ID du gène et du transcrit) sont renseignés. Les attributs séparés par des / correspondent respectivement à la position des fichiers renseignés dans la commande GffCompare, au nombre d'exons du transfrag, aux expressions relatives en FPKM et TPM, à la couverture et, finalement, à la longueur de superposition avec la référence si elle existe. Les échantillons dépourvus d'un transfrag particulier contiennent uniquement un tiret.

Comme présenté dans l'exemple à la **Table 12**, GffCompare ne résout pas la lecture des expressions fournies par TACO automatiquement - les mesures d'expression et de couverture sont nulles. Cette incompatibilité apparente n'a pas été considérée comme problématique dans la mesure où la filtration des transcrits selon leur expression a été réalisée préalablement par TACO.

Le nombre total de transcrits détectés (toutes classes confondues) s'élève à 55,698 pour 111 transcrits correspondants aux miRNAs.

### 4.5.2 Filtration et sélection des nouveaux transcrits

Sur base des résultats produits par GffCompare, le script Perl *selection.pl* a été conçu au cours de ce travail pour filtrer les transfrags sur base des codes de classe d'intérêt et générer leur liste en vue d'enrichir les annotations de référence. De plus, ce script compile les comptes de transfrags inclusifs (détectés dans un échantillon ou groupe d'échantillons donné) et exclusifs (détectés dans un échantillon ou groupe d'échantillons et propres à celui-ci) ainsi que les fréquences des codes de classe pour les combinaisons d'échantillons considérées (**Figure 7**).

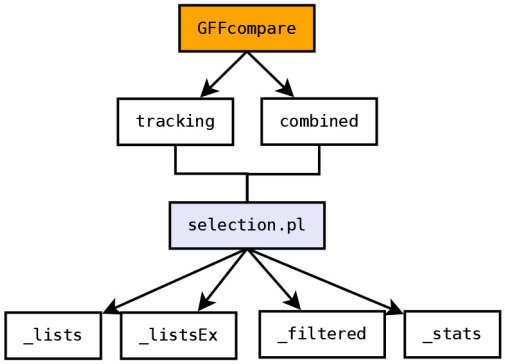

**Figure 7 :** Fichiers produits par selection.pl lors de la sélection des nouveaux transcrits

*selection.pl* attend trois arguments : (i) une liste de codes de classe d'intérêt pour lesquels la filtration est réalisée, (ii) une liste des combinaisons d'échantillons à considérer pour l'analyse (le premier échantillon passé en paramètre correspondant à l'indice 1, des combinaisons telles que 12, 1234 peuvent être renseignées) et (iii) la racine (*basename*) du fichier produit par GffCompare. Quatre fichiers sont produits : *_lists* recense les identifiants et codes de classe des transfrags inclusifs par combinaison d'échantillons (**Table 13**), *_listsEx* est le variant exclusif du premier, *_filtered* contient les transfrags conservés avec leurs annotations et les exons qui les composent (**Table 14**), et le fichier *_stats* compile les informations de comptage exclusif pour les transfrags et les codes de classe à travers les combinaisons d'échantillons (**Table 15**).

TABLE 13 – Structure des fichiers *_lists(Ex)* créés par *selection.pl*

| Combinaison | Gène \| transcrit de référence | Code | ID |
|---|---|---|---|
| 1 | AT5G26980\|AT5G26980.2 | n | TCONS_00048361 |
| 1 | AT5G26990\|AT5G26990.1 | m | TCONS_00044541 |
| | ... | | |
| 34 | AT1G26250\|AT1G26250.1 | o | TCONS_00009126 |
| 34 | AT1G26440\|AT1G26440.1 | n | TCONS_00012824 |
| | ... | | |
| 234 | AT1G01790\|AT1G01790.1 | j | TCONS_00000072 |
| 234 | AT1G02060\|AT1G02060.1 | k | TCONS_00004331 |
| | ... | | |
| 1234 | AT1G01060\|AT1G01060.6 | n | TCONS_00004265 |
| 1234 | - | u | TCONS_0000277 |
| | ... | | |

Les indices de la colonne *Combinaison* correspondent aux assemblages TACO passés en arguments à GffCompare. Par exemple, l'indice 1 cible le premier assemblage, WT_-Fe. La combinaison 1234 sonde les transfrags représentés dans tous les groupes. Les indices 1, 2, 3 et 4 correspondent respectivement aux groupes WT_-Fe, WT_Ctrl, sr45_-Fe et sr45_Ctrl. La deuxième colonne rassemble les informations relatives au gène et transcrit correspondant dans le fichier d'annotations de référence, un tiret est attribué si aucune annotation n'est présente. *Code* présente le code de classe associé et *ID* précise l'identifiant du transfrag selon GffCompare.

TABLE 14 – Structure du fichier *_filtered* créé par *selection.pl*

| Chromosome | Source | Nature | Début | Fin | Score | Brin | Cadre | Attributs |
|---|---|---|---|---|---|---|---|---|
| Chr1 | taco | transcript | 23,121 | 31,227 | . | + | . | transcript_id "TCONS_00000004"; gene_id "XLOC_000002"; gene_name "AT1G01040"; cmp_ref "AT1G01040.1"; class_code "j"; |
| Chr1 | taco | exon | 23,121 | 24,451 | . | + | . | transcript_id "TCONS_00000004"; gene_id "XLOC_000002"; exon_number "1"; |
| Chr1 | taco | exon | 24,542 | 24,655 | . | + | . | transcript_id "TCONS_00000004"; gene_id "XLOC_000002"; exon_number "2"; |
| | | | | | ... | | | |
| Chr2 | taco | transcript | 1,449,288 | 1,455,314 | . | + | . | transcript_id "TCONS_00015346"; gene_id "XLOC_005915"; gene_name "AT2G04235"; oId "TU9018"; cmp_ref "AT2G04235.2"; class_code "m"; |
| Chr2 | taco | exon | 1,449,288 | 1,451,759 | . | + | . | transcript_id "TCONS_00015346"; gene_id "XLOC_005915"; exon_number "1"; |
| Chr2 | taco | exon | 1,452,125 | 1,452,298 | . | + | . | transcript_id "TCONS_00015346"; gene_id "XLOC_005915"; exon_number "2"; |
| | | | | | ... | | | |

La première colonne indique le chromosome qui contient l'annotation, la colonne *Source* indique l'outil qui a généré l'information, *Nature* précise la nature de l'annotation (exon, transcrit, gène, ...), *Début* indique la position de départ de l'annotation et *Fin* celle de fin, *Score* reflète l'abondance pour un gène donné où 1000 équivaut à l'isoforme majoritaire, *Brin* révèle la direction du brin, *Cadre* indique le cadre de lecture mais n'est pas utilisé par TACO qui reporte systématiquement un point et la dernière colonne liste les attributs associés aux différentes annotations. Les attributs *gene_id* et *transcript_id* renseignent les identifiants des locus et transcrits attribués par GffCompare, *gene_name* indique le gène de référence auquel il se rapporte dans la référence, *oId* rapporte l'ancien identifiant du transcrit selon TACO, *comp_ref* associe le transfrag avec son transcrit de référence dans les annotations, *class code* indique le code de class attribué lors de la comparaison avec les annotations de référence et *tss_id* précise le site d'initiation de la transcription. Si la nature de l'annotation correspond à un exon, alors *exon_number* précise son numéro au sein du transfrag.

TABLE 15 – Structure du fichier *_stats* créé par *selection.pl*

| Echantillon(s) | Code | Fréquence |
|---|---|---|
| global | e | 76 |
| global | o | 1376 |
| global | i | 30 |
| global | u | 116 |
| global | * | 22064 |
| | ... | |
| 1 | k | 466 |
| 1 | m | 526 |
| | ... | |
| 1234 | u | 9 |
| 1234 | * | 1136 |

La première colonne indique les indices des fichiers pour lesquels les fréquences ont été calculées. *global* résume les comptes à travers tous les échantillons. Pour chaque indice, les fréquences des codes renseignées dans la commande sont calculées avec leur total, *. Par exemple, l'indice 1 cible le premier assemblage passé à GffCompare, issu de TACO. La combinaison 1234 sonde les transfrags représentés dans tous les groupes. Les indices 1, 2, 3 et 4 correspondent respectivement aux groupes WT_-Fe, WT_Ctrl, sr45_-Fe et sr45_Ctrl.

Un diagramme de Venn qui illustre le nombre de transcrits privatifs aux différents groupes d'échantillons a été construit sur base des listes générées dans le fichier *_lists* (**Figure 8**). De façon absolue, les génotypes mutants ainsi que les conditions de carences présentent plus de variants. Cependant, cette différence varie seulement de 0.6% à 1.6% au maximum, ce qui n'est pas significatif. Le nombre de nouveaux variants privatifs est relativement élevé et représente approximativement 20% des transfrags totaux pour les différents échantillons incluant le groupe *wt - Ctrl*.

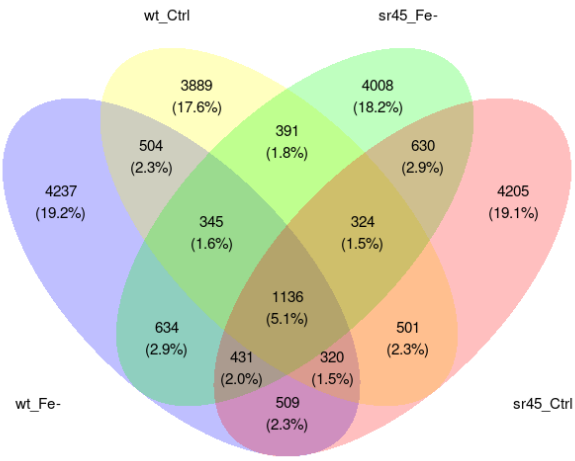


**Figure 8 :** Diagramme de Venn sur base des nouveaux transcrits sélectionnés pour chaque groupe d'échantillons : WT_-Fe (sauvage en condition de carence de fer), WT_Ctrl (sauvage en condition contrôle), sr45_-Fe (mutant *sr45-1* en condition de carence en fer) et sr45_Ctrl (mutant *sr45-1* en condition contrôle)

Les fréquences compilées par le script *selection.pl* sont illustrées à la **Figure 9** et **10**. La distribution des codes de classe à travers les différents échantillons conserve des ratios à peu près constants, avec une majorité en faveur de la classe **k**. Les codes les moins représentés correspondent aux **m**, **o** et **u**. Comme attendu, la classe **n** présente une contribution significative dans la distribution illustrant des évènements de rétention d'introns qui sont prépondérants chez les plantes supérieures [6]. A l'inverse, la classe **m** qui est plus restrictive est moins représentée.

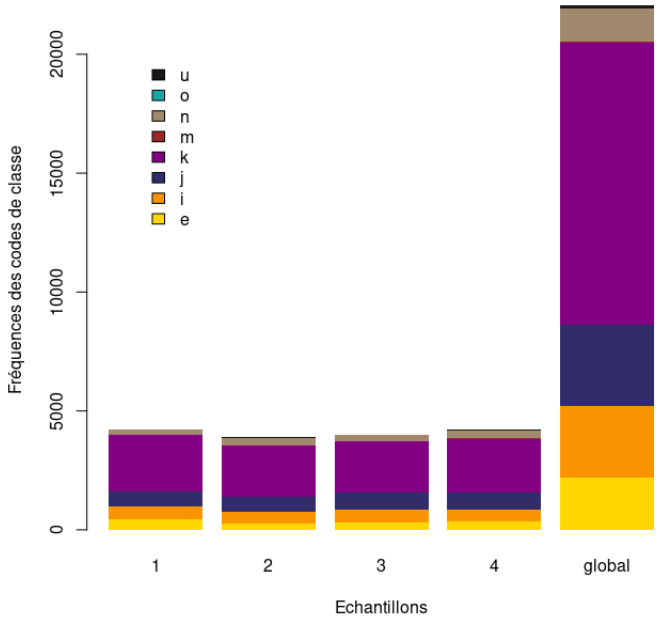


**Figure 9 :** Fréquences des codes de classe de nouveaux variants d'épissage par groupes d'échantillon(s). Les numéros 1, 2, 3 et 4 correspondent resp. aux assemblages WT_Fe, WT_Ctrl, sr45_Fe et sr45_Ctrl.

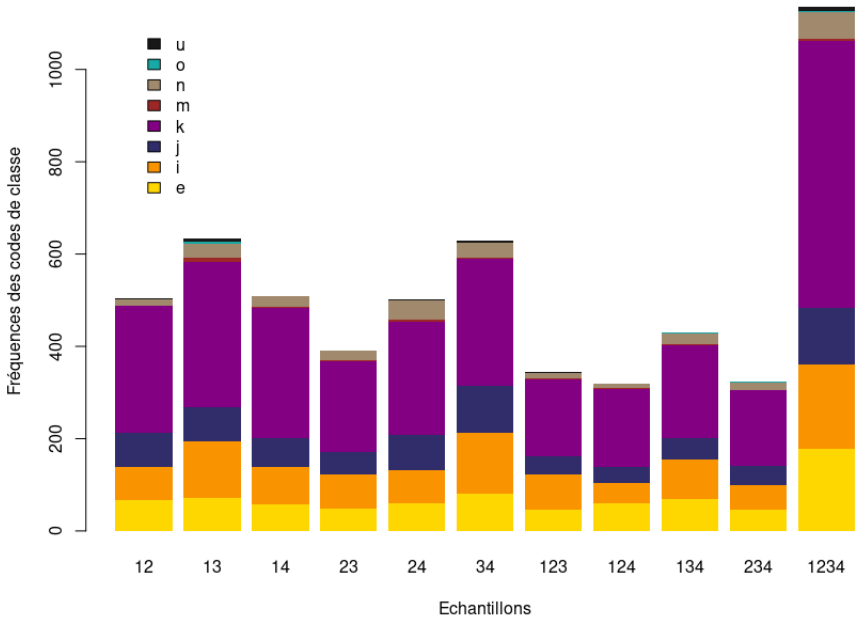


**Figure 10 :** Fréquences des codes de classe de nouveaux variants d'épissage partagés par plusieurs groupes d'échantillon(s). Les numéros 1, 2, 3 et 4 correspondent resp. aux assemblages WT_Fe, WT_Ctrl, sr45_Fe et sr45_Ctrl.

### 4.5.3 Filtration des miRNAs de code *k*

Parmi les nouveaux variants détectés, il s'avère que des variants de miRNAs de codes $k$ aberrants sont produits quand leurs positions génomiques surplombent d'autres transcrits, ce qui conduit à l'attribution des exons de ces derniers aux variants de miRNA qui contiennent alors plusieurs exons.
Les nouveaux variants ont été filtrés sur base des identifiants des miRNAs et des codes de classe $k$ attribués. Au total, 84 nouveaux transcrits relatifs aux miRNAs ont été écartés.

### 4.5.4 Ajustement des annotations

Les annotations des nouveaux variants détectés par GffCompare, sélectionnés sur base des codes de classe d'intérêt et filtrés pour les miRNAs aberrants, présentent parmi leurs attributs des champs *transcript_id*, *gene_id* et *gene_name* (**Table 14**). Le premier et le deuxième champ sont issus de nommages systématiques par GffCompare qui ne respectent pas la convention ATG (Arabidopsis Thaliana Gene) des annotations du projet TAIR. Le troisième champ associe le transfrag à son gène de référence. Dans la suite du protocole, l'association des isoformes avec leur gène respectif est réalisée sur base de l'attribut *gene_id*, tandis que *gene_name* n'est pas comptabilisé. Dans le but d'assurer cette correspondance, le script Perl formatGtf.pl a été écrit au cours de ce travail, répondant à la stratégie suivante : chaque ligne qui désigne un transcrit est sondée ; si les champs *gene_id* et *gene_name* sont détectés, la valeur du second est passée au premier et la modification est réalisée pour les exons qui le composent. Si la structure n'est pas détectée, le transcrit n'a pas d'équivalent dans la référence et n'est pas modifié.

## 4.6 Enrichissement

Les nouveaux variants ont été comptés et ajoutés au fichier d'annotations de référence. La concaténation conduit à la greffe des informations contenues dans les différents fichiers à la suite sans se soucier d'une quelconque évaluation d'ordre.
Les annotations résultantes de la fusion des fichiers ont été triées avec gff3sort.pl avec son paramètre **chr_order** par défaut (tri alphabétique des chromosomes) avec omission du paramètre **precis**, réservé aux annotations gff [39].

## 4.7 Alignement STAR avec les annotations enrichies

Un nouvel alignement STAR est réalisé dans le but d'incorporer les informations relatives aux nouveaux transcrits. Les résultats d'alignement au format BAM sont utilisés par rMATs pour réaliser l'étude des DASGs (*Differentially Alternatively Spliced Genes*) et les fichiers *toTranscriptome.out.bam* par RSEM pour la quantification. Les matrices de comptes générées sont alors passées à DESeq2 afin de mener les analyses d'expressions différentielles (**Figure 11**).

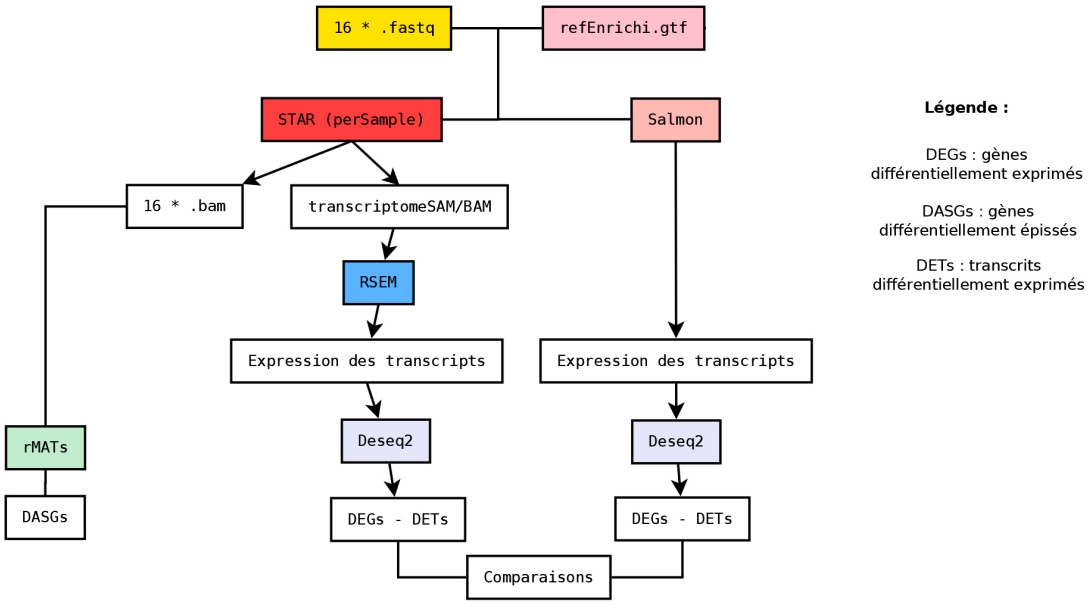


**Figure 11 :** Alignements, quantification, et identification des DEGs (*Differentially Expressed Genes*), DASGs (*Differentially Alternatively Spliced Genes*) et DETs (*Differentially Expressed Transcripts*).

### 4.7.1 Re-indexation du génome avec les annotations enrichies

Le génome d'*A. thaliana* est à nouveau indexé tel que décrit précédemment en renseignant les annotations enrichies.

### 4.7.2 Alignement et quantification avec STAR

Les reads sont alignés pour chaque échantillon, les résultats individualisés serviront à estimer la variabilité biologique entre les groupes d'échantillons (variant selon le génotype ou le traitement) [35]. L'option **quantMode** de STAR active le comptage des reads alignés sur un gène pendant le processus d'alignement. En plus de la quantification, l'option **quantMode** propose un paramètre *TranscriptomeSAM* qui génère conjointement un fichier qui reporte les résultats d'alignement traduit en coordonnées transcriptomiques, un fichier utilisé par certains outils de quantification tels que RSEM. L'identification de ces positions est réalisée *a posteriori* de la procédure d'alignement en tant que tel [53].

## 4.8 Quantification avec RSEM

#### 4.8.0.1 Préparation

En préliminaire à la quantification, RSEM nécessite une étape de préparation où les séquences relatives aux transcrits du génome sont extraites et pré-processées [59]. L'étape de préparation requiert un chemin vers un dossier qui contient un génome de référence à l'extension FA(STA) qui est automatiquement détecté. Le paramètre **gtf** accepte le chemin vers un fichier d'annotations de référence pour guider la construction de l'index. Un troisième argument définit le préfixe des fichiers à créer.

#### 4.8.0.2 Calcul des expressions

Les fichiers d'alignement sont listés après le paramètre **bam**, RSEM accepte également les extensions SAM et CRAM.
**no-bam-output** évite la production de fichiers BAM inutiles en aval. L'information quant à l'orientation du brin à considérer est renseignée par le paramètre **strandedness** avec l'argument *reverse*, correspondant aux résultats issus des kits Illumina Stranded [55]. Lors d'analyses sur des données *single-end*, RSEM propose un paramètre supplémentaire pour affiner le modèle utilisé en considérant la moyenne et l'écart-type des fragments générés lors de l'édification des librairies. Ces informations étant indisponibles, une moyenne de *350* a été utilisée avec le paramètre **fragment-length-mean** correspondant à la valeur typique obtenue pour des kits Illumina mRNA Stranded de bonne qualité [61].
**p** détermine le nombre de processus d'exécution, et les trois derniers arguments indiquent le chemin vers les fichiers d'alignement, vers l'index créé précédemment et le répertoire de réception des résultats [60].

Les résultats des calculs d'expressions sont de deux types bien qu'ils répondent à la même structure : le premier résume les comptes au niveau des gènes tandis que le second précise les comptes pour chaque transcrit. Les deux fichiers listent les gènes/transcrits sur base des annotations et fournissent pour chaque ligne la taille de l'élément considéré, les valeurs d'expressions calculées en FPKM et TPM et, pour les fichiers qui recensent l'information des transcrits, le pourcentage de l'isoforme par rapport à son gène parent.

## 4.9 Analyses d'expressions différentielles

### 4.9.1 DGE/DTE avec DESeq2

Initialement conçu afin de mener des analyses d'expressions différentielles au niveau des gènes, la littérature appuie l'extension de cette méthodologie appliquée au niveau des transcrits [41]. Le tableau résumé des résultats obtenus pour les gènes et transcrits est présenté à la **Table 16**. Les gènes et transcrits différentiellement exprimés dans les racines du WT et du mutant *sr45-1* ont été sélectionnés sur base d'un *FoldChange* >= 2 et d'un FDR <= 0.05 (Annexe xx).

Lorsque la comparaison est basée sur le traitement (carence versus contrôle), 2237 (8.8% des gènes totaux) et 1317 (5.2%) DEGs, respectivement pour les génotypes WT et *sr45-1*, sont identifiés. A l'inverse, le nombre de DEGs pour les comparaisons selon les génotypes sont inférieurs au pourcent et valent respectivement 25 (0.099%) et 86 (0.34%) pour les conditions contrôle et carence en fer. Tandis que les comptes de DEGs et DTEs sont très similaires pour les comparaisons basées sur les conditions de traitement, le nombre de DETs est 5 à 10 fois plus important que leur équivalent DEGs pour les comparaisons entre les génotypes sauvage et mutant *sr45-1*.

Table 16 – Résumé des DEGs et DETs identifiés par DESeq2

| Groupes | | Gènes | | | Transcrits | | |
|---|---|---|---|---|---|---|---|
| Génotypes | Conditions | Up | Down | Total | Up | Down | Total |
| WT | Ctrl vs -Fe | 1,618 (6.4) | 619 (2.4) | **2,237 (8.8)** | 1,640 (2.7) | 789 (1.3) | **2,429 (4)** |
| *sr45-1* | Ctrl vs -Fe | 780 (3.1) | 537 (2.1) | **1,317 (5.2)** | 742 (1.2) | 578 (0.97) | **1,320 (2.17)** |
| WT vs *sr45-1* | Ctrl | 10 (0.04) | 15 (0.059) | **25 (0.099)** | 115 (0.19) | 92 (0.15) | **207 (0.34)** |
| WT vs *sr45-1* | -Fe | 39 (0.15) | 47 (0.19) | **86 (0.34)** | 175 (0.29) | 244 (0.41) | **419 (0.7)** |

Les gènes et transcrits différentiellement exprimés (DEGs, DETs) dans les racines ont été déterminés sur base de quatre réplicats par combinaison de génotype et condition. Les conditions contrôle (Ctrl) et carence en fer (-Fe) correspondent à des concentrations de 10 et 0 µM de fer dans le milieu de culture, respectivement. Les pourcentages de DEGs et DETs par rapport au total des gènes et transcrits sont indiqués entre parenthèses.

## 4.10 Analyse de l'épissage alternatif différentiel

### 4.10.1 ASGs et DASGs avec rMATs

Les gènes alternativement épissés (ASGs) correspondent aux gènes qui subissent un phénomène d'épissage entre deux conditions considérées. Les DASGs partagent cette même définition à l'exception que les évènements d'épissage qui surviennent répondent à des critères de significativité qui impliquent le taux FDR et la valeur deltaPSI de l'évènement. Les résultats de rMATs sont traités et visualisés à travers les fonctions du package Maser. Les évènements supportés par un nombre de reads inférieur à 5 ont été filtrés avec le paramètre **filterByCoverage**. Les évènements significatifs ont été déterminés pour un FDR <= 0.05 et un deltaPSI >= 0.1. Ces valeurs reflètent les standards utilisés dans ce type d'analyses [42], [43]. Le nombre total d'évènements et leur nature par groupe d'échantillons sont présentés à la **Figure 12**. De façon absolue, le groupe *wt_ Fe - s45_ Fe* détient le plus d'évènements d'épissage (864), suivi de près par le groupe *wt_ Ctrl - s45_ Ctrl* (772), reflétant l'impact des géno-

types sur l'épissage. Le nombre d'évènements détectés pour les comparaisons effectuées entre les conditions Ctrl vs Fe- est réduit, et vaut respectivement (557) et (213) pour les génotypes sauvage et mutant. Les contributions des différents évènements restent proportionnelles avec une majorité en faveur des rétentions d'introns, et une minorité pour les évènements d'exclusion mutuelle d'exon.

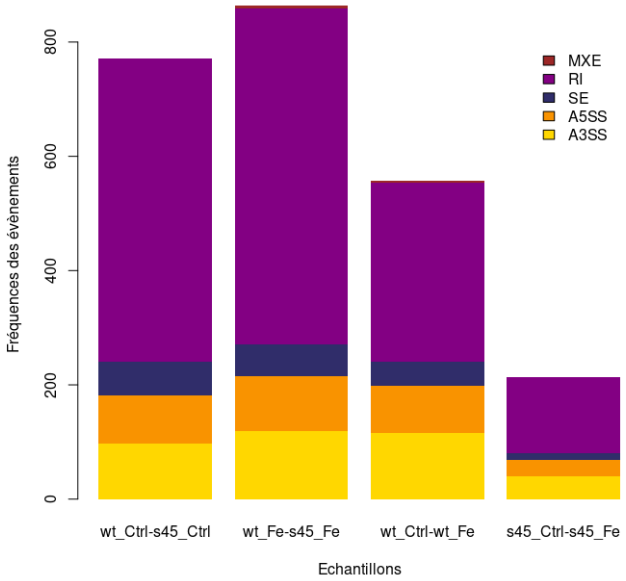


**Figure 12 :** Fréquences des évènements d'épissage en fonction du génotype et du traitement. MXE : Exclusion mutuelle d'exon (*Mutually Exclusive Exons*), RI : Rétention d'intron (*Retained Intron*), SE : Saut d'exon (*Skipped Exon*), A5SS : Site d'épissage alternatif en 5' (*alternative 3' splice junction*) et A3SS : Site d'épissage alternatif en 3' (*alternative 3' splice junction*)

## 4.11 Intégration DEGs-DASGs-ASGs

La **Figure 13** résume l'intégration des résultats des analyses d'expression et d'épissage différentiel. Les résultats des analyses basées sur les génotypes présentent un faible nombre de DEGs, respectivement 25 et 86 pour les traitements contrôle et carence en fer, et pour un nombre de DASGs plus conséquent, de respectivement 536 et 584. A l'inverse, les résultats des comparaisons en fonction des traitements démontrent une tendance inverse - le nombre de DEGs déterminés respectivement pour les génotypes sauvage et mutant est de 2237 et 1317, pour 393 et 181 DASGs. Le nombre d'ASGs détectés à travers les divers groupes est similaire en absolu et concerne approximativement 6800 gènes.

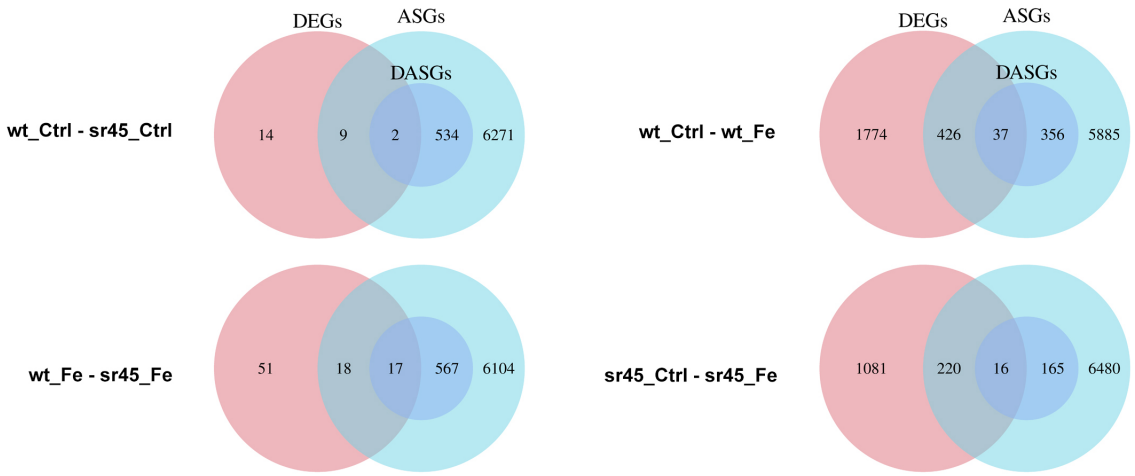


**Figure 13 :** Résultats des analyses DEGs (Gènes Différentiellement Exprimés), ASGs et DASGs (resp. Gènes Alternativement Epissés et Gènes à épissage alternatif différentiel).

# 5 Conclusions et perspectives

Les étapes de détection et de sélection des nouveaux variants présentent dans leur ensemble peu de différences entre les groupes étudiés (**Figure 8**). De façon interpellante, le nombre de nouveaux transcrits rapportés pour les échantillons wt en condition contrôle est aussi élevé que les comptes correspondants aux échantillons de mutants *sr45* et pour les conditions de carence en fer. Ces observations suggèrent des efforts supplémentaires futurs afin d'affiner la caractérisation des transcrits jugés pertinents, qui peut être modulée à plusieurs niveaux : une première filtration est réalisée par TACO lors de la fusion des transcrits reconstruits à travers trois paramètres. **isoform-frac** et **filter-min-expr** ont été laissés par défaut. Tandis que le premier exclut les isoformes dont la contribution dans l'expression totale du gène est inférieure à 5%, le second filtre uniquement les transcrits en-deçà d'un seuil d'expression fixé à 0.5 FPKM. La valeur de ce dernier paramètre constitue toutefois un critère plus stringent comparé à des études qui réalisent des analyses similaires et emploient une valeur seuil de 0.1 FPKM [42],[44]. Le troisième paramètre **filter-min-length**, bien que restreint à son minimum, n'apporte qu'une faible contribution au compte total de nouveaux transcrits de l'ordre du pourcent (**Tables 6** et **7**). Un deuxième niveau d'investigation concerne les codes de classe retenus pour l'analyse GffCompare. Après comparaison avec des études qui s'appuient sur GffCompare pour détecter les nouveaux variants, le code **k**, qui représente la classe la plus représentée (**Figure 9** et **10**), semble être systématiquement exclu [42],[44].

Les résultats relatifs à l'analyse de l'expression différentielle des gènes (**Table 16**) sont supportés par des résultats précédemment publiés [2]. Dans l'ensemble, les DEGs détectés pour les comparaisons basées sur les génotypes concernent quelques dizaines de gènes, 25 et 86 respectivement pour les conditions contrôle et carence. A l'inverse, 2237 et 1317 DEGs sont détectés sur base des traitements, respectivement pour le génotype sauvage et mutant.

Le nombre significativement réduit de DEGs détectés pour le génotype mutant entre les conditions de contrôle et de carence en fer par rapport au génotype sauvage a été interprêté comme un état de carence en fer intermédiaire induit par la mutation du gène *SR45* [2]. Les résultats d'analyses des DTEs (**Table 16**) pour les comparaisons entre conditions démontrent un ratio d'approximativement 1 DTE par DEG. Le ratio augmente drastiquement pour les comparaisons entre génotypes, ce qui suggère l'implication de plusieurs isoformes ?

L'intégration des résultats obtenus pour les analyses des DEGs, DASGs et ASGs (**Figure 13**) présente une tendance suivant les groupes comparés. Alors que les analyses menées sur l'impact des génotypes rapportent un nombre limité de DEGs, les DASGs concernent plus de 500 gènes. A l'inverse, les analyses entre les conditions de contrôle et carence en fer reportent plus de 1000 DEGs pour un nombre de DASGs légèrement plus faible, compris entre 150 et 400 gènes. Ces observations indiquent que le nombre de DASGs est davantage influencé par le génotype de l'individu plutôt que la condition de croissance. A l'inverse, le nombre de DEGs dépend principalement des conditions de traitement. Le nombre de DASGs obtenus pour les analyses entre traitements est inférieur de moitié pour le génotype mutant comparé au sauvage. Cette différence s'inscrit dans le même ordre d'idée que l'observation réalisée au niveau du nombre de DEGs, explicitée plus haut. Finalement, le nombre d'ASGs semble indépendant de ces deux facteurs. L'ensemble de ces résultats permet de répondre à la question initiale de ce travail et de conclure que les évènements d'épissage alternatif chez *Arabidopsis thaliana* sont effectivement conditionnés par la mutation de la protéine SR45. Dans une moindre mesure, la condition de carence en fer démontre également un impact notable sur le nombre de DASGs observés.

Comme représenté sur la **Figure A.0** (Annexe) qui résume le protocole, l'intention initiale consistait à incorporer une branche parallèle reposant sur l'utilisation de Salmon et à comparer les résultats obtenus avec ceux de RSEM. En plus de ses performances supérieures aux alternatives de sa catégorie [33], Salmon est construit autour d'une stratégie de pseudo-alignement. Son incorporation permettrait de by-passer la troisième utilisation de STAR, d'unifier cet alignement à l'étape de quantification tout en diminuant la durée computationnelle nécessaire, inhérent aux méthodes du pseudo-alignement [33]. Salmon, tout comme RSEM, inclut dans ses modèles la détection de biais typiques des séquençages RNA-Seq, tels que le contenu en GC ou les biais de positions. De plus, Salmon apparaît dans des protocoles d'études de l'épissage alternatif différentiel similaires à celui poursuivi [42],[45]. Cette étape du protocole n'a pas été réalisée à cause du manque de temps nécessaire à son application.

Une piste alternative pour améliorer ce protocole viserait à utiliser un autre outil pour l'analyse des DEGs, tel que EdgeR. Les deux outils sont similaires en terme de popularité [46] et reposent sur des modèles binomiaux négatifs pour modéliser les comptes. Une différence majeure repose sur leur stratégie de normalisation respective. EdgeR, en plus de sa capacité à détecter les DEGs, incorpore des fonctions pour tester l'usage différentiel des exons, une faculté qui le rend compétent dans l'approche de l'étude de l'épissage alternatif différentiel basée sur les comptes. De plus, comme présenté pour DESeq2, les méthodes mises en oeuvre pour la détermination des DEGs peuvent raisonnablement s'appliquer à l'étude des DETs [41]. En conséquence, l'utilisation conjointe de EdgeR et rMATs permettrait de balayer les trois approches d'étude de l'épissage alternatif différentiel, ces approches multiples étant encouragées afin d'appréhender de façon complète l'épissage différentiel à travers les échantillons

[41].

Les futures investigations envisagées sur base des résultats obtenus au cours de ce travail incorporent des analyses d'enrichissement des termes de *Gene Ontology* (*GO Enrichment Analysis*) menées sur les listes de DEGs et DASGs détectés entre les différents groupes étudiés. De la sorte, les fonctions rattachées aux gènes pourraient être extraites et comparées. Dans un premier temps, ces analyses permettraient de confirmer la présence de perturbations liées aux fonctions associées aux phénotypes du mutant *sr45-1*, notamment en lien avec l'homéostasie du fer [2], et, dans un second temps, de découvrir de nouvelles fonctions impactées.

En parallèle, la détermination des contributions relatives des nouvelles isoformes au sein des DETs détectés ainsi que leur caractérisation est à considérer. Dans le même ordre d'idée, les transcrits correspondants aux miRNA seraient l'objet d'une analyse afin d'en extraire les interprétations appropriées.

# Références


[1] Denghui Xing, Yajun Wang, Michael Hamilton, Asa Ben-Hur, and Anireddy Reddy. Transcriptome-wide identification of RNA targets of Arabidopsis SERINE/ARGININE-RICH45 uncovers the unexpected roles of this RNA binding protein in RNA processing. *The Plant cell*, 27 :3294–3308, nov 2015.

[2] Steven Fanara, Marie Schloesser, Marc Hanikenne, and Patrick Motte. Altered metal distribution in the sr45-1 Arabidopsis mutant causes developmental defects. *bioRxiv*, page 2021.04.02.438214, jan 2021.

[3] Saurabh Chaudhary, Waqas Khokhar, Ibtissam Jabre, Anireddy S.N. Reddy, Lee J. Byrne, Cornelia M. Wilson, and Naeem H. Syed. Alternative splicing and protein diversity : Plants versus animals, jun 2019.

[4] Benjamín Planells, Isabel Gómez-Redondo, Eva Pericuesta, Patrick Lonergan, and Alfonso Gutiérrez-Adán. Differential isoform expression and alternative splicing in sex determination in mice. *BMC Genomics*, 20(1) :202, mar 2019.

[5] Naeem H Syed, Maria Kalyna, Yamile Marquez, Andrea Barta, and John W. S. Brown. Alternative splicing in plants–coming of age. *Trends in plant science*, 17(10) :616–623, oct 2012.

[6] Runxuan Zhang, Cristiane P. G. Calixto, Yamile Marquez, Peter Venhuizen, Nikoleta A. Tzioutziou, Wenbin Guo, Mark Spensley, Juan Carlos Entizne, Dominika Lewandowska, Sara ten Have, Nicolas Frei dit Frey, Heribert Hirt, Allan B. James, Hugh G Nimmo, Andrea Barta, Maria Kalyna, and John W. S. Brown. A high quality Arabidopsis transcriptome for accurate transcript-level analysis of alternative splicing. *Nucleic Acids Research*, 45(9) :5061–5073, may 2017.

[7] Brian Alloway. Heavy Metals and Metalloids as Micronutrients for Plants and Animals. In *Heavy metals in soils- trace metals and metalloids in soils and their bioavailability*, volume 22, pages 195–209. Blackie Academic and Professional, jan 2013.

[8] Raquel F. Carvalho, Carolina V. Feijão, and Paula Duque. On the physiological significance of alternative splicing events in higher plants. *Protoplasma*, 250(3) :639–650, sep 2013.

[9] James L. Manley and Adrian R. Krainer. A rational nomenclature for serine/arginine-rich protein splicing factors (SR proteins). *Genes and development*, 24(11) :1073–1074, jun 2010.

[10] Gul Shad Ali, Saiprasad G. Palusa, Maxim Golovkin, Jayendra Prasad, James L. Manley, and Anireddy S. N. Reddy. Regulation of plant developmental processes by a novel splicing factor. *PloS one*, 2(5) :e471–e471, may 2007.

[11] Runxuan Zhang, Cristiane P. G. Calixto, Yamile Marquez, Peter Venhuizen, Nikoleta A Tzioutziou, Wenbin Guo, Mark Spensley, Juan Carlos Entizne, Dominika Lewandowska, Sara ten Have, Nicolas Frei dit Frey, Heribert Hirt, Allan B James, Hugh G Nimmo, Andrea Barta, Maria Kalyna, and John W. S. Brown. A high quality Arabidopsis transcriptome for accurate transcript-level analysis of alternative splicing. *Nucleic Acids Research*, 45(9) :5061–5073, may 2017.

[12] Kang Yan, Peng Liu, Chang-Ai Wu, Guo-Dong Yang, Rui Xu, Qian-Huan Guo, Jin-Guang Huang, and Cheng-Chao Zheng. Stress-Induced Alternative Splicing Provides a Mechanism for the Regulation of MicroRNA Processing in Arabidopsis thaliana. *Molecular Cell*, 48(4) :521–531, nov 2012.

[13] Dawid Bielewicz, Malgorzata Kalak, Maria Kalyna, David Windels, Andrea Barta, Franck Vazquez, Zofia Szweykowska-Kulinska, and Artur Jarmolowski. Introns of plant pri-miRNAs enhance miRNA biogenesis. *EMBO reports*, 14(7) :622–628, jul 2013.

[14] Luis A. Corchete, Elizabeta A. Rojas, Diego Alonso-López, Javier De Las Rivas, Norma C. Gutiérrez, and Francisco J. Burguillo. Systematic comparison and assessment of RNA-seq procedures for gene expression quantitative analysis. *Scientific Reports*, 10(1) :19737, nov 2020.

[15] Chi Zhang, Baohong Zhang, Lih-Ling Lin, and Shanrong Zhao. Evaluation and comparison of computational tools for RNA-seq isoform quantification. *BMC Genomics*, 18(1) :583, aug 2017.

[16] Alexander Dobin, Carrie A Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R Gingeras. STAR : ultrafast universal RNA-seq aligner. *Bioinformatics (Oxford, England)*, 29(1) :15–21, jan 2013.

[17] Brendan A Veeneman, Sudhanshu Shukla, Saravana M Dhanasekaran, Arul M Chinnaiyan, and Alexey I Nesvizhskii. Two-pass alignment improves novel splice junction quantification. *Bioinformatics*, 32(1) :43–49, jan 2016.

[18] Stephanie Schaarschmidt, Axel Fischer, Ellen Zuther, and Dirk K Hincha. Evaluation of Seven Different RNA-Seq Alignment Tools Based on Experimental Data from the Model Plant Arabidopsis thaliana, mar 2020.

[19] Ana Conesa, Pedro Madrigal, Sonia Tarazona, David Gomez-Cabrero, Alejandra Cervera, Andrew McPherson, Michał Szcześniak, Daniel Gaffney, Laura Elo, Xuegong Zhang, and Ali Mortazavi. A Survey of Best Practices for RNA-seq Data Analysis. *Genome Biology*, 17, jan 2016.

[20] Mihaela Pertea, Geo M Pertea, Corina M Antonescu, Tsung-Cheng Chang, Joshua T Mendell, and Steven L Salzberg. StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nature biotechnology*, 33(3) :290–295, mar 2015.

[21] Shunfu Mao, Lior Pachter, David Tse, and Sreeram Kannan. RefShannon : A genome-guided transcriptome assembler using sparse flow decomposition. *PloS one*, 15(6) :e0232946–e0232946, jun 2020.

[22] Jung Woo Park and Brenton R. Graveley. Complex alternative splicing. *Advances in experimental medicine and biology*, 623 :50–63, 2007.

[23] Shanrong Zhao, Zhan Ye, and Robert Stanton. Misuse of RPKM or TPM normalization when comparing across samples and sequencing protocols. *RNA (New York, N.Y.)*, 26(8) :903–909, aug 2020.

[24] Tamara Steijger, Josep Abril, Par Engstrom, Felix Kokocinski, Tim Hubbard, Roderic Guigo, Jennifer Harrow, Paul Bertone, Pär Engström, Daniel Zerbino, Stephen Searle, Simon White, Thomas Derrien, David Gonzalez, Julien Lagarde, Michael Sammeth, Sarah Djebali, Martin Akerman, and Thomas Wu. Assessment of transcript reconstruction methods for RNA-seq. *Nat Meth*, advance on, nov 2013.

[25] Yashar S. Niknafs, Balaji Pandian, Hariharan K. Iyer, Arul M. Chinnaiyan, and Matthew K. Iyer. TACO produces robust multisample transcriptome assemblies from RNA-seq. *Nature methods*, 14(1) :68–70, jan 2017.

[26] Po-Yen Wu and May D. Wang. The Selection of Quantification Pipelines for Illumina RNA-seq Data Using a Subsampling Approach. *IEEE-EMBS International Conference on Biomedical and Health Informatics*, 2016 :78–81, feb 2016.

[27] Geo Pertea and Mihaela Pertea. Gff utilities : Gffread and gffcompare. *F1000Research*, 9 :304, apr 2020.

[28] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, and 1000 Genome Project Data Processing Subgroup. The Sequence Alignment/Map format and SAMtools. *Bioinformatics (Oxford, England)*, 25(16) :2078–2079, aug 2009.

[29] Michael N. Edmonson, Jinghui Zhang, Chunhua Yan, Richard P. Finney, Daoud M. Meerzaman, and Kenneth H. Buetow. Bambino : a variant detector and alignment viewer for next-generation sequencing data in the SAM/BAM format. *Bioinformatics (Oxford, England)*, 27(6) :865–866, mar 2011.

[30] Haijing Jin, Ying-Wooi Wan, and Zhandong Liu. Comprehensive evaluation of RNA-seq quantification methods for linearity. *BMC Bioinformatics*, 18(4) :117, mar 2017.

[31] Bo Li, Victor Ruotti, Ron M. Stewart, James A. Thomson, and Colin N. Dewey. RNA-Seq gene expression estimation with read mapping uncertainty. *Bioinformatics*, 26(4) :493–500, feb 2010.

[32] Bo Li and Colin N. Dewey. RSEM : accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics*, 12(1) :323, aug 2011.

[33] Rob Patro, Geet Duggal, Michael I. Love, Rafael A. Irizarry, and Carl Kingsford. Salmon provides fast and bias-aware quantification of transcript expression. *Nature methods*, 14(4) :417–419, apr 2017.

[34] Juliana Costa-Silva, Douglas Domingues, and Fabricio Martins Lopes. RNA-Seq differential expression analysis : An extended review and a software tool. *PloS one*, 12(12) :e0190152–e0190152, dec 2017.

[35] Michael I. Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12) :550, 2014.

[36] Arfa Mehmood, Asta Laiho, Mikko S Venäläinen, Aidan J McGlinchey, Ning Wang, and Laura L Elo. Systematic evaluation of differential splicing tools for RNA-seq studies. *Briefings in Bioinformatics*, 21(6) :2052–2065, dec 2020.

[37] Shihao Shen, Juw Won Park, Zhi-xiang Lu, Lan Lin, Michael D. Henry, Ying Nian Wu, Qing Zhou, and Yi Xing. rMATS : Robust and flexible detection of differential alternative splicing from replicate RNA-Seq data. *Proceedings of the National Academy of Sciences*, 111(51) :E5593 LP – E5601, dec 2014.

[38] Alexander Dobin and Thomas R. Gingeras. Mapping RNA-seq Reads with STAR. *Current protocols in bioinformatics*, 51(1) :11.14.1–11.14.19, sep 2015.

[39] Tao Zhu, Chengzhen Liang, Zhigang Meng, Sandui Guo, and Rui Zhang. GFF3sort : a novel tool to sort GFF3 files for tabix indexing. *BMC Bioinformatics*, 18(1) :482, 2017.

[40] Alessio Adamo, John W. Pinney, Andrea Kunova, David R. Westhead, and Peter Meyer. Heat stress enhances the accumulation of polyadenylated mitochondrial transcripts in Arabidopsis thaliana. *PloS one*, 3(8) :e2889–e2889, aug 2008.

[41] Michael Love, Charlotte Soneson, and Robert Patro. Swimming downstream : statistical analysis of differential transcript usage following Salmon quantification. *F1000Research*, 7 :952, sep 2018.

[42] Chunlan Dong, Fei He, Oliver Berkowitz, Jingxian Liu, Pengfei Cao, Min Tang, Huichao Shi, Wujian Wang, Qiaolu Li, Zhenguo Shen, James Whelan, and Luqing Zheng. Alternative Splicing Plays a Critical Role in Maintaining Mineral Nutrient Homeostasis in Rice (Oryza sativa). *The Plant Cell*, 30(10) :2267–2285, oct 2018.

[43] Guiomar Martín, Yamile Márquez, Federica Mantica, Paula Duque, and Manuel Irimia. Alternative splicing landscapes in Arabidopsis thaliana across tissues and stress conditions highlight major functional differences with animals. *Genome biology*, 22(1) :35, jan 2021.

[44] Sungsam Gong, Francesca Gaccioli, Justyna Dopierala, Ulla Sovio, Emma Cook, Pieter-Jan Volders, Lennart Martens, Paul D. W. Kirk, Sylvia Richardson, Gordon C. S. Smith, and D. Stephen Charnock-Jones. The RNA landscape of the human placenta in health and disease. *Nature Communications*, 12(1) :2639, may 2021.

[45] Yilai Han, Lei Zhu, Li Li, Yanfang Wang, Mingzhu Zhao, Kangyu Wang, Chunyu Sun, Jing Chen, Lingyu Liu, Ping Chen, Jun Lei, Yi Wang, and Meiping Zhang. Characteristics of RNA alternative splicing and its potential roles in ginsenoside biosynthesis in a single plant of ginseng, Panax ginseng C.A. Meyer. *Molecular Genetics and Genomics*, 296(4) :971–983, jul 2021.

[46] Alemu Takele Assefa, Katrijn De Paepe, Celine Everaert, Pieter Mestdagh, Olivier Thas, and Jo Vandesompele. Differential gene expression analysis tools exhibit substandard performance for long non-coding RNA-sequencing data. *Genome Biology*, 19(1) :96, jul 2018.

## Références web et autres

[47] https ://tel.archives-ouvertes.fr/tel-01493669/document

[48] https ://www.bioinformatics.babraham.ac.uk/projects/fastqc/

[49] https ://combine-lab.github.io/salmon/

[50] https ://ccb.jhu.edu/software/stringtie/

[51] M. Hanikenne. Génomique - Cours de master 1, BBMC

[52] https ://www.arabidopsis.org/portals/genAnnotation/gene_structural_annotation/agicomplete.jsp

[53] https ://physiology.med.cornell.edu/faculty/skrabanek/lab/angsd/lecture_notes/STARmanual.pdf

[54] https ://www.biostars.org/p/383115/

[55] https ://rnabio.org/module-09-appendix/0009/12/01/StrandSettings/ ?fbclid=IwAR1S0rjo9qFNKhfVA9vK-IGZ6kCctyKPAtsNxi-xXbWtiE7_NcC6rbo98DE

[56] https ://tacorna.github.io/

[57] https ://ccb.jhu.edu/software/stringtie/gffcompare.shtml

[58] https ://groups.google.com/g/rna-star/c/yo4ULvC5qdU

[59] https ://github.com/deweylab/RSEM

[60] https ://deweylab.github.io/RSEM/rsem-calculate-expression.html

[61] https ://support.illumina.com/content/dam/illumina-support/documents/documentation/chemistry_documentation/illum
stranded-mrna-reference-guide-1000000124518-02.pdf

# A Annexes

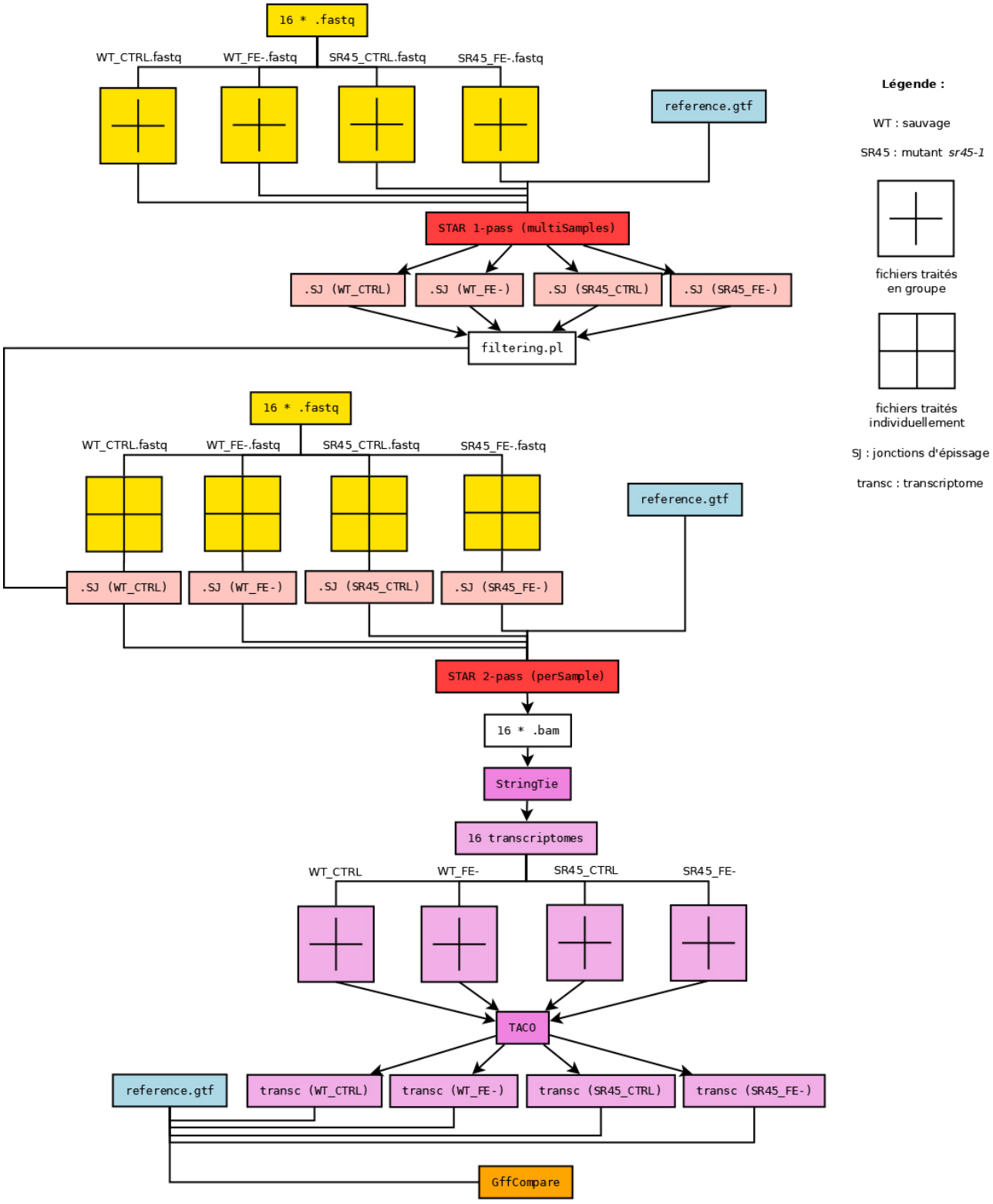


**Figure A.0.1 :** Schéma du protocole global résumé, partie 1

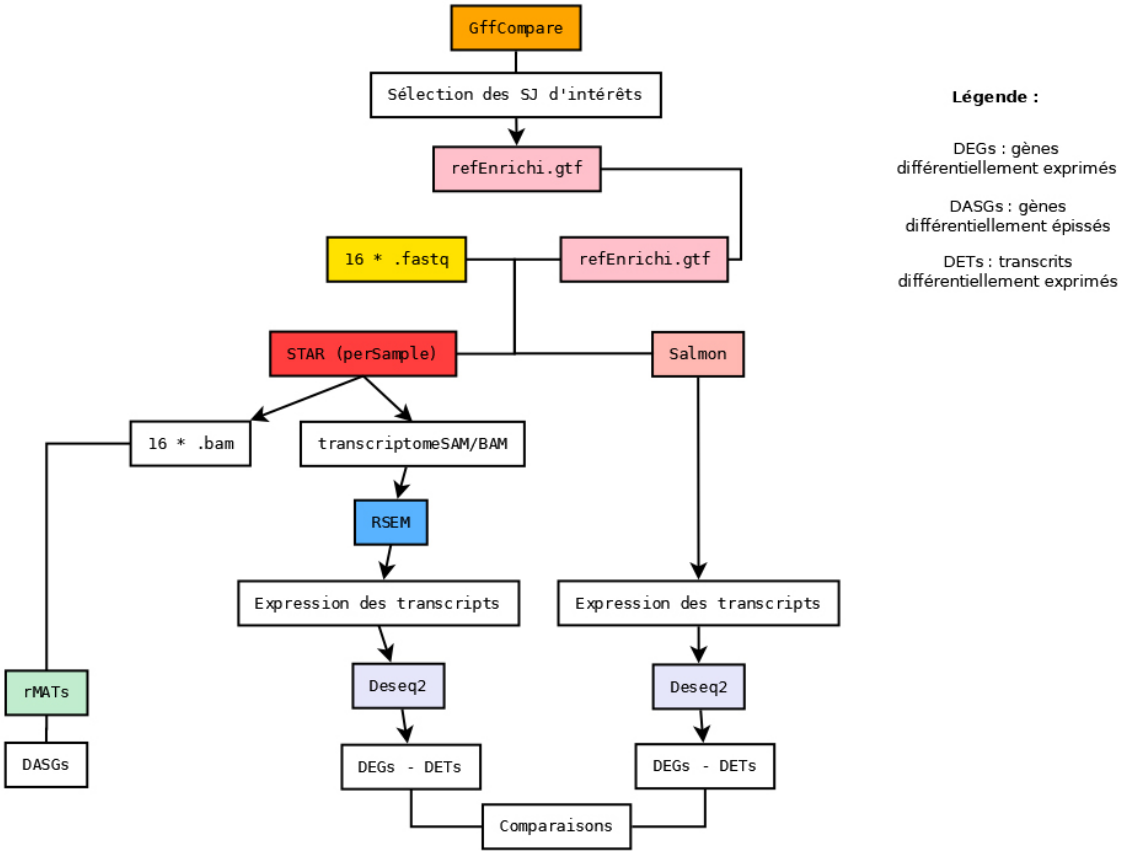


**Figure A.0.2 :** Schéma du protocole global résumé, partie 2

## A.1 Rapport FastQC pré-filtration

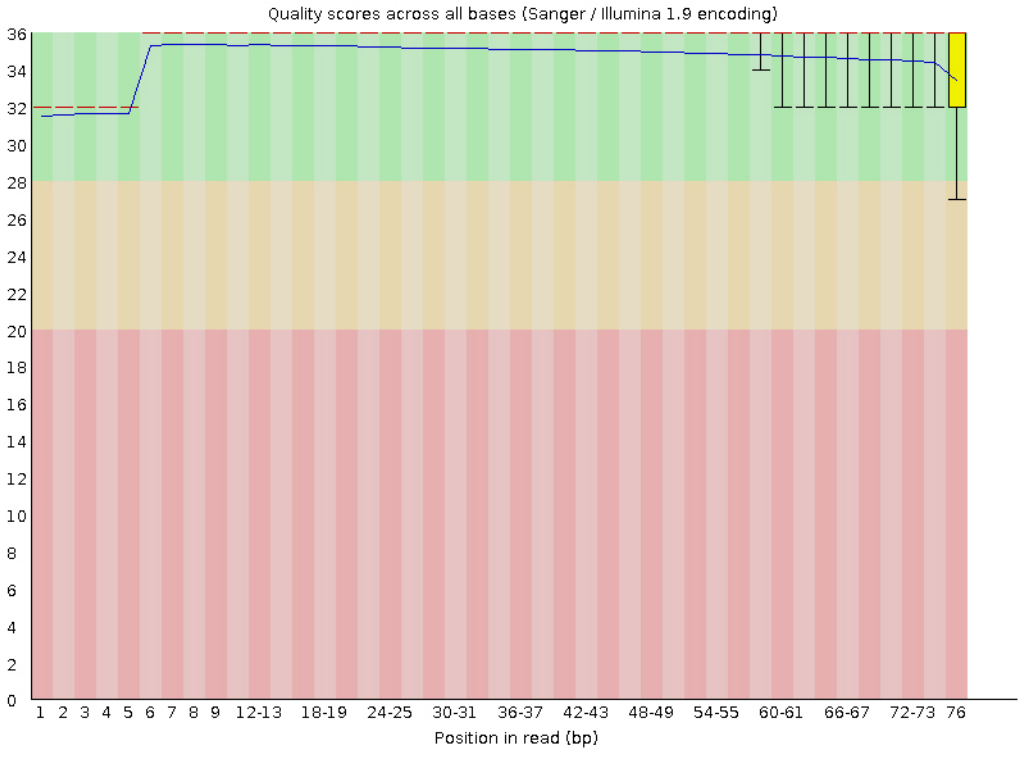

**Figure A.1.1 :** Qualité des séquences par base, avant filtration

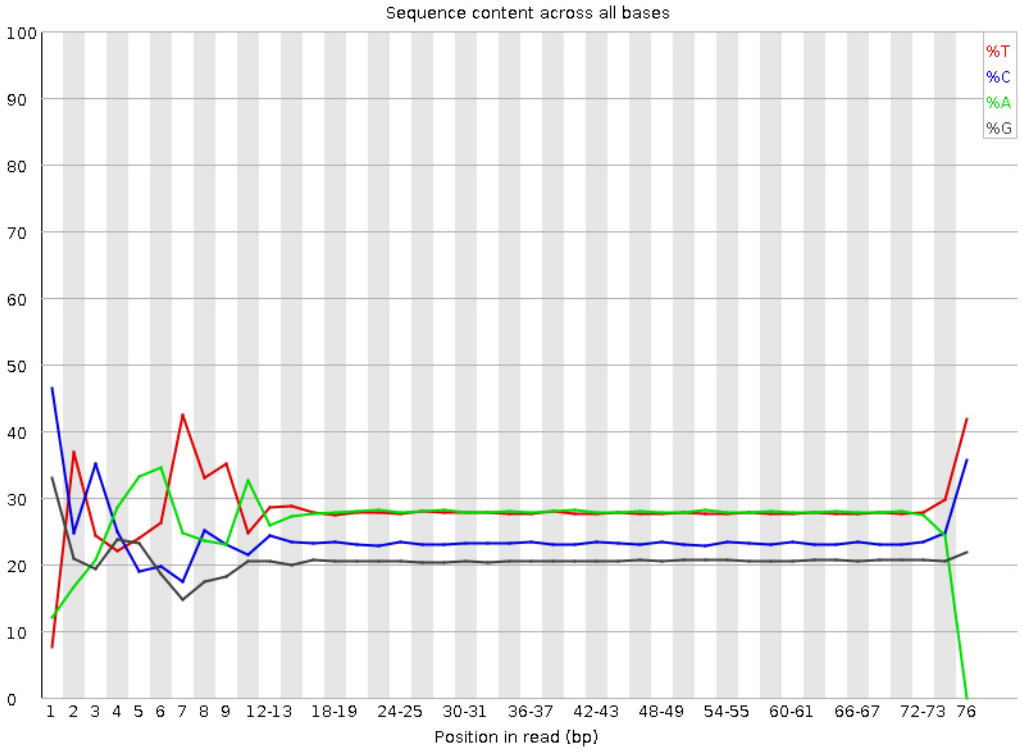


**Figure A.1.2 :** Contenu des séquences par base, avant filtration

## A.2 Rapport FastQC post-filtration

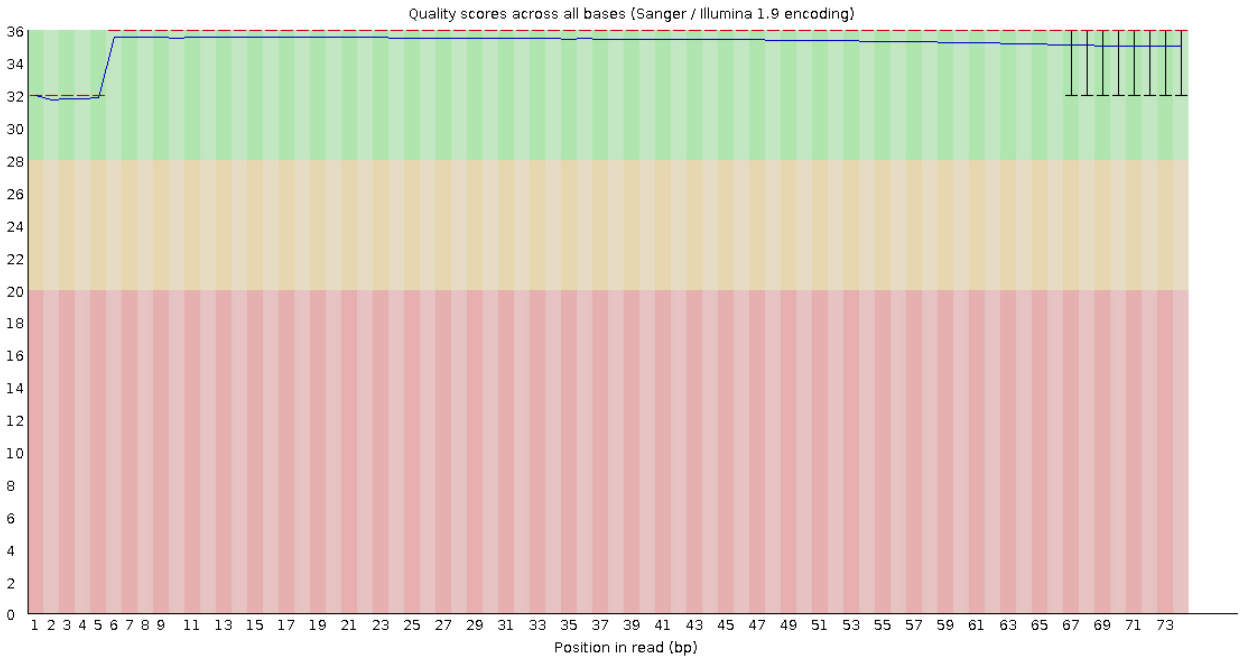


**Figure A.2.1 :** Qualité des séquences par base, après filtration

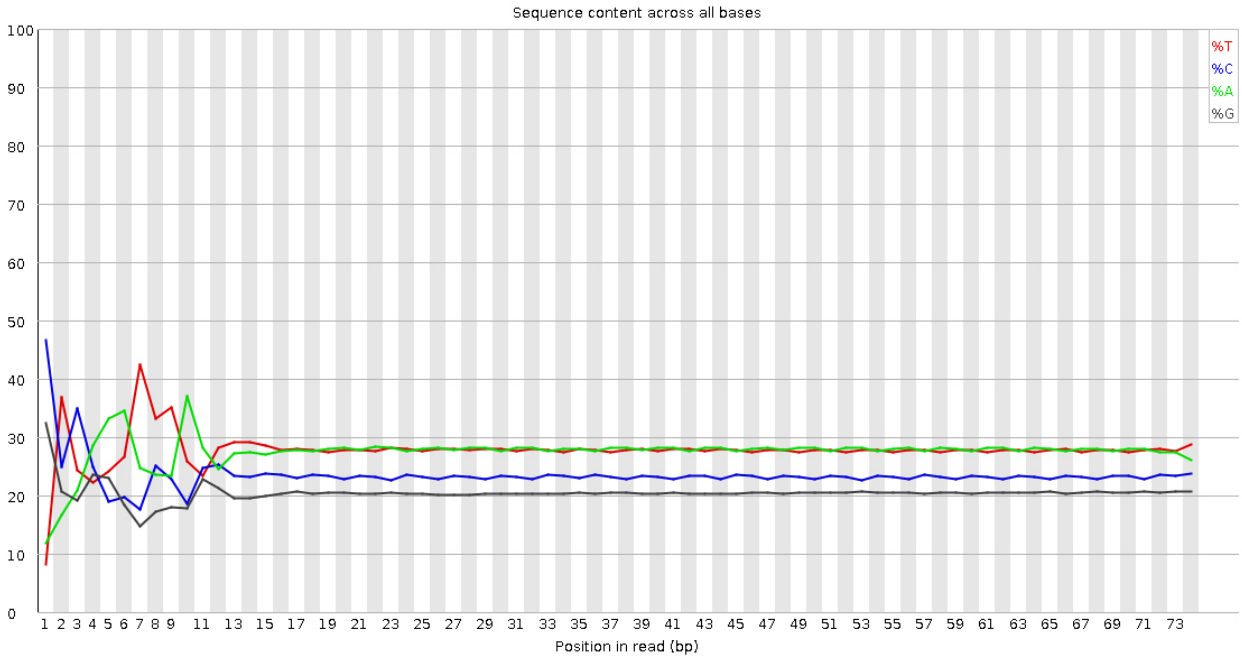


**Figure A.2.2 :** Contenu des séquences par base, après filtration

## A.3 Filtration via Trimmomatic et concaténation

### A.3.1 Template

```
#trimmo.tt
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -q [% queue %]
#$ -m beas
#$ -N trimmo_[% basename %]
java -jar trimmomatic-0.32.jar SE -threads [% threads %] -phred33 path/[% basename %].fastq
    results/[% basename %].trimmed.fastq \
ILLUMINACLIP:/adapters/adapters_hedera.fasta:2:30:20 LEADING:[% leading %] TRAILING:[% trailing
    %] SLIDINGWINDOW:[% slidingwindow %] CROP:[% crop %] MINLEN:[% minlen %]
```

### A.3.2 Définition des scripts

```
for f in `cat samples.ids`; do tpage --define queue=bignode.q --define basename=$f --define
    threads=1 --define leading=26 --define trailing=26 --define slidingwindow=10:26 --define
    crop=74 --define minlen=70 trimmoSE.tt > trimmo_$f.sh; done
```

### A.3.3 Concaténation des fichiers

```
for i in `cat headSamples.id`;do cat "$i"* > ${i}_merged ; done
```

### A.3.4 Extractions des comptes par échantillon

```
grep -c \@NS5 *.fastq | cut -d':' -f2 | perl -nle '$count = $count + $_;$c += 1; if($c==4){$c =
    0;print $count; $count = 0}'
```

## A.4 Premier alignement STAR

### A.4.1 Indexation du génome

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -q bignode.q
#$ -m beas
#$ -N STAR_genomeIndex
STAR --runThreadN 6 \
--genomeSAindexNbases 12 \
--runMode genomeGenerate \
--genomeDir Arab_Gen_Index \
--genomeFastaFiles Arabidopsis.fasta \
--sjdbGTFfile Arabidopsis.gtf \
--sjdbOverhang 73
```

### A.4.2 Alignement par groupe d'échantillons, template

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -q [% queue %]
#$ -m beas
#$ -N STAR_Align
STAR --genomeDir Arab_Gen_Index/ \
--runThreadN 6 \
--readFilesIn [% file1 %], [% file2 %], [% file3 %], [% file4 %] \
--outFileNamePrefix resultsByGC/Aligned_[% file %]
```

## A.5 Filtration des jonctions mitochondriales

### A.5.1 Comptes des jonctions mitochondriales

```
for f in `ls aligned*SJ.out.tab`; do echo $f; grep "ChrM" $f | wc -l; done
```

### A.5.2 Comptes des nouvelles jonctions

```
for f in `ls filtered*`; do echo $f; perl -F"\t" -anle 'if($F[5] eq "0"){print$F[2]}' $f | wc
    -l; done
```

### A.5.3 SJ_filter.pl

```
#!/usr/bin/env perl

use Modern::Perl '2011';
use autodie;
use Smart::Comments;
use Path::Class 'file';
use File::Basename;

unless (@ARGV == 1) {
 die <<"EOT";

Usage: $0 <infile>
This tool is designed to format a file, removing lines with unsatisfied criteria.
Example: $0 file
EOT
}


my $infile = shift;


my ($basename,$dir,$ext) = fileparse($infile,qr{\.[^.]*}xms);
my $outfile = file($dir,$basename . '_filtered');


open my $in, '<', $infile;

open my $out, '>', $outfile;


# Filtering :
# if line[0] == chrM
# line[6] == 0
# line[8] == 0 / line[7] <= 2  # first case corresponds to SJ only supported by multi-mappers


my $cond; # defined by default and set undefined if disrespect one of the criteria
```

```
my $count=0;
my $removed=0;
LINE:
while (my $line = <$in>) {
 chomp $line;
 my @words = split ' ', $line;
 $cond = 1;

 # checking for criteria
 if( ( $words[0] eq 'ChrM') ){  #or ($words[4]==0) or ($words[6]<=2) ){
  $cond = undef;
  $removed++;
 }


 if(defined $cond){
  say {$out} join "\t", @words;
 }
 $count++;


}

my $filtered = $removed/$count;
my $remaining = $count - $removed;
print "Initial: $count, final: $remaining, Filtered: ";
printf("%.2f", $filtered);
print "\n";
```

### A.5.4 Filtration

```
for f in ls aligned*SJ.out.tab; do ../SJ_filter.pl $f;done
```

## A.6 Seconds alignements STAR

### A.6.1 Template

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -q [% queue %]
#$ -m beas
#$ -N STAR_Align2
STAR --genomeDir Arab_Gen_Index/ \
--runThreadN 6 \
```

```
--readFilesIn [% file %] \
--outFileNamePrefix results2passByGC/aligned2pass_[% file %] \
--outSAMtype BAM SortedByCoordinate \
--sjdbFileChrStartEnd resultsByGC/[% SJ %]
```

### A.6.2 Définition

```
$ for n in sed -n '1,4 p' samplesMerged.id; do tpage --define queue=smallnodes.q --define
    file="$n" --define SJ="filtered_14SJ.out.tab" align2passByGC.tt > results2pass/$n.sh; done

$ for n in sed -n '5,8 p' samplesMerged.id; do tpage --define queue=smallnodes.q --define
    file="$n" --define SJ="filtered_58SJ.out.tab" align2passByGC.tt > results2pass/$n.sh; done

$ for n in sed -n '9,12 p' samplesMerged.id; do tpage --define queue=smallnodes.q --define
    file="$n" --define SJ="filtered_912SJ.out.tab" align2passByGC.tt > results2pass/$n.sh; done

$ for n in sed -n '13,16 p' samplesMerged.id; do tpage --define queue=smallnodes.q --define
    file="$n" --define SJ="filtered_1316SJ.out.tab" align2passByGC.tt > results2pass/$n.sh;
    done
```

## A.7 Reconstruction des transcriptomes

### A.7.1 Modification du fichier d'annotations

```
perl -F"\t" -anle 'if( $F[2] eq "miRNA_primary_transcript"
    ){s/miRNA_primary_transcript/exon/}elsif($F[2] eq "miRNA"){s/miRNA/CDS/}print;'
    Arabidopsis.gtf > modifArabidopsis.gtf
```

### A.7.2 StringTie

```
for f in cat samples.id; do stringtie --rf -G ../files/modifArabidopsis.gtf -o ${f}.gtf $f;
    done
```

### A.7.2.1 Exemple d'un résultat de reconstruction avec StringTie

TABLE 17 – Exemple des douze premières lignes obtenues pour la reconstruction de l'échantillon 1 avec StringTie

| Chromosome | Source | Nature | Début | Fin | Score | Brin | Cadre | Attributs |
|---|---|---|---|---|---|---|---|---|
| Chr1 | StringTie | transcript | 3631 | 5899 | 1000 | + | . | gene_id "STRG.1"; transcript_id "STRG.1.1"; reference_id "AT1G01010.1"; ref_gene_id "AT1G01010"; cov "47.760662"; FPKM "28.552521"; TPM "48.318645"; |
| Chr1 | StringTie | exon | 3631 | 3913 | 1000 | + | . | gene_id "STRG.1"; transcript_id "STRG.1.1"; exon_number "1"; reference_id "AT1G01010.1"; ref_gene_id "AT1G01010"; cov "25.897526"; |
| Chr1 | StringTie | exon | 3996 | 4276 | 1000 | + | . | gene_id "STRG.1"; transcript_id "STRG.1.1"; exon_number "2"; reference_id "AT1G01010.1"; ref_gene_id "AT1G01010"; cov "55.120995"; |
| Chr1 | StringTie | exon | 4486 | 4605 | 1000 | + | . | gene_id "STRG.1"; transcript_id "STRG.1.1"; exon_number "3"; reference_id "AT1G01010.1"; ref_gene_id "AT1G01010"; cov "57.316666"; |
| Chr1 | StringTie | exon | 4706 | 5095 | 1000 | + | . | gene_id "STRG.1"; transcript_id "STRG.1.1"; exon_number "4"; reference_id "AT1G01010.1"; ref_gene_id "AT1G01010"; cov "54.005127"; |
| Chr1 | StringTie | exon | 5174 | 5326 | 1000 | + | . | gene_id "STRG.1"; transcript_id "STRG.1.1"; exon_number "5"; reference_id "AT1G01010.1"; ref_gene_id "AT1G01010"; cov "52.418301"; |
| Chr1 | StringTie | exon | 5439 | 5899 | 1000 | + | . | gene_id "STRG.1"; transcript_id "STRG.1.1"; exon_number "6"; reference_id "AT1G01010.1"; ref_gene_id "AT1G01010"; cov "47.379608"; |
| Chr1 | StringTie | transcript | 6788 | 9130 | 1000 | - | . | gene_id "STRG.2"; transcript_id "STRG.2.1"; reference_id "AT1G01020.1"; ref_gene_id "AT1G01020"; cov "11.981360"; FPKM "7.162757"; TPM "12.121337"; |
| Chr1 | StringTie | exon | 6788 | 7069 | 1000 | - | . | gene_id "STRG.2"; transcript_id "STRG.2.1"; exon_number "1"; reference_id "AT1G01020.1"; ref_gene_id "AT1G01020"; cov "18.158529"; |
| Chr1 | StringTie | exon | 7157 | 7232 | 1000 | - | . | gene_id "STRG.2"; transcript_id "STRG.2.1"; exon_number "2"; reference_id "AT1G01020.1"; ref_gene_id "AT1G01020"; cov "24.143866"; |
| Chr1 | StringTie | exon | 7384 | 7450 | 1000 | - | . | gene_id "STRG.2"; transcript_id "STRG.2.1"; exon_number "3"; reference_id "AT1G01020.1"; ref_gene_id "AT1G01020"; cov "21.028894"; |
| Chr1 | StringTie | exon | 7564 | 7649 | 1000 | - | . | gene_id "STRG.2"; transcript_id "STRG.2.1"; exon_number "4"; reference_id "AT1G01020.1"; ref_gene_id "AT1G01020"; cov "24.760557"; |
| ... | | | | | | | | |

La première colonne indique le chromosome qui contient l'annotation, la colonne *Source* indique l'outil qui a généré l'information, *Nature* précise la nature de l'annotation (exon, transcrit, gène, ...), *Début* indique la position de départ de l'annotation et *Fin* celle de fin, *Score* associe un score qualité qui n'est pas utilisé et défini à 1000 pour toutes les annotations, *Brin* révèle la direction du brin, *Cadre* indique le cadre de lecture mais n'est pas utilisé par StringTie qui reporte systématiquement un point et la dernière colonne liste les attributs associés aux différentes annotations. Ces attributs renseignent les identifiants des gènes et transcrits attribués par StringTie, un numéro qui identifie l'exon si la nature de l'annotation correspond à un exon, *reference_id* et *ref_gene_id* associent les éléments avec leurs équivalents dans le fichier d'annotations de référence, *cov* délivre la couverture calculée par base et des mesures d'expressions aux unités FPKM et TPM sont renseignées pour les transcrits.

## A.8 Assemblage des transcriptomes

### A.8.1 Génération des listes de chemins

```
for n in sed -n '1,4 p' ../bamByGC/samples.id; do echo ../bamByGC/$n.gtf; done > pathGtf1_4
for n in sed -n '5,8 p' ../bamByGC/samples.id; do echo ../bamByGC/$n.gtf; done > pathGtf5_8
for n in sed -n '9,12 p' ../bamByGC/samples.id; do echo ../bamByGC/$n.gtf; done > pathGtf9_12
for n in sed -n '13,16 p' ../bamByGC/samples.id; do echo ../bamByGC/$n.gtf; done > pathGtf13_16
```

### A.8.2 Exécution TACO

```
taco_run -p 2 --filter-min-length 0 pathGtf1_4 -o taco1_4
taco_run -p 2 --filter-min-length 0 pathGtf5_8 -o taco5_8
taco_run -p 2 --filter-min-length 0 pathGtf9_12 -o taco9_12
taco_run -p 2 --filter-min-length 0 pathGtf13_16 -o taco13_16
```

### A.8.3 Exemple d'un résultat d'assemblage obtenu avec TACO

TABLE 18 – Exemple des dix premières lignes obtenues pour l'assemblage des échantillons 1 à 4 avec TACO correspondant au groupe WT_-Fe

| Chromosome | Source | Nature | Début | Fin | Score | Brin | Cadre | Attributs |
|---|---|---|---|---|---|---|---|---|
| Chr1 | taco | transcript | 3631 | 5899 | 1000 | + | . | tss_id "TSS1"; locus_id "L1"; abs_frac "1.00000"; rel_frac "1.00000"; expr "50.715"; transcript_id "TU1"; gene_id "G1"; |
| Chr1 | taco | exon | 3631 | 3913 | 1000 | + | . | tss_id "TSS1"; locus_id "L1"; transcript_id "TU1"; gene_id "G1"; |
| Chr1 | taco | exon | 3996 | 4276 | 1000 | + | . | tss_id "TSS1"; locus_id "L1"; transcript_id "TU1"; gene_id "G1"; |
| Chr1 | taco | exon | 4486 | 4605 | 1000 | + | . | tss_id "TSS1"; locus_id "L1"; transcript_id "TU1"; gene_id "G1"; |
| Chr1 | taco | exon | 4706 | 5095 | 1000 | + | . | tss_id "TSS1"; locus_id "L1"; transcript_id "TU1"; gene_id "G1"; |
| Chr1 | taco | exon | 5174 | 5326 | 1000 | + | . | tss_id "TSS1"; locus_id "L1"; transcript_id "TU1"; gene_id "G1"; |
| Chr1 | taco | exon | 5439 | 5899 | 1000 | + | . | tss_id "TSS1"; locus_id "L1"; transcript_id "TU1"; gene_id "G1"; |
| Chr1 | taco | transcript | 6788 | 9130 | 1000 | - | . | tss_id "TSS3"; locus_id "L2"; abs_frac "0.42745"; rel_frac "1.00000"; expr "9.714"; transcript_id "TU3"; gene_id "G3"; |
| Chr1 | taco | transcript | 6788 | 9130 | 661 | - | . | tss_id "TSS3"; locus_id "L2"; abs_frac "0.28237"; rel_frac "0.66060"; expr "6.417"; transcript_id "TU4"; gene_id "G3"; |
| Chr1 | taco | transcript | 6788 | 9130 | 259 | - | . | tss_id "TSS3"; locus_id "L2"; abs_frac "0.11054"; rel_frac "0.25861"; expr "2.512"; transcript_id "TU5"; gene_id "G3"; |
| | | | | | | | | ... |

La première colonne indique le chromosome qui contient l'annotation, la colonne *Source* indique l'outil qui a généré l'information, *Nature* précise la nature de l'annotation (exon, transcrit, gène, ...), *Début* indique la position de départ de l'annotation et *Fin* celle de fin, *Score* reflète l'abondance pour un gène donné où 1000 équivaut à l'isoforme majoritaire, *Brin* révèle la direction du brin, *Cadre* indique le cadre de lecture mais n'est pas utilisé par TACO qui reporte systématiquement un point et la dernière colonne liste les attributs associés aux différentes annotations. Les attributs *gene_id* et *transcript_id* renseignent les identifiants des gènes et transcrits attribués par TACO, *locus_id* associe un identifiant de locus, *tss_id* précise le site d'initiation de la transcription, *expr* délivre l'expression au niveau des isoformes dont les unités dépendent des transfrags étudiés, *rel_frac* exprime l'abondance relative des isoformes par rapport à l'isoforme majoritaire d'un gène (diffère du *Score* par les unités utilisées) et *abs_frac* délivre les abondances relatives des isoformes comparées à l'expression totale de toutes les isoformes pour un gène considéré.

### A.8.4 Résultats des comptes de transfrags filtrés sur base des critères de longueur, d'expression et des jonctions définis

TABLE 19 – Filtration des transcrits avec **filter-min-length** *200* par TACO

| Groupe | sample_id | num_transfrags | filtered_length | filtered_expr | filtered_splice |
|---|---|---|---|---|---|
| WT_-Fe | 1 | 39268 | 492 | 3986 | 0 |
| | 2 | 39405 | 487 | 3908 | 0 |
| | 3 | 39292 | 480 | 3954 | 0 |
| | 4 | 39067 | 494 | 3642 | 0 |
| WT_Ctrl | 1 | 37924 | 452 | 3984 | 0 |
| | 2 | 37733 | 450 | 3971 | 0 |
| | 3 | 38552 | 443 | 4115 | 0 |
| | 4 | 37827 | 455 | 3967 | 0 |
| sr45_-Fe | 1 | 38470 | 515 | 3844 | 0 |
| | 2 | 38574 | 513 | 3700 | 0 |
| | 3 | 39301 | 509 | 3857 | 0 |
| | 4 | 38161 | 510 | 3631 | 0 |
| sr45_Ctrl | 1 | 37327 | 435 | 3681 | 0 |
| | 2 | 42323 | 525 | 6316 | 0 |
| | 3 | 38269 | 458 | 3787 | 0 |
| | 4 | 39007 | 485 | 4073 | 0 |

La première colonne désigne l'échantillon considéré : chaque résultat TACO est issu de l'assemblage supporté par quatre réplicats, indiqués par les indices 1 à 4. *num_transfrags* recense le nombre total de transfrags détectés, les trois dernières colonnes contiennent les comptes des transfrags filtrés respectivement sur base du seuil de taille et d'expression minimales choisies ainsi que la canonicité (filtration de jonctions suivant leur motif).

TABLE 20 – Filtration des transcrits avec **filter-min-length** *0* par TACO

| Groupe | sample_id | num_transfrags | filtered_length | filtered_expr | filtered_splice |
|---|---|---|---|---|---|
| WT_-Fe | 1 | 39268 | 0 | 3986 | 0 |
| | 2 | 39405 | 0 | 3908 | 0 |
| | 3 | 39292 | 0 | 3954 | 0 |
| | 4 | 39067 | 0 | 3642 | 0 |
| WT_Ctrl | 1 | 37924 | 0 | 3984 | 0 |
| | 2 | 37733 | 0 | 3971 | 0 |
| | 3 | 38552 | 0 | 4115 | 0 |
| | 4 | 37827 | 0 | 3967 | 0 |
| sr45_-Fe | 1 | 38470 | 0 | 3844 | 0 |
| | 2 | 38574 | 0 | 3700 | 0 |
| | 3 | 39301 | 0 | 3857 | 0 |
| | 4 | 38161 | 0 | 3631 | 0 |
| sr45_Ctrl | 1 | 37327 | 0 | 3681 | 0 |
| | 2 | 42323 | 0 | 6316 | 0 |
| | 3 | 38269 | 0 | 3787 | 0 |
| | 4 | 39007 | 0 | 4073 | 0 |

La première colonne désigne l'échantillon considéré : chaque résultat TACO est issu de l'assemblage supporté par quatre réplicats, indiqués par les indices 1 à 4. *num_transfrags* recense le nombre total de transfrags détectés, les trois dernières colonnes contiennent les comptes des transfrags filtrés respectivement sur base du seuil de taille et d'expression minimales choisies ainsi que la canonicité (filtration de jonctions suivant leur motif).

### A.8.5 Extraction des expressions

```
for f in ls *.gtf; do a=$(echo $f | cut -d'.' -f1);perl -nle 'print $1 if m/ transcript .*
    transcript_id .* FPKM \s \"(.*)\";\s /mx;' $f >
    FPKM_distributions/FPKM_distribution_$a;done
```

## A.9 Détection et sélection des nouveaux variants

### A.9.1 GffCompare

```
gffcompare -R -r Arabidopsis.gtf -o gffcompAll_R taco1_4/assembly.gtf taco5_8/assembly.gtf
    taco9_12/assembly.gtf taco13_16/assembly.gtf
```

### A.9.2 Comptes des transcrits

```
wc -l gffcompare_results.tracking
```

### A.9.3 Comptes des miRNAs

```
for l in `cat ../files/miRNAs.id`; do grep $l gffcompAll_R.combined.gtf | perl -nle 'print $1
    if m/ cmp_ref \s \" ( .*? ) \" ; /xms';done | sort | uniq > miRNAs_combined.id
```

### A.9.4 selection.pl

#### A.9.4.1 Code

```
#!/usr/bin/env perl

use Modern::Perl '2011';
use autodie;
use Smart::Comments '####';
use File::Basename;
use Path::Class 'file';
use Array::Utils qw(:all);

unless (@ARGV > 0) {
 die <<"EOT";


Usage: $0 <classCodes.file> <comb.file> <toAnalyze.file>
This script requires three files : the first one contains classCodes attributed by gffCompare
    that
should be conserved (other class codes are filtered), the second file contains sample
    combinations of
interest and the third one corresponds to file.tracking produced by gffCompare after
    comparisons.
```

```
Outputs are derived from toAnalyze.file basename : result of filtration (_filtered), stats
    (_stats),
inclusive and exclusive list of transcripts resp. (_list) and (_listEx).
Example: $0 file1 file2 file3
Pratically: ./selection.pl classCodes.txt comb.txt gffcompResults
EOT
}

# lecture du fichier qui contient les codes d'interets et stockage
# dans un array "classCodes"

my $classCodes = shift;

open my $in, '<', $classCodes;

my @classCodes; # accepte les codes de classe

CLASSCODES:
while (my $line = <$in>) {
 chomp $line;
 push @classCodes, $line;
}
close $in;



# lecture du fichier 'comb.txt', extraction des fichiers passes
# en argument et des combinaisons a considerer

my $combFile = shift;

open $in, '<', $combFile;

my @comb; # accepte les combinaisons de fichiers
my @levelsComb; # les differents fichiers a comparer
my $isNew; # booleen
my @diff;

COMB:
while (my $line = <$in>) {
 chomp $line;
 push @comb, $line;

 my @comb_unit = split //, $line;

 $isNew = 1;

```

```
 @diff = array_minus(@comb_unit,@levelsComb);

 if(@diff){
  push @levelsComb, @diff;
 }

}
close $in;

# lecture du fichier '*.tracking'

# col 0 = identifiant unique
# col 1 = locus
# col 2 = gene/transcrit de reference
# col 3 = code

my $infile = shift;

# creation des noms de fichiers en ajoutant des postfixes

my ($basename,$dir, $ext) = fileparse($infile, qr{\.[^.]*}xms);
my $inTracking = file($dir, $basename . '.tracking');
my $inCombined = file($dir, $basename . '.combined.gtf');
my $outCmbFiltered = file($dir, $basename . '.combined_filtered');
my $outStats = file($dir, $basename . '_stats');
my $outList = file($dir, $basename . '_list');
# inclusif
my $outListEx = file($dir, $basename . '_listEx');
# exclusif


# initialisation des hashes
my %classCodesByComb; # recense les informations par combinaison de fichier(s)

for my $comb (@comb){
 for my $class (@classCodes){
  $classCodesByComb{$comb}{$class} = 0;
 }
}

my %classCodesGlobal; # recense les informations globales

for my $class (@classCodes){
 $classCodesGlobal{$class} = 0;
}

# lecture du fichier '*.combined'
```

```
111
112 my $thisCode;
113 my $thisTranscript;
114 my $codeOfInterest; # booleen : a conserver?
115
116 open $in, '<', $inCombined;
117 open my $out, '>', $outCmbFiltered;
118
119 COMBINED:
120 while (my $line = <$in>) {
121  chomp $line;
122  my @words = split "\t", $line;
123
124  $words[8] =~ m/ .*  class_code \s+ \" ( .*? ) \" .* /xms;
125
126  # l'attribut classCode n'est defini que pour les transcrits (pas les exons)
127  if(defined $1){
128
129   $codeOfInterest = 0;
130   $thisCode = $1;
131
132   for $classCodes (@classCodes){
133
134    if($thisCode eq $classCodes){
135     $codeOfInterest = 1;
136    }
137   }
138
139  } # obtention du nom du transcrit et son interet
140
141  if($codeOfInterest){
142
143   say {$out} join "\t", @words;
144  }
145
146 }
147
148 close $out;
149
150
151
152 my $isRepresented; # booleen
153 my $correspondance = 3;
154 # accorde les fichiers consideres (1,2,3,4) et
155 # les indices de colonnes de *.tracking
156 my $ind;
157
```

```
158 my @list;
159 my @listEx; # equivalent exlusif
160
161 my @filtered;
162
163 open $in, '<', $inTracking;
164
165 TRACKING:
166 while (my $line = <$in>) {
167  chomp $line;
168
169  my @words = split "\t", $line;
170
171  # compter le nombre de codes avec un hash,
172  # $words[3] capture les codes
173  $classCodesGlobal{$words[3]}++;
174
175  # le code du transcrit est-il d'interet?
176  $codeOfInterest = 0;
177
178  for $classCodes (@classCodes){
179   if($words[3] eq $classCodes){
180    $codeOfInterest = 1;
181   }
182  }
183
184  # iteration suivante si le code n'est pas d'interet
185  next TRACKING if !$codeOfInterest;
186
187
188  # correspondance :
189  # l'information relative aux fichiers dans tracking est notee
190  # des colonnes 4 -> 7, et de 1 -> 4 dans @comb <=>
191  # le fichier 1 correspond au 4 de tracking, y = x + 3
192
193  for my $comb (@comb){
194   $isRepresented = 1;
195   my @comb_unit = split //, $comb;
196
197   # diff capture les echantillons non repris dans la combinaison
198   @diff = array_minus(@levelsComb,@comb_unit);
199
200   # comptes inclusifs
201
202   for my $comb_unit (@comb_unit){
203
204    $ind = $comb_unit + $correspondance;
```

```
205
206     # capture des lignes exempte d'un '-' note => le transcrit
207     # est represente pour ces fichiers
208     if($words[$ind] eq '-'){
209       $isRepresented = 0;
210     }
211    }
212
213    if($isRepresented){
214     push @list, [$comb, $words[2], $words[3], $words[0]];
215     # resultat inclusif
216    }
217
218    # comptes exlusifs
219
220    # la representation des transcrits pour une combinaison consideree
221    # ne doit pas etre partagee pour d'autres combinaison
222     for my $diff (@diff){
223
224      $ind = $diff + $correspondance;
225
226      if($words[$ind] ne '-'){
227       $isRepresented = 0;
228      }
229     }
230
231    # addition de la condition
232     if($isRepresented){
233      $classCodesByComb{$comb}{$words[3]}++;
234      push @listEx, [$comb, $words[2], $words[3], $words[0]]; # modif
235     }
236
237   }
238
239   push @filtered, $words[0];
240 }
241
242
243 # tri des listes par combinaison
244 my @listSorted = sort {$a->[0] <=> $b->[0] || $b->[1] cmp $a->[1]} @list;
245 my @listSortedEx = sort {$a->[0] <=> $b->[0] || $b->[1] cmp $a->[1]} @listEx;
246
247 open my $outL, '>', $outList;
248
249 # ecriture dans leur fichier respectif
250
251 for my $elemOutL (@listSorted){
```

```
 say {$outL} join "\t", @$elemOutL;
}

close $outL;
open my $outLEx, '>', $outListEx;

for my $elemOutLEx (@listSortedEx){
 say {$outLEx} join "\t", @$elemOutLEx;
}

close $outLEx;

open $out, '>', $outStats;

# infos generales
my $totGlobal=0;
for my $class (@classCodes){

 say {$out} join "\t", 'global', $class, $classCodesGlobal{$class};
 $totGlobal += $classCodesGlobal{$class};
}
say {$out} join "\t", 'global', '*', $totGlobal;


my $totalByComb;

for my $comb (@comb){

 $totalByComb = 0;

 for my $class (@classCodes){

  $totalByComb += $classCodesByComb{$comb}{$class};
  say {$out} join "\t", $comb, $class, $classCodesByComb{$comb}{$class};

 }

 say {$out} join "\t", $comb, '*', $totalByComb;
}
```

#### A.9.4.2 Exécution

```
./selection.pl classCodes.txt comb.txt gffcompAll_R
```

### A.9.5 Diagramme de Venn

```
library(ggvenn)
```

```
2
3 data<-read.table("gffcompAll_R_lists",sep="\t")
4 # la liste contient quatre colonnes :
5 # 1 = quel(s) echantillon(s) le supporte(nt)
6 # 2 = nom du gene|transcrit de reference (si annote)
7 # 3 = code de classe, 4 = ID du transcrit
8
9 lev<-levels(as.factor(data[,1]))[1:4]
10 # recuperation des niveaux
11
12 f<-function(l){
13   r<-data$V4[data$V1==l]
14   r
15 }
16
17 x<-sapply(lev,f)
18
19 # extraction des ID des transcrits en fonction
20 # des niveaux, x est une liste de quatre colonnes
21
22 ggvenn(
23   x,  stroke_size = 0.5, set_name_size = 4
24 )
```

### A.9.6 Filtration des miRNAs 'k'

#### A.9.6.1 Code miRNAs_filter.pl

```
1 #!/usr/bin/env perl
2
3 use Modern::Perl '2011';
4 use autodie;
5 use Smart::Comments '####';
6 use File::Basename;
7 use Path::Class 'file';
8 use Array::Utils qw(:all);
9
10 unless (@ARGV > 0) {
11  die <<"EOT";
12
13 Usage: $0 <classCode.character> <AnnotationsToFilter.file> <transcripts.list>
14 This script requires three arguments : a class code, an annotations file to filter and
15 a list of transcripts.
16 Annotations matching transcript names from list AND the class code given will be extracted to
17 _removed file, filtered annotations are output in _filtered file.
18 Example: $0 character file listOfStrings
19 Pratically: ./selection.pl k gffcompAll_R.combined_filtered miRNAs.id
```

```
EOT
}


my $classCode = shift;
my $inFile = shift;
my $miRNAsFile = shift;

my @miRNAs;

# extraction des miRNAs

open my $in, '<', $miRNAsFile;

MIRNAS:
while (my $line = <$in>) {
        chomp $line;
        push @miRNAs, $line;
}
close $in;

# creation des fichiers resultats avec postfixes
my ($basename,$dir, $ext) = fileparse($inFile, qr{\.[^.]*}xms);
my $outFiltered = file($dir, $basename . '.miRNAs_k_filtered');
my $outRemoved = file($dir, $basename . '.miRNAs_k_removed');

my $codeOfInterest; # boolean
my $thisCode;
my $thisTranscript;
my $kept=0;
my $removed=0;

open $in, '<', $inFile;
open my $out, '>', $outFiltered;
open my $outR, '>', $outRemoved;

# ### $classCode

COMBINED:
while (my $line = <$in>) {
 chomp $line;
 my @words = split "\t", $line;

 $words[8] =~ m/

 .* cmp_ref \s+ \" ( .*? ) \" ; \s+ class_code \s+ \" ( .*? ) \" .*

```

```
                /xms;

 if(defined $1 && defined $2){
 # si $1 defini, alors ce n'est ni une nouvelle annotation, ni un exon
 # si $2 defini, alors ce n'est pas une ligne correspondant a un transcrit
 # mais a un de ses exons
  $codeOfInterest = 1;
  $thisTranscript = $1;
  $thisCode = $2;

  for my $miRNA (@miRNAs){

   if($thisTranscript eq $miRNA){ # ce transcrit correspond a un miRNA
    if($thisCode eq $classCode){ # le code de class correspond au code
           # passe en argument
     $codeOfInterest = 0;
    }
   }
  }
 }

 if($codeOfInterest){

  say {$out} join "\t", @words;
  if($words[2] eq "transcript"){
   $kept++;
  }
 }else{
  say {$outR} join "\t", @words;
  if($words[2] eq "transcript"){
   $removed++;
  }
 }
}

say "filtered: ", $removed, ", conserved: ", $kept;
```

#### A.9.6.2 Extraction des miRNAs à partir des annotations

```
perl -F"\t" -anle 'if($F[2] eq "miRNA_primary_transcript" ){print $1 if $F[8] =~ m/
    transcript_id \s \"(.*?)\" ; /mxs}' Arabidopsis.gtf > miRNAs.id
```

#### A.9.6.3 Exécution

```
./miRNAs_filter.pl k gffcompAll_R.combined_filtered miRNAs.id
```

#### A.9.6.4 Comptes

```
perl -nle 'print $1 if m/ transcript .* transcript_id \s \"(.*?)\";\s .* /mx;'
    newTranscripts.gtf | sort | uniq | wc -l
perl -nle 'print $1 if m/ transcript .* transcript_id \s \"(.*?)\";\s .* /mx;'
    newTranscripts.miRNAs_k_filtered.gtf | sort | uniq | wc -l
```

### A.9.7 FormatGtf.pl

#### A.9.7.1 Code

```
#!/usr/bin/env perl

use Modern::Perl '2011';
use autodie;
use Smart::Comments '####';
use File::Basename;
use Path::Class 'file';
use Array::Utils qw(:all);

unless (@ARGV > 0) {
 die <<"EOT";
This script requires an annotation file generated by GffCompare. The value of "gene_id"
    attribute
will be replaced by "gene_name" value unless that field does not exist.
Usage: $0 <file>
Example: $0 file
Pratically: ./formatGtf.pl my.gtf > formated.gtf
EOT
}


my $infile = shift;

open my $in, '<', $infile;

# lecture de gene_name et modification de l'attribut gene_id

my $currentTairGene;
my $currentTacoGene;
my $noChange; # booleen

LINE:
while (my $line = <$in>) {
 chomp $line;
 my @words = split "\t", $line;

 $noChange=1;
```

```
37
38  if($words[2] eq "transcript"){
39   $words[8] =~ m/ gene_id \s+ \" ( .*? ) \"; \s+ gene_name \s+ \" ( .*? ) \"; /xms;
40
41   if(defined $2){
42    $currentTacoGene = $1;
43    $currentTairGene = $2;
44   }else{
45    $noChange=0; # si $2 non-defini
46   }
47  }
48
49  if($noChange){
50   $words[8] =~ s/$currentTacoGene/$currentTairGene/;
51  }
52
53  say join "\t", @words;
54
55 }
```

#### A.9.7.2 Execution

```
1 ./formatGtf.pl newTranscripts.gtf > nTGeneIdRename.gtf
```

## A.10 Enrichissement des annotations de référence

### A.10.1 Comptes des nouveaux transcrits

```
1 perl -nle 'print $1 if m/ transcript .* transcript_id \s \"(.*?)\";\s .* /mx;'
      nTGeneIdRenamed.gtf | sort | uniq | wc -l
```

### A.10.2 Concaténation

```
1 (cat modifArabidopsis.gtf && cat nTGeneIdRenamed.gtf) > enriched.gtf
```

### A.10.3 Tri

```
1 ./gff3sort.pl enriched.gtf > enrichedSorted.gtf
```

## A.11 Alignement STAR avec les annotations enrichies

### A.11.1 Re-indexation du génome

```
1  #!/bin/bash
2  #$ -S /bin/bash
3  #$ -V
```

```
#$ -cwd
#$ -q bignode.q
#$ -m beas
#$ -N STAR_genomeIndex

STAR --runThreadN 6 \
--genomeSAindexNbases 12 \
--runMode genomeGenerate \
--genomeDir Arab_Gen_Index_Enriched \
--genomeFastaFiles Arabidopsis.fasta \
--sjdbGTFfile enriched.gtf \
# correspond a enrichSorted.gtf
--sjdbOverhang 73
```

### A.11.2 Template pour l'alignement et la quantification STAR

```
#!/bin/bash
#$ -S /bin/bash
#$ -V
#$ -cwd
#$ -q [% queue %]
#$ -m beas
#$ -N STAR_Align

STAR --genomeDir ../Arab_Gen_Index_Enriched/ \
--runThreadN 6 \
--readFilesIn [% file %] \
--outFileNamePrefix resultsEnriched/Enriched_[% file %] \
--outSAMtype BAM Unsorted SortedByCoordinate \
--quantMode TranscriptomeSAM GeneCounts
```

### A.11.3 Définition des scripts

```
for n in `cat samplesMerged.id`; do tpage --define queue=bignode.q --define file="$n"
    alignEnriched.tt > alignScriptsEnriched/Enriched_$n.sh; done
```

## A.12 RSEM

### A.12.1 Préparation

```
rsem-prepare-reference --gtf enrichedSorted.gtf pathToRefFasta/ ref/A_ref
```

### A.12.2 Calculs des expressions et nommage

```
for l in `cat samples.id`; do a=$(echo -n "sample"; echo $l | perl -nle 'print $1 if
    m/-(\d+)_/xms'); rsem-calculate-expression --bam --no-bam-output --strandedness reverse
    --fragment-length-mean 350 -p 16 toTranscriptomeBam/${l} ref/A_ref ${a}; done
```

## A.13 rMATs

### A.13.1 Génération des groupes de tests

```
# x et y indiquent les numeros d'echantillons a collecter
for l in `ls *.bam | sed -n 'x,y p'`; do echo -n bamEnriched/$l,; done > b_file.txt; truncate
    -s-1 b_file.txt
```

### A.13.2 Commande rMATs

```
python rmats.py --b1 b1.txt --b2 b2.txt --gtf enrichedSorted.gtf -t single
    --variable-read-length --allow-clipping --readLength 74 --nthread 16 --libType
    fr-firststrand --od rMATsResults/ --tmp TMP/
```

## A.14 Résultats des DEGs, DETs, DASGs et ASGs

### A.14.1 Script R illustrant les protocoles d'analyses, leur intégration et la génération des graphiques

```
library("tximportData")
library("tximport")
library("DESeq2")
library("limma")
library("gplots")
library("RColorBrewer")
library("genefilter")
library("ggplot2")
library("maser")
library("rtracklayer")
library("ggvenn")
library("Vennplots")
library("eulerr")
library("VennDiagram")

# DEGs :

# wt_fe vs wt_ctrl

txi_wt_fe_wt_ctrl<-tximport(c("sample1.genes.results","sample2.genes.results",
                              "sample3.genes.results","sample4.genes.results",
                              "sample5.genes.results","sample6.genes.results",
                              "sample7.genes.results","sample8.genes.results"),
```

```
                                type="rsem", txIn = FALSE, txOut = FALSE)

genot=factor(rep(c("WT"),each=8))
treat=factor(c(rep(c("Fe"),each=4),rep(c("Ctrl"),each=4)))

countData<-txi_wt_fe_wt_ctrl$counts
txi_wt_fe_wt_ctrl$length[txi_wt_fe_wt_ctrl$length == 0] <- 1 # longueurs nulles definies a 1
g_t=as.factor(paste(genot,treat,sep="_"))
colData=data.frame(g_t,genot,treat,row.names=colnames(countData))
dds <- DESeq(DESeqDataSetFromTximport(txi_wt_fe_wt_ctrl, colData, ~treat))
res_wt_fe_wt_ctrl=results(dds, lfcThreshold = 1, alpha = 0.05)

# s45_fe vs s45_ctrl

txi_s45_fe_s45_ctrl<-tximport(c("sample9.genes.results","sample10.genes.results",
                                "sample11.genes.results","sample12.genes.results",
                                "sample13.genes.results","sample14.genes.results",
                                "sample15.genes.results","sample16.genes.results"),
                              type="rsem", txIn = FALSE, txOut = FALSE)

genot=factor(rep(c("SR45"),each=8))
treat=factor(c(rep(c("Fe"),each=4),rep(c("Ctrl"),each=4)))

countData<-txi_s45_fe_s45_ctrl$counts
txi_s45_fe_s45_ctrl$length[txi_s45_fe_s45_ctrl$length == 0] <- 1
g_t=as.factor(paste(genot,treat,sep="_"))
colData=data.frame(g_t,genot,treat,row.names=colnames(countData))
dds <- DESeq(DESeqDataSetFromTximport(txi_s45_fe_s45_ctrl, colData, ~treat))
res_s45_fe_s45_ctrl=results(dds, lfcThreshold = 1, alpha = 0.05)

# wt_fe vs s45_fe

txi_wt_fe_s45_fe<-tximport(c("sample1.genes.results","sample2.genes.results",
                             "sample3.genes.results","sample4.genes.results",
                             "sample9.genes.results","sample10.genes.results",
                             "sample11.genes.results","sample12.genes.results"),
                           type="rsem", txIn = FALSE, txOut = FALSE)

genot=factor(c(rep(c("WT"),each=4),rep(c("S45"),each=4)))
treat=factor(c(rep(c("Fe"),each=8)))

countData<-txi_wt_fe_s45_fe$counts
txi_wt_fe_s45_fe$length[txi_wt_fe_s45_fe$length == 0] <- 1
g_t=as.factor(paste(genot,treat,sep="_"))
colData=data.frame(g_t,genot,treat,row.names=colnames(countData))
dds <- DESeq(DESeqDataSetFromTximport(txi_wt_fe_s45_fe, colData, ~genot))
res_wt_fe_s45_fe=results(dds, lfcThreshold = 1, alpha = 0.05)
```

```
71
72 # wt_ctrl vs s45_ctrl
73
74 txi_wt_ctrl_s45_ctrl<-tximport(c("sample5.genes.results","sample6.genes.results",
75                                  "sample7.genes.results","sample8.genes.results",
76                                  "sample13.genes.results","sample14.genes.results",
77                                  "sample15.genes.results","sample16.genes.results"),
78                                type="rsem", txIn = FALSE, txOut = FALSE)
79
80 genot=factor(c(rep(c("WT"),each=4),rep(c("S45"),each=4)))
81 treat=factor(c(rep(c("Ctrl"),each=8)))
82
83 countData<-txi_wt_ctrl_s45_ctrl$counts
84 txi_wt_ctrl_s45_ctrl$length[txi_wt_ctrl_s45_ctrl$length == 0] <- 1
85 g_t=as.factor(paste(genot,treat,sep="_"))
86 colData=data.frame(g_t,genot,treat,row.names=colnames(countData))
87 dds <- DESeq(DESeqDataSetFromTximport(txi_wt_ctrl_s45_ctrl, colData, ~genot))
88 res_wt_ctrl_s45_ctrl=results(dds, lfcThreshold = 1, alpha = 0.05)
89
90 # DETs :
91
92 # wt_fe vs wt_ctrl
93
94 txi_wt_fe_wt_ctrl<-tximport(c("sample1.isoforms.results","sample2.isoforms.results",
95                               "sample3.isoforms.results","sample4.isoforms.results",
96                               "sample5.isoforms.results","sample6.isoforms.results",
97                               "sample7.isoforms.results","sample8.isoforms.results"),
98                             type="rsem", txIn = TRUE, txOut = TRUE)
99
100 genot=factor(rep(c("WT"),each=8))
101 treat=factor(c(rep(c("Fe"),each=4),rep(c("Ctrl"),each=4)))
102
103 countData<-txi_wt_fe_wt_ctrl$counts
104 txi_wt_fe_wt_ctrl$length[txi_wt_fe_wt_ctrl$length == 0] <- 1
105 g_t=as.factor(paste(genot,treat,sep="_"))
106 colData=data.frame(g_t,genot,treat,row.names=colnames(countData))
107 dds <- DESeq(DESeqDataSetFromTximport(txi_wt_fe_wt_ctrl, colData, ~treat))
108 rest_wt_fe_wt_ctrl=results(dds, lfcThreshold = 1, alpha = 0.05)
109
110 # s45_fe vs s45_ctrl
111
112 txi_s45_fe_s45_ctrl<-tximport(c("sample9.isoforms.results","sample10.isoforms.results",
113                                 "sample11.isoforms.results","sample12.isoforms.results",
114                                 "sample13.isoforms.results","sample14.isoforms.results",
115                                 "sample15.isoforms.results","sample16.isoforms.results"),
116                               type="rsem", txIn = TRUE, txOut = TRUE)
117
```

```
genot=factor(rep(c("SR45"),each=8))
treat=factor(c(rep(c("Fe"),each=4),rep(c("Ctrl"),each=4)))

countData<-txi_s45_fe_s45_ctrl$counts
txi_s45_fe_s45_ctrl$length[txi_s45_fe_s45_ctrl$length == 0] <- 1
g_t=as.factor(paste(genot,treat,sep="_"))
colData=data.frame(g_t,genot,treat,row.names=colnames(countData))
dds <- DESeq(DESeqDataSetFromTximport(txi_s45_fe_s45_ctrl, colData, ~treat))
rest_s45_fe_s45_ctrl=results(dds, lfcThreshold = 1, alpha = 0.05)

# wt_fe vs s45_fe

txi_wt_fe_s45_fe<-tximport(c("sample1.isoforms.results","sample2.isoforms.results",
                             "sample3.isoforms.results","sample4.isoforms.results",
                             "sample9.isoforms.results","sample10.isoforms.results",
                             "sample11.isoforms.results","sample12.isoforms.results"),
                           type="rsem", txIn = TRUE, txOut = TRUE)

genot=factor(c(rep(c("WT"),each=4),rep(c("S45"),each=4)))
treat=factor(c(rep(c("Fe"),each=8)))

countData<-txi_wt_fe_s45_fe$counts
txi_wt_fe_s45_fe$length[txi_wt_fe_s45_fe$length == 0] <- 1
g_t=as.factor(paste(genot,treat,sep="_"))
colData=data.frame(g_t,genot,treat,row.names=colnames(countData))
dds <- DESeq(DESeqDataSetFromTximport(txi_wt_fe_s45_fe, colData, ~genot))
rest_wt_fe_s45_fe=results(dds, lfcThreshold = 1, alpha = 0.05)

# wt_ctrl vs s45_ctrl

txi_wt_ctrl_s45_ctrl<-tximport(c("sample5.isoforms.results","sample6.isoforms.results",
                                 "sample7.isoforms.results","sample8.isoforms.results",
                                 "sample13.isoforms.results","sample14.isoforms.results",
                                 "sample15.isoforms.results","sample16.isoforms.results"),
                               type="rsem", txIn = TRUE, txOut = TRUE)

genot=factor(c(rep(c("WT"),each=4),rep(c("S45"),each=4)))
treat=factor(c(rep(c("Ctrl"),each=8)))

countData<-txi_wt_ctrl_s45_ctrl$counts
txi_wt_ctrl_s45_ctrl$length[txi_wt_ctrl_s45_ctrl$length == 0] <- 1
g_t=as.factor(paste(genot,treat,sep="_"))
colData=data.frame(g_t,genot,treat,row.names=colnames(countData))
dds <- DESeq(DESeqDataSetFromTximport(txi_wt_ctrl_s45_ctrl, colData, ~genot))
rest_wt_ctrl_s45_ctrl=results(dds, lfcThreshold = 1, alpha = 0.05)


```

```
# SUMMARIES

c(summary(res_wt_fe_wt_ctrl),summary(res_s45_fe_s45_ctrl),
  summary(res_wt_fe_s45_fe),summary(res_wt_ctrl_s45_ctrl))

c(summary(rest_wt_fe_wt_ctrl),summary(rest_s45_fe_s45_ctrl),
  summary(rest_wt_fe_s45_fe),summary(rest_wt_ctrl_s45_ctrl))

# DASGs :

minReads<-5

# wt_ctrl_wt_fe

das_wt_ctrl_wt_fe<-maser("../rMATs/rMATsResults/WT_CTRL_vs_WT_FE/", c("WT_CTRL", "WT_FE"),
    ftype = "JCEC")

das_wt_ctrl_wt_fe_filtered<-filterByCoverage(das_wt_ctrl_wt_fe, avg_reads = minReads)
# les evenements peu supportes ne sont pas comptabilises
das_wt_ctrl_wt_fe_top <- topEvents(das_wt_ctrl_wt_fe_filtered, fdr = 0.05, deltaPSI = 0.1)
# selection des evenements significatifs

# s45_ctrl_s45_fe

das_s45_ctrl_s45_fe<-maser("../rMATs/rMATsResults/S45_CTRL_vs_S45_FE/", c("SR45_CTRL",
    "SR45_FE"), ftype = "JCEC")
das_s45_ctrl_s45_fe_filtered<-filterByCoverage(das_s45_ctrl_s45_fe, avg_reads = minReads)
das_s45_ctrl_s45_fe_top <- topEvents(das_s45_ctrl_s45_fe_filtered, fdr = 0.05, deltaPSI = 0.1)

# wt_ctrl_s45_ctrl

das_wt_ctrl_s45_ctrl<-maser("../rMATs/rMATsResults/WT_CTRL_vs_S45_CTRL/", c("SR45WT_FE",
    "SR45WT_CTRL"), ftype = "JCEC")
das_wt_ctrl_s45_ctrl_filtered<-filterByCoverage(das_wt_ctrl_s45_ctrl, avg_reads = minReads)
das_wt_ctrl_s45_ctrl_top <- topEvents(das_wt_ctrl_s45_ctrl_filtered, fdr = 0.05, deltaPSI = 0.1)

# wt_fe_s45_fe

das_wt_fe_s45_fe<-maser("../rMATs/rMATsResults/WT_FE_vs_S45_FE/", c("WT_FE", "SR45_CTRL"),
    ftype = "JCEC")
das_wt_fe_s45_fe_filtered<-filterByCoverage(das_wt_fe_s45_fe, avg_reads = minReads)
das_wt_fe_s45_fe_top <- topEvents(das_wt_fe_s45_fe_filtered, fdr = 0.05, deltaPSI = 0.1)


# obtention sous forme de listes des DEGs, DASGs et DETs

getDEX<-function(data){ # get Differentially Expressed Genes/Transcripts
```

```
  up<-rownames(data[which(data$log2FoldChange > 1 & data$padj < 0.05),])
  down<-rownames(data[which(data$log2FoldChange < -1 & data$padj < 0.05),])
  c(up,down)
}

getASG<-function(data){ # get Alternatively Spliced Genes

  events<-c("A3SS","A5SS","SE","RI","MXE")

  getGenesByEvents<-function(events,data){

    res<-try( (summary(data, type=events))$GeneID, silent = TRUE)
    # gestion des erreurs quand le vecteur est vide

    if(class(res) == "try-error"){
      say<-c()
    }else{
      say<-(summary(data, type=events))$GeneID
    }
    say
  }
  res<-unique(unlist(sapply(events,getGenesByEvents,data)))
  res
}

# recuperation des listes :

DEGs_wt_fe_wt_ctrl<-getDEX(res_wt_fe_wt_ctrl)
DEGs_s45_fe_s45_ctrl<-getDEX(res_s45_fe_s45_ctrl)
DEGs_wt_fe_s45_fe<-getDEX(res_wt_fe_s45_fe)
DEGs_wt_ctrl_s45_ctrl<-getDEX(res_wt_ctrl_s45_ctrl)

DETs_wt_fe_wt_ctrl<-getDEX(rest_wt_fe_wt_ctrl)
DETs_s45_fe_s45_ctrl<-getDEX(res_s45_fe_s45_ctrl)
DETs_wt_fe_s45_fe<-getDEX(res_wt_fe_s45_fe)
DETs_wt_ctrl_s45_ctrl<-getDEX(res_wt_ctrl_s45_ctrl)

DASGs_wt_ctrl_wt_fe<-getASG(das_wt_ctrl_wt_fe_top)
DASGs_s45_ctrl_s45_fe<-getASG(das_s45_ctrl_s45_fe_top)
DASGs_wt_ctrl_s45_ctrl<-getASG(das_wt_ctrl_s45_ctrl_top)
DASGs_wt_fe_s45_fe<-getASG(das_wt_fe_s45_fe_top)

# les ASGs sont recuperes comme les DASGs, sans etre soumis a l'etape de
# filtration sur base des criteres de significativites (deltaPSI et FDR)

ASGs_wt_ctrl_wt_fe<-getASG(das_wt_ctrl_wt_fe_filtered)
ASGs_s45_ctrl_s45_fe<-getASG(das_s45_ctrl_s45_fe_filtered)
```

```
ASGs_wt_ctrl_s45_ctrl<-getASG(das_wt_ctrl_s45_ctrl_filtered)
ASGs_wt_fe_s45_fe<-getASG(das_wt_fe_s45_fe_filtered)

# construction des barplots representants les frequences
# des evenements a travers les differents echantillons

plotFreqByEvents<-function(ev,data){

  getGenesByEvents<-function(ev,data){

    res<-try( (summary(data, type=ev))$GeneID, silent = TRUE)
    if(class(res) == "try-error"){
      say<-c()
    }else{
      say<-(summary(data, type=ev))$GeneID
    }
    length(say)
  }
  resEvents<-unlist(lapply(ev,getGenesByEvents,data))
  resEvents
}

events<-c("A3SS","A5SS","SE","RI","MXE")


dPlot<-cbind(plotFreqByEvents(events,das_wt_ctrl_s45_ctrl_top),
             plotFreqByEvents(events,das_wt_fe_s45_fe_top),
             plotFreqByEvents(events,das_wt_ctrl_wt_fe_top),
             plotFreqByEvents(events,das_s45_ctrl_s45_fe_top))

rownames(dPlot)<-events
colnames(dPlot)<-c("wt_Ctrl-s45_Ctrl","wt_Fe-s45_Fe","wt_Ctrl-wt_Fe","s45_Ctrl-s45_Fe")

col<-c("#ffd700", "#f99300", "#302d6a", "#830083", "#9c2727")
events<-c("A3SS","A5SS","SE","RI","MXE")

H1<-barplot(dPlot,col = col, border=NA,
            ylab="Frequences des evenements", xlab="Echantillons")
legend(4,800,legend=rev(events),
       fill=rev(col),bty = "n")

# construction des diagrammes d'Euler

plotVenn<-function(name,deg,dag,asg){
  venn.plot <- venn.diagram(
    x = list(DEGs=deg, DAGs=dag, ASGs=asg),
    euler.d = TRUE,
```

```
    fill=c("#de6e7a", "#8db2f3", "#6fd4e5"), #
    filename = name,
    #category.names = c("","",""),
    category.names = c("DEGs","DASGs","ASGs"),
    col=NA,
    cex = 1.8,
    cat.cex = 2.3,
    cat.pos = 0,
    scaled=TRUE
  );
}

plotVenn("R_scripts/plots/wt_fe_vs_wt_ctrl",DEGs_wt_fe_wt_ctrl,
         DASGs_wt_ctrl_wt_fe,ASGs_wt_ctrl_wt_fe)
plotVenn("R_scripts/plots/s45_fe_s45_ctrl",DEGs_s45_fe_s45_ctrl,
         DASGs_s45_ctrl_s45_fe,ASGs_s45_ctrl_s45_fe)
plotVenn("R_scripts/plots/wt_fe_s45_fe",DEGs_wt_fe_s45_fe,
         DASGs_wt_fe_s45_fe,ASGs_wt_fe_s45_fe)
plotVenn("R_scripts/plots/wt_ctrl_s45_ctrl",DEGs_wt_ctrl_s45_ctrl,
         DASGs_wt_ctrl_s45_ctrl,ASGs_wt_ctrl_s45_ctrl)
```