Rapport de stage

Master 2 : Bioinformatique et Biologie des Systèmes

Comparaison d'outils pour l'analyse de modifications d'ARN non codant à partir de données nanopore "ARN direct"

Annabelle BRU

Encadrantes : Christine GASPIN et Nathalie VIALANEIX

2023

I. Introduction..... 5

1. Présentation du laboratoire..... 5

a - INRAE..... 5

b - MIAT..... 5

2. Contexte scientifique du stage : ARN et modifications..... 6

II. Matériel et Méthodes..... 10

1. Les données..... 10

2. Logiciels..... 11

3. Étapes de détection de variants..... 12

4. Penguin..... 14

5. Traçabilité des analyses..... 18

III. Résultats..... 18

1. Détection de variants..... 18

a. Visualisation IGV..... 18

b. Comparaison avec la base de données..... 20

c. Analyse de la contribution de la profondeur sur la qualité des résultats..... 23

d. Application des filtres..... 27

e. Résultats avec graphmap..... 28

2. Détection de pseudo-uridylations par apprentissage avec Penguin..... 30

a. Description globale des événements alignés pour les données Hek293..... 30

b. Prédictions de Penguin..... 32

c. Prédictions après modification du script..... 36

IV. Conclusion..... 37

a. Perspective..... 37

b. Conclusion personnelle..... 38

V. Bibliographie..... 38

VI. Annexes..... 40

Abstract

La pseudouridylation, processus d'isomérisation d'une Uridine en Pseudouridine, est une des modifications de l'ARN les plus fréquentes. Cette modification a été détectée auparavant par des techniques encore limitées. Le séquençage nanopore ARN direct est une méthode prometteuse, permettant de séquencer de l'ARN natif sans étape préalable d'amplification ou de reverse transcription. En plus de la suppression des biais induits par ces deux étapes, le séquençage nanopore a permis de déterminer que les Pseudouridines sont détectables en tant que C au niveau du pore au lieu d'un U. Les deux approches mises en place à partir de cette observation se basent sur des caractéristiques complémentaires. La première approche se concentre sur l'erreur d'attribution de base lors de l'appel de base, par une méthode de détection de variants. La deuxième, se base sur les caractéristiques du signal électrique de 5-mers pour les implémenter dans des méthodes d'apprentissage automatiques implémentées par le logiciel Penguin afin de prédire les Pseudouridines. L'objectif est de comparer ces différentes approches sur l'ARNr de l'humain ainsi que sur celui d'*Arabidopsis thaliana* avec des jeux de données nanopore ARN direct provenant du porc, de l'humain et *Arabidopsis thaliana*.

Les résultats de la première approche montrent d'une part que les données porcines, malgré une ribodéplétion, s'alignent sur l'ARNr humain, avec des résultats similaires aux données humaines, ce qui confortent l'utilisation des modifications connues chez l'humain comme modèle chez le porc. D'autre part, la profondeur de séquençage a un impact entre les données humaines et porcines ainsi qu'au sein d'*Arabidopsis thaliana* mais ne permet pas d'expliquer la répartition des différentes classes de prédiction.

Quant aux résultats de Penguin, les prédictions ne se révèlent pas fiables dû aux incohérences trouvées au niveau de la construction des jeux d'apprentissage et de test.

Finalement ce travail a pû mettre en évidence, dans le cadre des données utilisées, que la première approche de détection de variant était plus efficace, avec de potentielles nouvelles positions de pseudouridines à vérifier, ainsi que la présence d'une erreur potentielle dans l'outil publié.

Pseudouridylation, the process of isomerizing a Uridine into a Pseudouridine, is one of the most frequent RNA modifications. This modification has previously been detected by techniques, however they are still limited. Direct nanopore RNA sequencing is a promising method, enabling native RNA to be sequenced without prior amplification or reverse transcription. In addition to eliminating the biases induced by these two steps, nanopore sequencing has determined that Pseudouridines are detectable as a C at the pore level instead of a U. The two approaches developed from this observation are based on complementary features. The first approach focuses on the base assignment error during base calling, using a variant detection method. The second, based on the characteristics of the 5-mers electrical signal, implements them in machine learning methods implemented by the Penguin software to predict Pseudouridines. The aim is to compare these different approaches on human and *Arabidopsis thaliana* rRNA with direct nanopore RNA datasets from pig, human and *Arabidopsis thaliana*.

The results of the first approach show that, despite ribosomal RNA depletion, porcine data align with human rRNA, with results similar to human data, supporting the use of known human modifications as a porcine model. On the other hand, sequencing depth has an impact between human and porcine data, as well as within *Arabidopsis thaliana*, but does not explain the distribution of the different prediction classes.

As for the Penguin results, the predictions proved unreliable due to inconsistencies found in the construction of the training and test datasets.

Finally, this work demonstrated, within the framework of the data used, that the first variant detection approach was more effective, with potential new pseudouridine positions to be verified, as well as the presence of a potential error in the published tool.

I. Introduction

1. Présentation du laboratoire

a - INRAE

L'institut national de recherche pour l'agriculture, l'alimentation et l'environnement (INRAE) est un organisme public de recherche scientifique placé sous la double tutelle du ministère de l'Enseignement Supérieur et de la Recherche, et du ministère de l'Alimentation, de l'Agriculture et de la Pêche. INRAE est composé de 14 départements scientifiques répartis sur 18 centres de recherche régionaux.
Les principales missions d'INRAE sont de :

- produire et diffuser des connaissances scientifiques ;
- concevoir des innovations et des savoir-faire pour la société ;
- fournir une expertise aux institutions publiques et aux acteurs du privé ;
- débattre des progrès et avancées scientifiques ;
- former à la recherche.

À des fins de développement durable, INRAE concentre ses recherches sur les thématiques concernant le changement climatique, l'agroécologie, l'alimentation ainsi que l'environnement et la biodiversité.

b - MIAT

L'Unité de Mathématiques et Informatique Appliquées de Toulouse (MIAT) est une unité du département Mathématiques et Numérique (MathNum) d'INRAE dont l'objectif est de développer et mettre en œuvre des méthodes mathématiques et informatiques afin de promouvoir une recherche pluridisciplinaire au sein d'INRAE et de favoriser le développement de méthodes mathématiques et informatiques qui permettent de mieux répondre aux problématiques scientifiques des autres départements de l'institut. L'unité est composée de deux équipes de recherche :

- SciDyn (Simulation, Contrôle et Inférence de DYNamiques agroécologiques et biologiques) : met en œuvre des méthodes en statistique, informatique et intelligence artificielle pour la modélisation, la simulation et le pilotage des dynamiques des systèmes agro-écologiques, biologiques et forestiers ;
- SaAB (Statistiques et Algorithmique pour la Biologie) : développe et met à disposition des biologistes des méthodes mathématiques, statistiques et informatiques permettant de contribuer à la compréhension du vivant au niveau moléculaire et cellulaire.

L'unité dispose aussi de trois équipes plateforme :

- Genotoul-Bioinfo (Plateforme bioinformatique du GIS GENOTOUL - Génopole Toulouse Midi-Pyrénées) ;
- RECORD (Plateforme de modélisation et de simulation des agro-écosystèmes) ;
- SIGENAE (Plateforme Systèmes d'information des génomes des animaux d'élevage).

Le stage au sein de l'unité MIAT a été encadré par Mme Christine GASPIN et Mme Nathalie VIALANEIX, toutes les deux Directrices de Recherche au sein de l'unité MIAT, à cheval entre les équipes SaAB et Genotoul-Bioinfo. Mon affiliation au sein de l'unité était la plateforme Genotoul-Bioinfo.

2. Contexte scientifique du stage : ARN et modifications

L'acide ribonucléique (ARN) provient de la transcription de l'acide désoxyribonucléique (ADN), porteur de l'information génétique. L'ARN est généralement sous forme monocaténaire. Ce dernier est composé de 4 nucléotides : Adénine (A), Uracile (U), Cytosine (C) et Guanine (G). L'ARN est sujet à plus de 150 modifications post-transcriptionnelles qui sont majoritairement des méthylations. Ces modifications ont un impact dans divers processus biologiques (efficacité de la traduction, stabilité, dégradation etc…). Dans le cadre du stage que j'ai effectué, on s'intéresse plus particulièrement à une modification de l'ARN, la pseudouridylation.

La Pseudouridine est un ribonucléotide dérivé de l'Uridine et se trouve dans certains ARN non codants (ARNnc) ainsi que dans des ARN messagers. Comme montré dans la figure 1, elle résulte de modifications post-transcriptionnelles de résidus d'Uridine ayant subi

une rotation à 180° des atomes N3 et C6 du cycle pyrimidique. Cela ne modifie pas les propriétés d'appariement Watson-Crick par rapport à l'Uridine mais cela modifie la nature de la liaison glycosidique entre le ribose et la base. Cependant, la Pseudouridine possède 2 groupes NH qui permettent de former une nouvelle liaison hydrogène. Cette capacité est utilisée pour stabiliser la structure de certains ARNs.

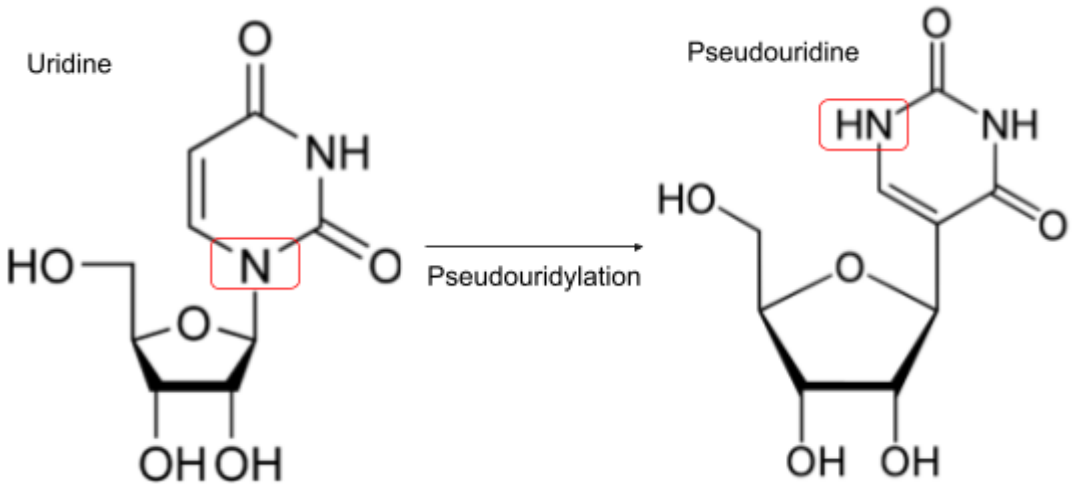


Figure 1 : Schéma de l'Uridine (à gauche) et de la Pseudouridine (à droite).

L'une des boucles des ARNt porte un résidu de Pseudouridine conservé impliqué dans la stabilisation de la structure tridimensionnelle. Pour cette raison, cette boucle est appelée boucle « TψC », d'après la séquence conservée centrée sur la Pseudouridine. La pseudouridylation peut être catalysée par la Pseudouridine synthase (Pus) présente dans les 3 domaines du vivant, qui modifie les ARNt ainsi que quelques ARNnc. Elle peut aussi être catalysée par l'intermédiaire des ARN nucléolaires (snoRNA) qui sont des ARN présents dans le nucléole des cellules eucaryotes, et qui aident à la maturation des ARNr. Plus particulièrement, un type de snoRNA est responsable de la pseudouridylation : il s'agit du type H/ACA (Ganot *et al*, 1997) retrouvé chez les eucaryotes et les archées. Les snoRNA, comme illustré dans la figure 2, prennent la plupart du temps une forme de double épingle à cheveux (certains peuvent en avoir une seule), avec une zone non appariée (boucle interne) dans chaque épingle. Cette zone de non-appariement, nommée poche de pseudouridylation, se lie par complémentarité à un ARNr cible, en laissant 2 nucléotides non appariés, dont une uridine qui sera isomérisée en Pseudouridine. La pseudouridylation est réalisée par une protéine associée à l'ARN, la dyskérine.

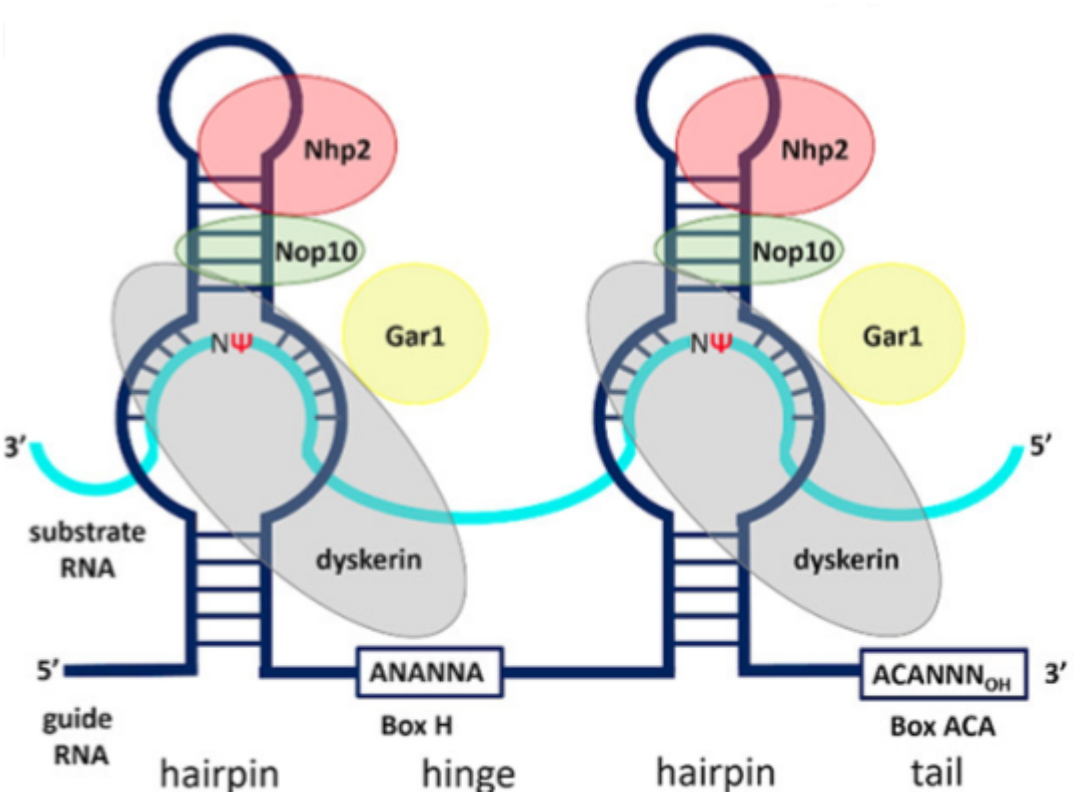


Figure 2 : Schéma de la machinerie de la pseudouridylation (Kiss *et al*, 2022). En gris, la dyskerine responsable de la pseudouridylation sur l'ARN cible accompagnée par trois autres protéines formant le complexe Nop10 en vert, Nhp2 en rouge et Gar1 en jaune. Le snoRNA est l'ARN guide en forme de double épingle à cheveux.

À ce jour de nombreuses techniques ont été mises en place afin de détecter et prédire ces modifications. Ces techniques, telles que la "Pseudouridine site identification sequencing" ou PSI-seq (Lovejoy *et al*, 2014) ou l'immunoprécipitation d'anticorps couplée au séquençage de nouvelle génération (NGS), restent encore limitées. En effet, un des problèmes de ces approches NGS c'est qu'elles ne sont souvent pas quantitatives, ne permettent pas d'apporter des informations sur les isoformes spécifiques et l'étape d'amplification ajoute encore des biais dans les données séquencées. Pour pallier ces problèmes, Oxford Nanopore Technologies (ONT) a mis en place une alternative qui consiste à faire du séquençage nanopore sur de l'ARN direct sans étape d'amplification et de reverse transcription. L'avantage de cette approche est qu'elle est quantitative, permet de ne pas avoir de biais de l'amplification par PCR et se situe à la résolution d'une seule molécule. Le principe décrit dans la figure 3, est identique avec de l'ADN ou de l'ARN. La séquence d'ARN passe dans un **pore** implémenté dans une membrane reliée à une puce qui mesure le **courant électrique** à travers ce nanopore. Le principe est lorsqu'une molécule passe dans ce pore, le **courant est perturbé** ce qui produit un "*squiggle*" qui sera ensuite **décodé** par un algorithme d'appel de

base implémenté par un réseau de neurones (NN). Ce dernier permet de passer du signal électrique à la séquence en nucléotides.

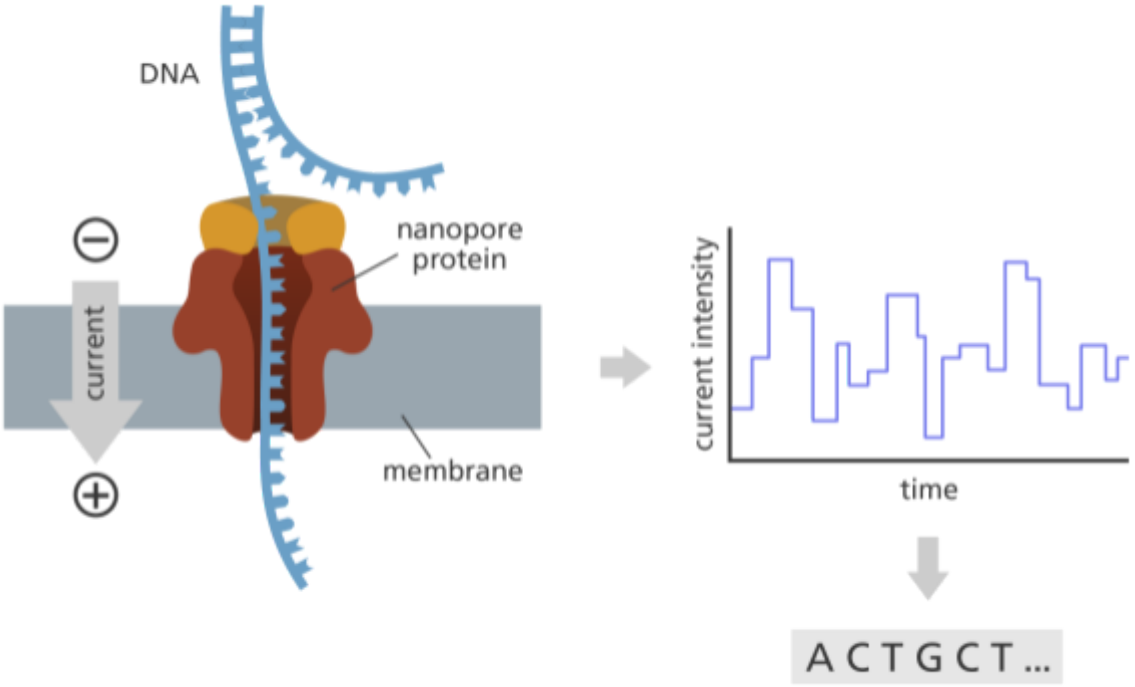


Figure 3 : Schéma du séquençage nanopore. (What is Oxford Nanopore Technology (ONT) sequencing?)
A gauche le nanopore implanté dans une membrane où un seul brin de l'ADN passe dans le pore ce qui résulte d'un changement d'intensité de courant (à droite) qui peut être traduit en nucléotide.

Les autres avantages du séquençage nanopore est qu'il permet de séquencer directement l'ARN natif sans avoir à le convertir en ADNc. Cette approche a aussi permis de mettre en évidence que la Pseudouridine était détectable au niveau du pore. En effet, lors de l'appel de base effectué par l'algorithme Guppy intégré dans leur suite MinKnow, il a été observé qu'une Pseudouridine est détectée en tant que C et non en tant que U. Partant de cette observation, l'objectif de ce stage a été de comparer deux approches bioinformatiques pour la détection de Pseudouridines à partir de données de séquençage d'ARN direct par nanopore :

- une se basant sur l'erreur d'attribution de base lors de l'étape d'appel de base (donc sur une recherche de variants à partir des données nanopores) ;
- l'autre se concentrant sur l'analyse directe du signal électrique à travers un logiciel implémentant des modèles d'apprentissage automatique.

Pour cette dernière approche, nous avons fait appel à un logiciel nommé Penguin (Hassan *et al*, 2022) qui incorpore trois modèles d'apprentissage automatique dont l'objectif est de prédire et d'identifier les sites des Pseudouridines à partir des caractéristiques extraites du signal électrique brut des reads du séquençage nanopore (k-mer, moyenne, écart-type et taille du signal) et par conséquent de déterminer si le signal émis est perturbé par une modification liée à la Pseudouridine.

II. Matériel et Méthodes

1. Les données

Les données qui ont été mises à ma disposition pour le stage sont des données de séquençage nanopore sur l'ARN direct (au format fastq) provenant de trois espèces différentes :

- deux jeux de données de séquençage nanopore d'échantillons de porc, la particularité étant qu'elles n'ont pas été produites pour de la recherche de modifications et ont fait l'objet d'une ribodéplétion des ARNr. Malgré cela, de nombreux reads s'alignent encore sur l'ARNr humain (similaire à l'ARNr du porc et utilisé dans cette étude pour l'identification des positions modifiées communes à l'homme et au porc) ;
- trois jeux de données de séquençage nanopore d'échantillons des lignées cellulaires rénales humaines Hek293 et 2 jeux de données de séquençage nanopore d'échantillons de cellules cancéreuses Hela, qui sont les données utilisées par l'article présentant Penguin ;
- un jeu de données de séquençage nanopore d'un échantillon d'*Arabidopsis thaliana* obtenues au cours du stage qui ont été produites par la plateforme de séquençage GeT-PlaGe pour évaluer l'apport de la technologie "Nanopore ARN direct" à l'identification de modifications dans les cadres du projet ANR MetRibo et du projet régional SeqOccIn.

Les séquences des sous-unités 5.8S, 18S et 28S de l'ARNr humain ainsi que celles des sous-unités 5.8S, 18S et 25S de l'ARNr d'*Arabidopsis thaliana* m'ont également été fournies au format fasta.

Les pseudouridylations sont bien connues chez certaines espèces modèles notamment chez l'humain et *Arabidopsis thaliana*. On recense des bases de données dédiées à répertorier les modifications de l'ARNr pour ces espèces entre autres snoRNABASE LBME ((Lestrade & Weber, 2006), https://www-snorna.biotoul.fr/index.php) et Plant snoRNA database ((Brown *et al*, 2003), http://www.scri.sari.ac.uk/plant_snoRNA/) respectivement. A noter que snoRNABASE LBME n'est plus accessible depuis quelques semaines. En ce qui concerne la base de données pour *Arabidopsis thaliana*, elle a été alimentée par enrichissement de petits ARN (Chen & Wu, 2009) ainsi que par Pseudouridine-sequencing (Pseudo-seq) (Sun *et al*, 2019) ce qui a permis de confirmer les positions présentes dans la base de données ainsi que d'en inférer de nouvelles. Ces informations permettent de constituer les références nécessaires afin d'évaluer les approches abordées. Ainsi la référence pour l'humain et *Arabidopsis thaliana* compte chacune 99 positions connues de pseudouridylation sur les 3 sous-unités.

Enfin les fast5 associées aux données de Hek293 ont été récupérées au NCBI par le numéro Sequence Read Archive (SRA : SRP298206) et les fast5 d'*Arabidopsis thaliana* ont été fournis par la plateforme de séquençage GeT-PlaGe.

2. Logiciels

Pour travailler sur les données, j'ai demandé l'ouverture d'un compte utilisateur sur le cluster de la plateforme Genotoul-Bioinfo. Les analyses ont été faites majoritairement sur ce cluster de calcul. En ce qui concerne les logiciels, j'ai utilisé samtools-1.9 (http://www.htslib.org/), bcftools-1.9, minimap2-2.5 (https://github.com/lh3/minimap2), graphmap-v0.5.2 pour la partie sur l'alignement et la détection de variants puis IGV-2.15.4 (https://software.broadinstitute.org/software/igv/home) pour la visualisation. Concernant les analyses j'ai utilisé la version 4.4.2 de R ainsi que les librairies tidyverse-1.3.2, ggplot2-3.4.0, ggvenn-0.1.9, plotly-4.10.1. Enfin, pour la partie avec Penguin, j'ai utilisé Python-3.7.4, Nanopolish-0.14.0 (https://github.com/jts/nanopolish), numpy-1.18.1, pandas-1.0.1, sklearn-0.22.2.post1, tensorflow-2.0.0, keras-2.3.1 et jvarkit tools (https://github.com/lindenb/jvarkit.git).

3. Étapes de détection de variants

L'objectif est de créer un fichier Variant Call Format (VCF) qui permet de stocker les variations des séquences génétiques. Pour cela, la méthode utilisée, mpileup (Li, 2011), repose sur l'utilisation de modèles probabilistes basés sur des hypothèses de distribution d'allèles et sur la probabilité d'observer des variations dans les séquences d'ADN pour identifier les SNP (single nucleotide polymorphism) et calculer les fréquences alléliques. Les variants prédits contenus dans le VCF sont ensuite comparés à une référence. Dans le cadre de notre analyse, des filtres sont appliqués par la suite afin de sélectionner les variants d'intérêt (mutation de U en C). Les différentes étapes sont représentées dans la figure 4.

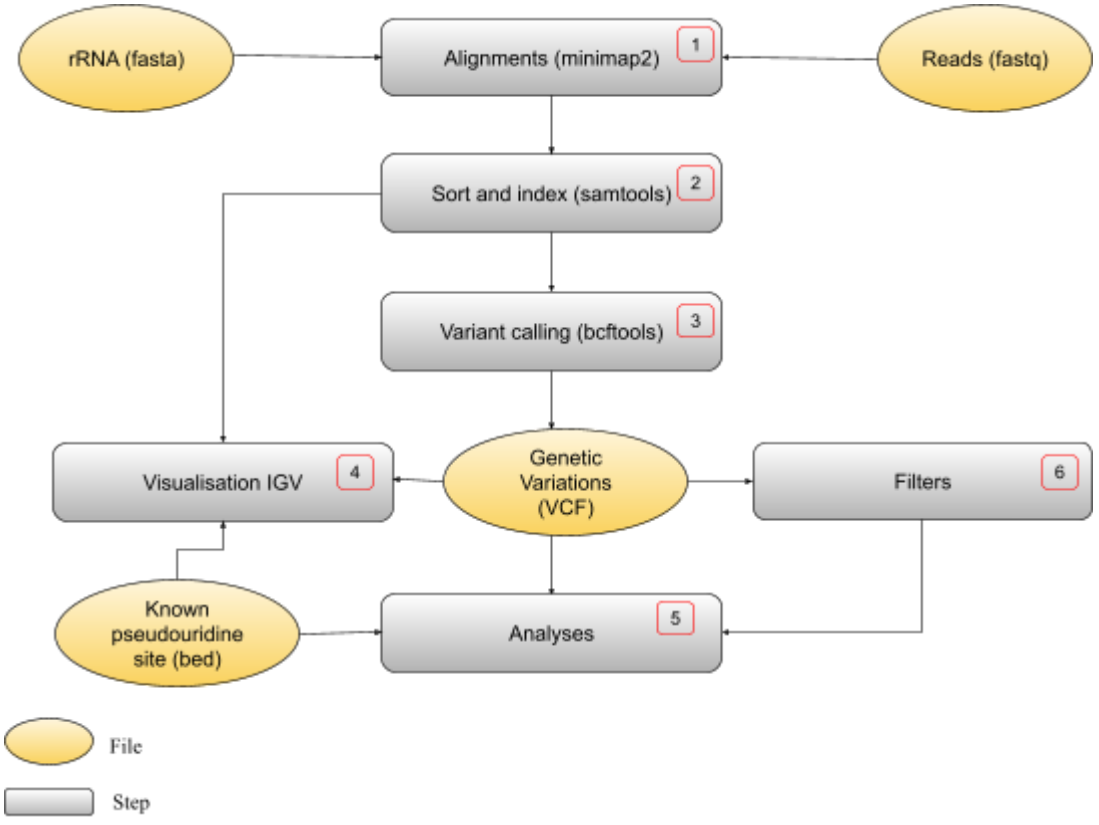


Figure 4 : Pipeline du déroulement de la détection de SNP. Les fichiers sont représentés par les cercles jaunes et les étapes par des rectangles gris.

La première étape (1) importante était d'aligner les données nanopore (les reads) sur les ARNr (la référence à notre disposition). Plusieurs alignements ont été effectués dont :

- l'alignement des données porcines sur l'ARNr humain ;
- l'alignement des données humaines Hek293 sur l'ARNr humain ;

- l’alignement des données *Arabidopsis thaliana* (*A. thaliana*) sur l’ARNr d’*A. thaliana*.

Pour ces alignements, un outil d’alignement adapté aux longs reads de séquençage nanopore, minimap2, a été choisi. Dans un second temps, un autre logiciel adapté pour les longs reads, graph map, a été testé dans le but de comparer d’éventuelles différences. Le principe de ces outils d'alignement est de partitionner l'ensemble des reads en sous-chaînes de longueur k (k-mers) puis de créer une table de hachage où la similarité entre les k-mers va être utilisée afin de détecter les chevauchements entre les reads. Contrairement à graphmap qui utilise tout l'ensemble de k-mers pour construire la table de hachage, minimap2 sélectionne un ensemble minimum représentatif de k-mers (les minimiseurs) ce qui améliore grandement les besoins de stockage et la durée de recherche des chevauchements. Par la suite, ces fichiers d’alignements ont été triés et indexés (2) pour les étapes de détection de variants (3) ainsi que pour la visualisation dans IGV (4). La détection de variants consiste, dans ce cas, à détecter les SNPs dans le but d’observer des mutations U en C caractéristiques de la pseudouridylation. Dans cette optique là, la suite d’outils bcftools est utilisée afin de générer un fichier VCF contenant les informations des variants présents. Cette prédiction s'effectue en utilisant tous les reads s'alignant à une position donnée des ARNr de référence et par conséquent la prédiction est liée à la position des ARNr et non aux reads. En d'autre termes la prédiction est restrictive dans le cas où pour une position donnée, certains reads ont une Pseudouridine et d'autres n'en ont pas. Ensuite, pour comparer les positions prédites avec celles qui sont déjà connues (5), un fichier comprenant ces positions a été conçu (Known_pseudouridine_site) à l’aide des bases de données que nous avons constituées à partir des ressources disponibles chez l’humain et chez *A. thaliana*.
Pour la construction des filtres (6), dans une première partie un filtre strict a été testé. Dans ce cas là, dans le fichier VCF de la détection de variants, une colonne “ALT” désigne les bases alternatives à la base de référence (qui se trouve dans la colonne “REF”). Dans cette colonne “ALT”, on peut trouver plusieurs bases (C, A par exemple). Le but de ce filtre strict est de ne garder que les C qui ne sont pas accompagnés d'une autre base. Dans une seconde partie, d'autres filtres ont été implémentés, se basant plus finement sur la composition en bases à chaque position, obtenus à partir des fichiers d'alignements BAM. Enfin, un dernier filtre sur la qualité (colonne “QUAL” du fichier vcf) a également été testé.

4. Penguin

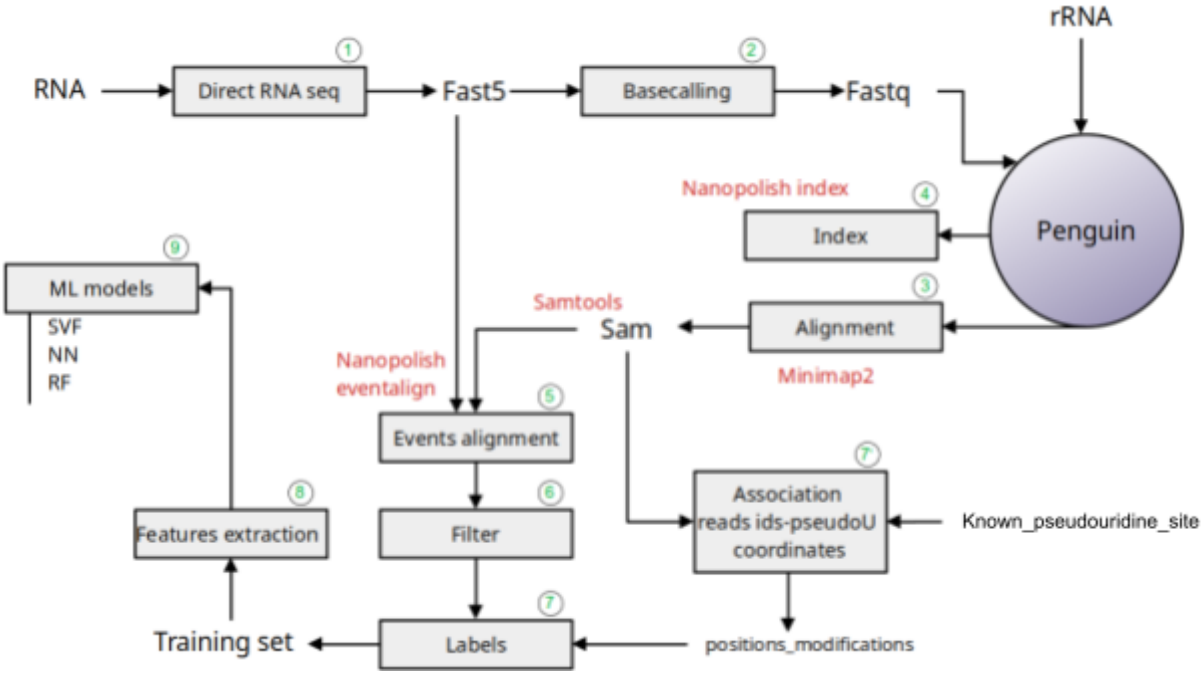


Figure 5 : Représentation du workflow utilisant le logiciel Penguin. Les outils utilisés sont en rouge et la numérotation des différentes étapes en vert.

Dans cette deuxième partie, le logiciel Penguin (https://github.com/Janga-Lab/Penguin) a été utilisé pour la détection des Pseudouridines. Penguin est constitué d'une suite de scripts python qui automatise les différentes étapes du pipeline présenté dans la figure 5. Il implémente des méthodes de machine learning afin de **détecter les Pseudouridines grâce aux métriques calculées à partir des signaux électriques et du résultat de l'appel de base**. Dans le workflow, les étapes 1 et 2 sont réalisées en amont de l'exécution de Penguin. La première étape (*1*) correspond au séquençage nanopore de l'ARN direct qui fournit les signaux électriques au format fast5 qui vont être, dans une seconde étape (*2*), convertis en séquences nucléiques (les reads au format fastq). Penguin prend en entrée ces reads ainsi que le génome de référence, qui dans notre cas sont les mêmes ARNr utilisés dans la partie de détection de variants. Avec ces données, Penguin effectue un alignement des reads sur les ARNr avec minimap2 (*3*). L'étape suivante (*4*) indexe les fichiers fast5 puis aligne les événements du nanopore (fast5) sur les ARNr avec la suite d'outils Nanopolish (*5*). Cette étape permet d'obtenir un fichier texte « eventalign » où chaque read est décomposé en autant d'événements que de bases dans le read.

contig	position	reference_kmer	read_index	strand	event_index	event_level_mean	event_stdv	event_length	model_kmer	model_mean	model_stdv	standardized_level
X03205.1	11	TCCTG	12	t	1	67.79	1.082	0.00299	TCCTG	71.64	2.35	-1.29
X03205.1	11	TCCTG	12	t	2	72.86	1.568	0.00531	TCCTG	71.64	2.35	0.41
X03205.1	11	TCCTG	12	t	3	71.15	1.752	0.01862	TCCTG	71.64	2.35	-0.16
X03205.1	12	CCTGC	12	t	4	85.27	1.421	0.00664	CCTGC	84.20	2.85	0.38
X03205.1	12	CCTGC	12	t	5	82.63	1.483	0.00332	CCTGC	84.20	2.85	-0.43
X03205.1	13	CTGCC	12	t	6	97.26	2.468	0.00531	CTGCC	90.93	3.13	1.59
X03205.1	14	TGCCA	12	t	7	108.67	3.312	0.00232	TGCCA	101.35	3.29	1.75
X03205.1	14	TGCCA	12	t	8	102.44	2.165	0.00199	TGCCA	101.35	3.29	0.26
X03205.1	14	TGCCA	12	t	9	104.75	4.762	0.00631	TGCCA	101.35	3.29	0.81

Figure 6 : Extrait du fichier “eventalign”. “X03205.1” correspond à la sous-unité 18S.

Les « événements » visibles dans la figure 6, correspondent à une position sur l’ARNr (colonne « position ») à laquelle sont associées les caractéristiques de l’événement (k-mers de 5 nucléotides commençant à cette position, moyenne, écart-type de l’intensité du signal et taille du signal). En d’autres termes, chaque ligne correspond à un événement. Ce fichier sert de base pour construire les jeux de données d'apprentissage et de test après **filtrage** (6) des **k-mers** pour ne garder que ceux ayant un **T** au milieu (les détails de la constitution du jeu d'apprentissage et test sont donnés dans la section suivante Mise en œuvre de Penguin). En parallèle, le fichier « positions_modifications », servant de référence est construit (6'). Ce fichier est obtenu grâce au fichier précédemment construit « Known_pseudouridine_site » (se référer à la section 3. Étapes de détection de variants) et au fichier d'alignements obtenus lors de l'étape utilisant minimap2 (*3*). Le fichier positions_modifications est composé de toutes les positions des sites de Pseudouridines connues dans les sous-unités 5.8S, 18S et 28S (25S pour *Arabidopsis thaliana*) de l’ARNr et ces positions sont associées aux différents reads, c'est-à-dire à une position donnée de l’ARNr est associée la position où le read s'aligne ainsi que l'identifiant du read. C’est ce fichier de référence qui permet d’associer, à chaque position du fichier « eventalign », une « vérité » (Pseudouridine ou non) qui est utilisée pour l’apprentissage et l’évaluation de la méthode (*7*). Le principe est donc que les positions du fichier « eventalign » qui sont présentes dans le fichier de référence sont considérées comme modifiées (échantillons positifs) et le reste est non modifié (échantillons négatifs).
Les caractéristiques sur la nature des k-mers, ainsi que sur la moyenne, l’écart-type et la taille du signal sont alors utilisées en entrée d’un modèle de machine learning pour l’apprentissage (*8* et *9*). Penguin propose 3 méthodes dont un réseau de neurones (NN, (Gurney, 1997)), une forêt aléatoire (RF, (Breiman, 2001)) et un Support Vector Machine (SVM, (Cortes & Vapnik, 1995)). En se référant à leur article, Penguin préconise d’utiliser la méthode SVM incluse dans le script SVM_validate.py qui produit de meilleures performances parmi les trois modèles.

Une prédiction est ainsi effectuée pour chaque événement (chaque position de l'ARNr pour chaque read qui s'y aligne) du jeu de données test. En théorie, pour deux événements alignés à une même position de l'ARNr (donc avec le même k-mer), on pourrait obtenir deux prédictions différentes puisque les signaux électriques des deux événements sont différents. Par contre, le calcul des performances du modèle, c'est-à-dire le taux de vrai positifs (TP), vrai négatifs (TN), faux positifs (FP) et faux négatifs (FN), est fait sur la base de la même référence que l'apprentissage, c'est-à-dire qu'une position est uniformément supposée être une Pseudouridine ou non.

Mise en oeuvre de Penguin

Dans le but d'exécuter Penguin correctement, des étapes préalables sont nécessaires. Dans un premier temps, par souci de volumes de données, il faut échantillonner les reads alignés avec minimap2 sur l'ARNr pour *Arabidopsis thaliana* en amont de l'alignement des événements sur ce même ARNr avec Nanopolish eventalign. Le choix de l'échantillonnage est arbitraire, se reposant uniquement sur le fait d'avoir un bon compromis entre quantité d'information et temps d'exécution du logiciel. Nous avons utilisé seulement 10% des reads s'alignant sur l'ARNr de référence.

La construction des différents fichiers est illustrée figure 7. Nous nous sommes basées sur l'alignement des événements sur le fichier de sortie de minimap2. Ainsi, avec tous les reads des données Hek293, 5 fichiers ont été construits, celui qui contient tous les événements alignés (Hek293_eventalign), 2 fichiers issus de ce dernier correspondant à une partition aléatoire par moitié (Hek293_train et Hek293_test) et également, un fichier qui ne comporte que les événements associés à la sous-unité 18S et un autre associé à la sous-unité 28S. Dans le premier cas (séparation aléatoire), les mêmes positions sont utilisées en apprentissage et test, ce qui donne une prédiction optimiste. Dans le second cas, des positions différentes (correspondant à deux sous-unités) sont utilisées en apprentissage et test.

Le même principe a été utilisé pour les données d'*Arabidopsis thaliana*. Dans ce cas là, les deux fichiers pour le jeu de données d'apprentissage et le jeu de données test ont été obtenus respectivement par les deux sorties des alignments minimap2 contenant 10% des reads alignés (Arabidopsis_train, Arabidopsis_test) avec respectivement 40 061 et 40 008 reads. Et comme précédemment, un fichier (à partir de Arabidopsis_train) pour chaque événement des différentes sous-unités (5.8S, 18S et 25S) a également été constitué. Pour ces données là, nous avons donc aussi obtenu 5 fichiers eventalign :

- Arabidopsis_train ;

- Arabidopsis_test ;
- Arabidopsis_5.8S ;
- Arabidopsis_18S ;
- Arabidopsis_25S.

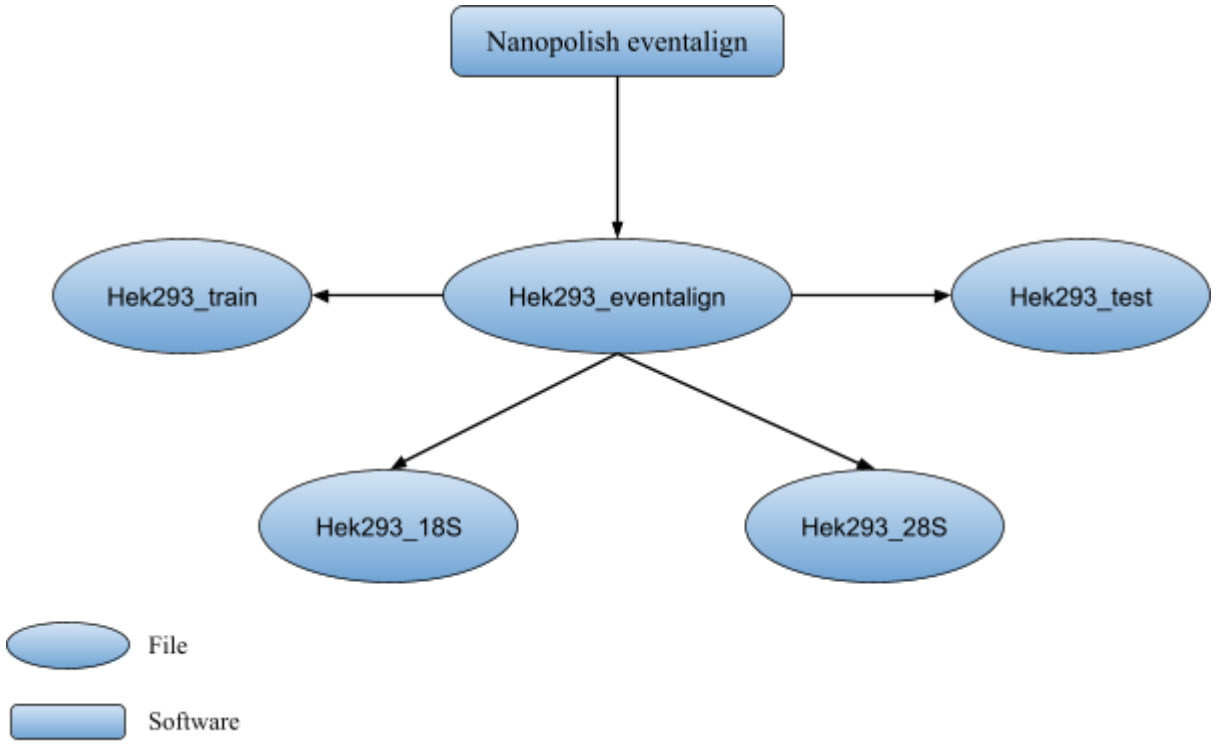


Figure 7 : Schéma d'obtention des différents fichiers eventalign pour les données Hek293.

Une fois ces différents fichiers obtenus, ils sont passés en entrée du script SVM_validate.py comme indiqué dans la figure 8.

```
df1 = pd.read_csv("positions_modifications_Hek293.txt",sep=' ',skiprows=(0),header=(0))
df2 = pd.read_csv("Hek293_eventalign.txt",sep='\t',skiprows=(0),header=(0))
df3 = pd.read_csv("Arabidopsis_test.txt",sep=' ',skiprows=(0),header=(0))
df4 = pd.read_csv("positions_modifications_Arabidopsis.txt",sep='\t',skiprows=(0),header=(0))
```

Figure 8 : Exemple d'entrée du script SVM_validate.py avec les deux fichiers qui vont constituer le jeu d'entraînement pour Hek293 (positions_modifications_Hek293.txt et Hek293_eventalign.txt) ainsi que les deux fichiers qui vont constituer la partie test du modèle sur *Arabidopsis thaliana* (positions_modifications_Arabidopsis.txt et Arabidopsis_test.txt).

Dans un second temps, dans le script SVM_validate.py de Penguin, l'ajout d'une fonction est nécessaire afin de récupérer les positions qui ont été prédites en TP, TN, FP et FN.

5. Traçabilité des analyses

Au cours du stage, les scripts réalisés, les fichiers obtenus et les résultats ont été déposé sur gitlab et plus particulièrement la forgemia de l'inrae (https://forgemia.inra.fr/annabelle.bru/rna-pseudouridylation-detection.git) pour une meilleure organisation et suivie de déroulement des étapes. Les modifications apportées au script du logiciel Penguin sont disponibles sur github (https://github.com/AnnabBru/Penguin, issu d'un « fork » du logiciel initial), ainsi qu'en Annexe.

III. Résultats

1. Détection de variants

a. Visualisation IGV

Dans un premier temps, les résultats ont été mis sur IGV pour donner une première visualisation globale des positions prédites des Pseudouridines ainsi que de la couverture de l'ARNr de référence par les reads et de la composition des alignements en bases.

Humain Hek293

Les alignements sur l'ARNr de référence montrent une faible couverture globale par rapport à ce qui pourrait être attendu. En effet, le taux de reads alignés sur l'ARNr est faible. Sur les 1 278 666 reads du jeu de données de départ, seuls 513 (soit 0,04%) reads s'alignent sur l'ARNr dont 8 sur la sous-unité 5.8S, 111 sur le 18S et 394 sur le 28S. Or, le séquençage de l'ARN direct a permis d'obtenir l'ARN total pour les cellules Hek293. De plus, sachant qu'environ 80% des ARN sont des ARNr, ce qui pourrait être attendu est une couverture globale plus importante, ce qui n'est pas le cas ici et ce qui est aussi constaté par la suite. Par ailleurs, cette couverture est bien inférieure à la couverture des données porcines ayant subi une ribodéplétion.

Figure 9 : Visualisation de l'alignement des données de séquençage humaines Hek293 au niveau de la sous-unité 5.8S sur IGV. En dessous de la séquence, le fichier gff (équivalent au Known_pseudouridine_site) répertoriant les positions des Pseudouridines connues dans la base de données. La composition des bases est montrée selon un code couleur, rouge pour T (U), marron pour G, bleu pour C et vert pour A. Le fichier VCF indique la position où un variant a été détecté.

Dans le cas de la sous-unité 5.8S dans la figure 9, un des jeux de données ne s'aligne pas avec cette sous-unité. La détection de variant a détecté un variant U vers C caractéristique de la Pseudouridine en position 69 qui correspond à ce qui est trouvé dans la base de données. L'autre position 55 présente dans la base de données n'est, cependant, pas prédite par la méthode. Ces résultats peuvent s'expliquer par la faible profondeur de séquençage ou une faible sensibilité de l'approche.

Porc

Concernant les alignements des données porc, ce qui est observable en figure 10, dans un premier temps est une plus grande couverture et profondeur des alignements par rapport à l'humain. Effectivement sur les 14 158 276 reads du jeu de données porcins, 8 233 (0,06%) reads s'alignent sur l'ARNr soit 16 fois de plus que pour les données humaines, dont 5 272 sur le 5.8S, 766 sur le 18S et 2 195 sur le 28S. Malgré la ribodéplétion, un nombre important de reads s'alignent, ce qui montre, d'une part, que la ribodéplétion ne permet pas d'éliminer tous les ARNr et d'autre part, que le 5.8S du porc et de l'humain sont identiques. Là où, pour les cellules Hek293, une seule des deux positions des Pseudouridines a été prédite au niveau du 5.8S, chez le porc les deux le sont, comme attendu chez l'humain. Il semble donc pertinent d'utiliser les modifications connues chez l'humain comme modèle pour le porc.

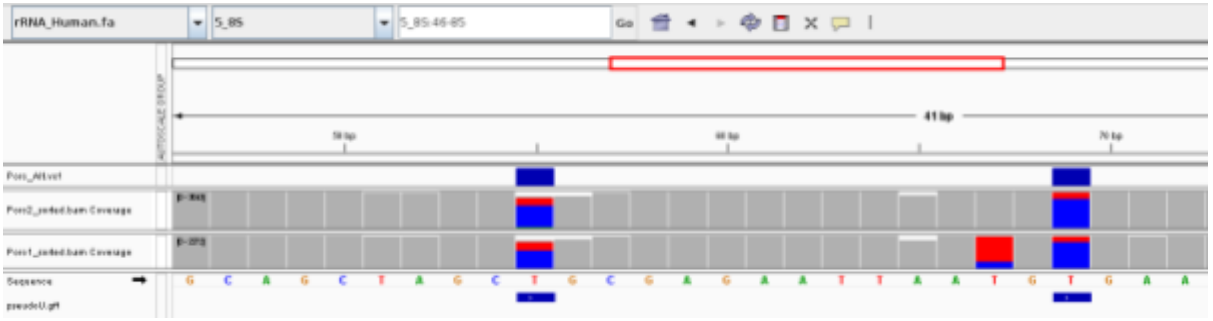


Figure 10 : Visualisation de l'alignement des données de séquençage porc au niveau de la sous-unité 5.8S sur IGV.

b. Comparaison avec la base de données

Hek293 et le porc

Une analyse à l'échelle de l'ensemble des ARNr synthétisée par le diagramme de Venn en figure 11 a permis de mettre en évidence que la majorité des Pseudouridines sont prédites par la méthode. Sur les 99 positions connues de la base de données, 65 (65,66%) sont retrouvées comme des TP pour les cellules Hek293 et 64 (64,65%) le sont pour le porc. Par contre, dans les deux jeux de données, un nombre élevé de FP est obtenu. Parmi ces faux positifs, certaines des positions prédites pourraient correspondre à de nouvelles modifications de type Pseudouridine.

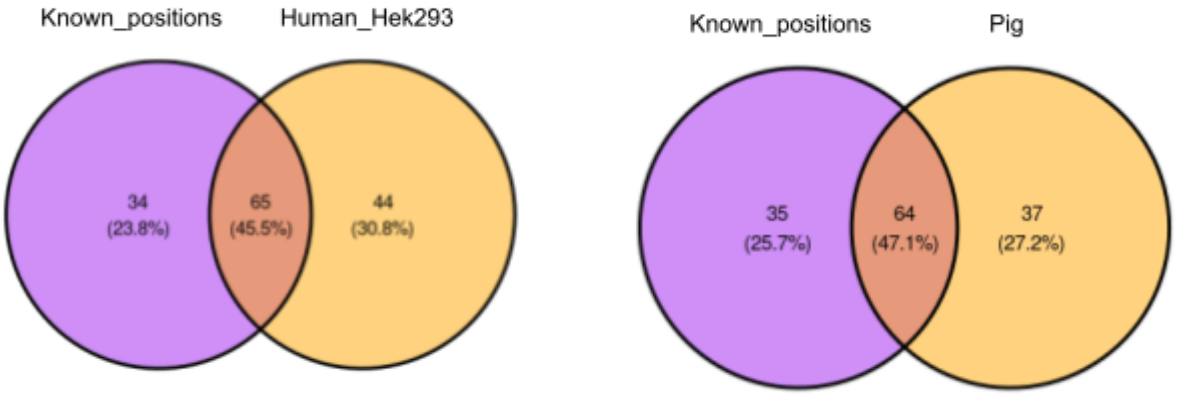


Figure 11 : Comparaison des Pseudouridines connues de la base de données avec les variants prédits par la méthode de détection de variant pour les cellules Hek293 (à gauche) et pour le porc (à droite).

Ils peuvent aussi provenir d'une mauvaise prédiction en lien avec une faible profondeur ou correspondre à d'autres modifications de l'ARN non encore répertoriées dans la base de données.

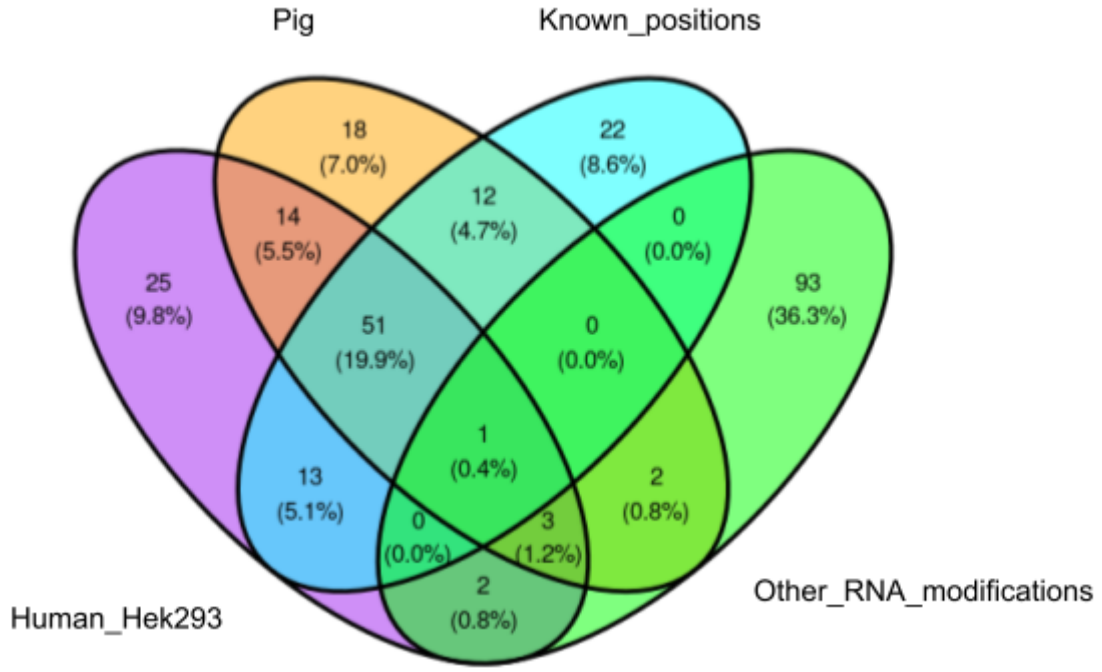


Figure 12 : Comparaison des prédictions des cellules humaines Hek293 (violet) et du porc (orange) avec la base de données (cyan). Les autres modifications de l'ARN (vert) sont celles connues de la base de données.

Les ARNr étant très similaires entre l'humain et le porc, une analyse globale des résultats a été réalisée afin d'observer d'éventuelles conservations entre positions pour les FP prédits. Le diagramme de Venn figure 12 ci-dessus montre qu'il y a 14 prédictions communes entre les cellules Hek293 et le porc non présentes dans la base de données donc il n'est pas improbable que ce soit des Pseudouridines ou encore d'autres modifications de l'ARN. Ce diagramme montre aussi qu'il y a des prédictions (7 au total) qui sont communes avec d'autres modifications de la base de données. Pour les prédictions spécifiques au porc (18) et à Hek293 (25), cela peuvent être des modifications pas encore répertoriées et qui pourrait aussi se traduire par une mutation U vers C. Il est possible aussi que ce soit en réalité des TP spécifiques de chacun des organismes. En ce qui concerne les FN, cela peut s'expliquer par un manque de puissance du signal causé par une faible profondeur mais aussi par une perturbation du signal par la présence d'autres variations/modifications proches sur l'ARNr ou par la sensibilité de la méthode elle-même. Un fait observable sur les données Hek293 appuyant une perturbation du signal, est que sur les 44 FP détectés, 18 (40,91%) sont situés 3 bases ou moins d'un TP prédit.

Arabidopsis thaliana

Concernant l'alignement des données d'*Arabidopsis thaliana*, 401 255 reads ont été alignés sur l'ARNr *d'Arabidopsis thaliana* dont 40 607 sur le 5.8S, 112 734 sur le 18S ainsi que 247 914 sur le 25S.

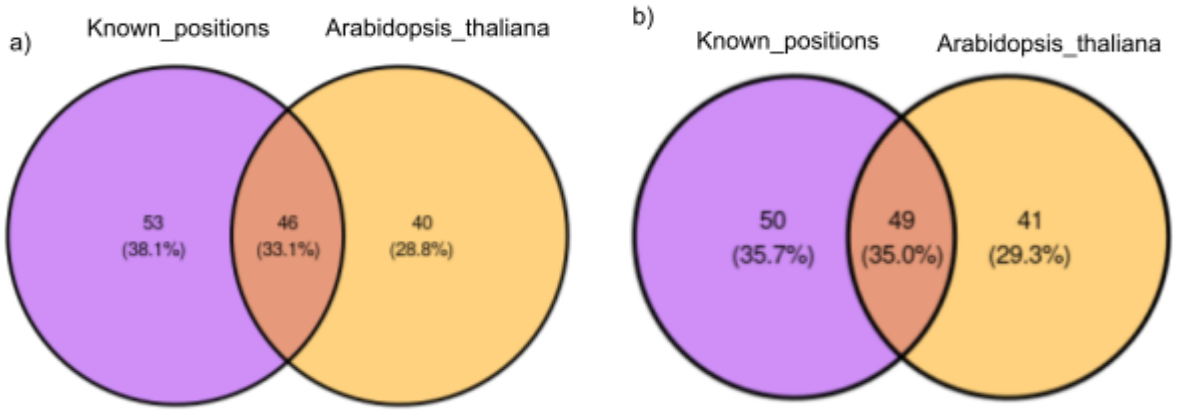


Figure 13 : Comparaison des positions des Pseudouridines connues avec les variants prédits par la méthode de détection de variant pour les données issues du séquençage d'*Arabidopsis thaliana* a) avec max-depth par défaut (8000), b) avec un max-depth à 100 000.

Le diagramme de Venn présenté figure 13 compare les positions connues d'*Arabidopsis thaliana* avec celles qui ont été prédites par l'analyse de variant dans deux cas de figure. Le premier cas en prenant en compte un seuil de profondeur (8000), qui est par défaut dans le logiciel bcftools mpileup (figure 13a). Le second cas (figure 13b) étend ce seuil de profondeur à 100 000. La différence entre ces deux cas de figure reflète qu'avec une profondeur plus importante, la précision de la prédiction s'améliorent très légèrement (53,49% de précision dans le premier cas contre 54,44% dans le second cas).

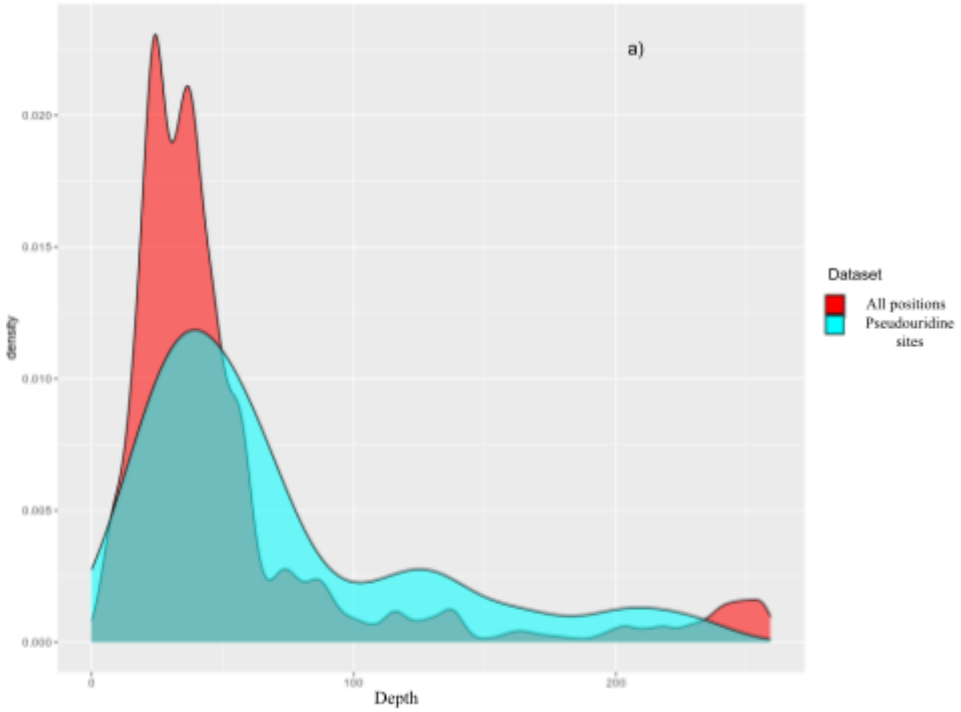
En revanche, malgré une importante couverture pour *Arabidopsis thaliana* par rapport aux données Hek293 et porcines, la précision globale n'est pas meilleure. Afin de compléter ces résultats pour A.thaliana, un alignement des sous-unités 5.8S, 18S et 25S a été effectué ultérieurement entre A.thaliana, l'humain et la levure dont l'objectif est de déterminer des positions de sites de Pseudouridine conservées entre ces espèces. Cela permet de voir si pour une position donnée, elle est modifiée chez la levure et/ou l'humain et par conséquent d'en inférer de potentielles nouvelles pour *A.thaliana*. Cette comparaison montre que 12 des 41 FP trouvés par la détection de variants sont des positions modifiées chez l'humain et/ou la levure (Le détail des positions se situe en Annexes A.1). Si on part de l'hypothèse que les positions conservées chez la levure et l'humain, le sont aussi pour A.thaliana alors la détection de

variants prédit potentiellement 61 TP pour 29 FP ce qui augmente sa précision à 67,78% et le recall à 61,62%, ce qui en ferait la meilleure précision par rapport au porc et à l'humain.

Pour évaluer la contribution de la profondeur sur la qualité des résultats, nous avons réalisé des analyses plus fines.

c. Analyse de la contribution de la profondeur sur la qualité des résultats

Une première vue d'ensemble sur les données Hek293 et les données porcines est effectuée figure 14, représentant la distribution de la profondeur afin d'avoir une idée de sa répartition sur l'ARNr.



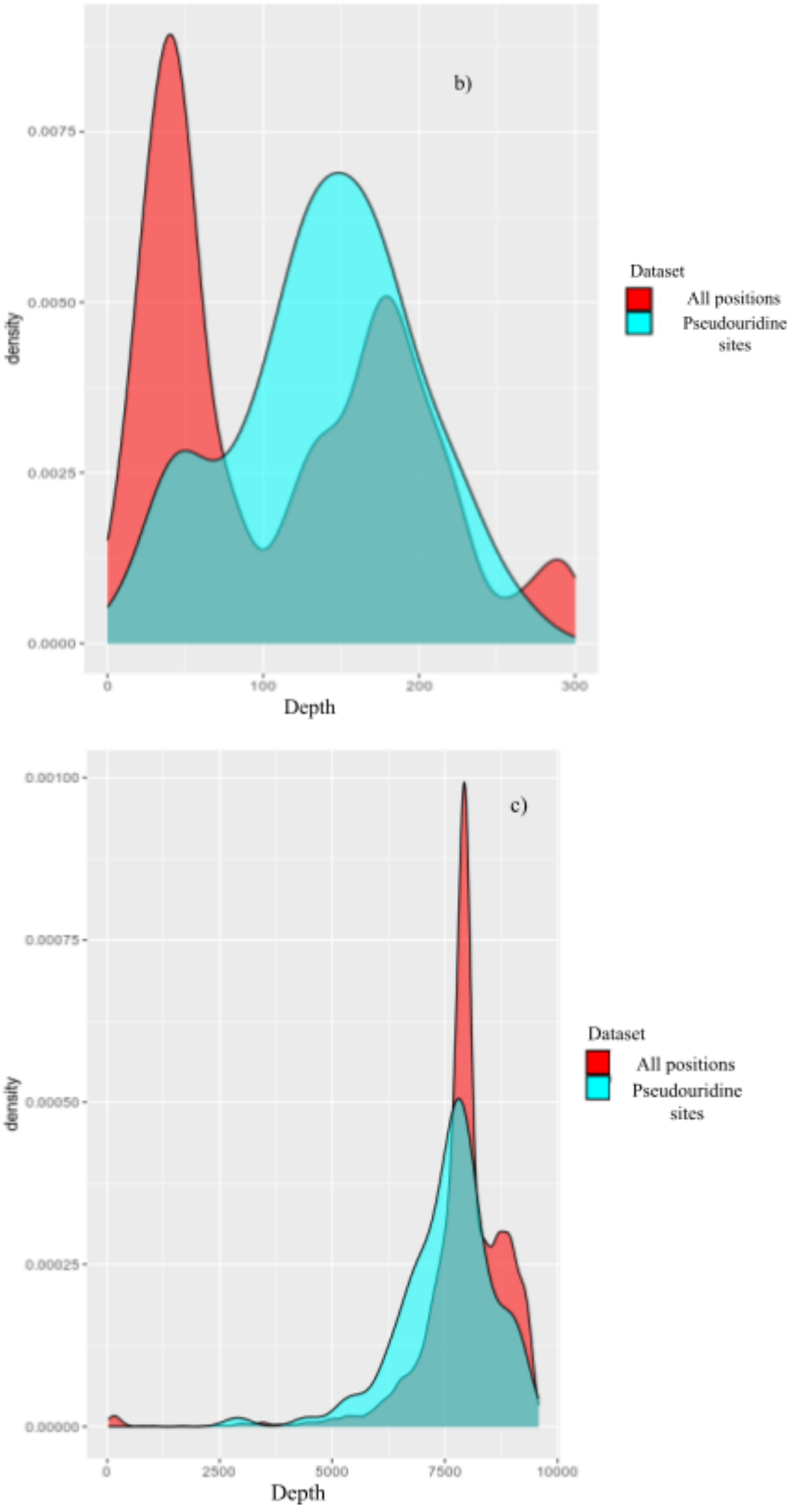


Figure 14 : Répartition de la profondeur en fonction de toutes les positions dans l'ARNr (rouge) ainsi que les sites de Pseudouridines connues (cyan) pour a) les données Hek293 et b) les données porcines et c) les données d'*Arabidopsis thaliana*

Au niveau des profondeurs, la répartition n'est pas la même selon si on est chez le porc ou l'humain. D'une part pour les cellules humaines Hek293, la majorité (74,93%) des positions ont tendance à être couvertes par une profondeur aux alentours de 50. La couverture des sites de Pseudouridine montre une courbe suivant cette même tendance, avec une couverture assez faible. D'autre part, le constat est assez similaire si ce n'est une profondeur plus élevée, généralement supérieure à 100 chez le porc. Pour les sites de Pseudouridines, la couverture est globalement autour de 150 de profondeur. Pour des soucis de lisibilités de la figure, la profondeur au-delà de 300 n'est pas montrée, néanmoins on constate un très léger pic vers 4500 recouvrant des sites de Pseudouridines. Dans le cas d'*Arabidopsis thaliana*, la profondeur moyenne se situe environ à 7500-8000, ce qui représente au moins 150 fois la profondeur moyenne des autres données. Ici aussi la couverture des sites de Pseudouridine suit celle de toutes les positions. La comparaison avec ces profondeurs permet par la suite d'estimer le gain d'information obtenu lorsque la profondeur augmente.

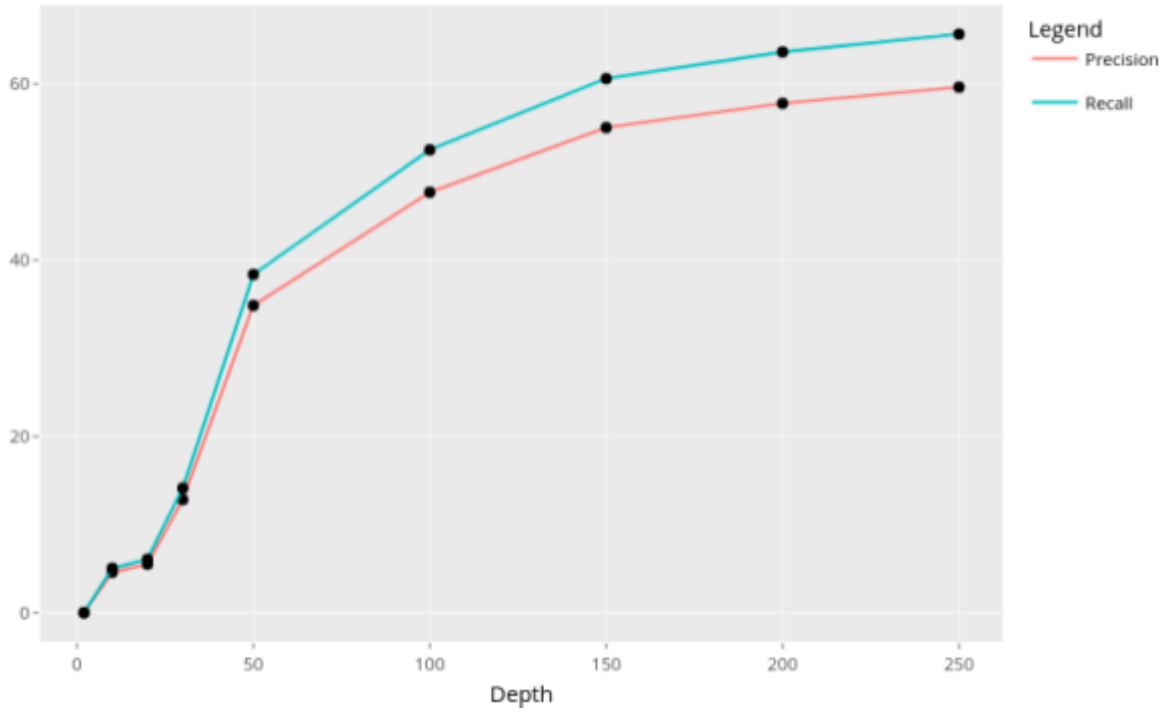


Figure 15 : Evolution de la précision et du recall en fonction de la profondeur pour les données Hek293.

Ce qui est observable dans la figure 15, c'est que la précision et le recall s'améliorent lorsque la profondeur augmente. Le recall a tendance à plus augmenter que la précision. Les résultats et allures des courbes sont similaires chez le porc. De plus, le gain de précision et de recall a tendance à se stabiliser au-delà d'une certaine profondeur. La suite s'intéresse à comparer les

différentes classes prédites afin de déterminer s' il y a un lien entre les classes prédites (TP, FP, FN) et la profondeur associée.

Figure 16 : Répartition de la profondeur en fonction des classes (FN, FP et TP) pour a) Hek293 et b) le porc.

Comme illustré dans la figure 16, il n'y a pas de détermination nette entre les différentes classes. La répartition reste assez homogène dans son ensemble et par conséquent la profondeur seule ne permet pas d'expliquer la répartition des prédictions. Néanmoins pour les

FN, 2 points se distinguent chez l'humain et 3 chez le porc. En effet, ces points ont une profondeur élevée mais n'ont pas été prédits positifs. Par ailleurs, parmi ces points, 2 sont communs entre le porc et l'humain. Il s'agit des positions 4491 et 4606 de la sous-unité 28S. En analysant leur composition en bases, le C est présent faiblement (moins de 20% pour les 2 FN communs). Cela peut illustrer le côté restrictif de la méthode qui ne base pas sa prédiction sur les reads (prédit une valeur uniforme pour une position donnée), puisque le faible taux de C n'est pas lié à un faible taux de reads ayant une Pseudouridine.

d. Application des filtres

Les résultats des différentes prédictions ont amené à la construction de filtres dont le but est d'essayer d'améliorer les prédictions en augmentant le nombre de TP et en diminuant le nombre de FP.

Filter	TP	FP	FN	Precision	Recall
No	65	44	34	59.36 %	65.66 %
C > A	63 (-2)	39 (-5)	36 (+2)	61.76 %	63.64 %
C > A & C > G	62 (-3)	38 (-6)	37 (+3)	62 %	62.63 %
C > T	34 (-31)	9 (-35)	37 (+3)	79.09 %	34.34 %
C > A & C + T ≥ 80	60 (-5)	38 (-6)	51 (+5)	61.22 %	60.61 %
C + T ≥ 50 & C ≥ 25 & C > A	57 (-8)	24 (-20)	42 (+8)	70.37 %	57.58 %
C > 50	32 (-33)	9 (-35)	67 (+33)	78.05 %	32.32 %
C ≥ 25	57 (-8)	25 (-19)	42 (+8)	69.51 %	57.58 %
QUAL ≥ 20	54 (-11)	28 (-16)	45 (+11)	65.85 %	55.88 %
Strict	42 (-23)	29 (-15)	57 (+23)	59.15 %	42.42 %

Table 1 : Filtres appliqués sur les données issues du séquençage des cellules humaines Hek293.

La table 1 montre les différents types de filtres utilisés et leurs effets sur les résultats des prédictions pour les données Hek293. Globalement, l'application de filtres ne permet pas une amélioration nette des résultats. Il faut toutefois noter que certains filtres permettent

d'améliorer grandement la précision mais au détriment du recall (composition en base C qui doit être supérieur à 50% dans la table 1 et dans la table 2). Cependant, ces derniers permettent de prioriser, c'est-à-dire d'être sûr des TP prédits. C'est le cas du filtrage C > A (la composition en C supérieure à la composition en A), car ce qui est souvent observé sont des mésappariements T vers C associés à des mésappariements T vers A, ce qui se traduit dans le fichier VCF par ces deux bases présentes en tant que bases alternatives. C'est un phénomène visible notamment chez *Arabidopsis thaliana*. Ce filtrage a pour conséquence qu'environ 3% des TP ne sont plus prédits mais environ 11% des FP sont perdus. La même situation est observable dans la table 2 pour Arabidopsis thaliana où le pourcentage de TP perdus est le même mais environ 14% des FP ne sont plus présents.

Filter	TP	FP	FN	Precision	Recall
No	49	41	50	54.44 %	49.49 %
C > A	47 (-2)	35 (-6)	53 (+3)	57.32 %	47.47 %
C ≥ 50	30 (-19)	18 (-23)	69 (+19)	62.50 %	30.30 %
C ≥ 25	46 (-3)	35 (-6)	53 (+3)	56.79 %	46.46 %
C > T	33 (-16)	24 (-17)	66 (+16)	57.89 %	33.33 %
QUAL ≥ 20	43 (-6)	33 (-8)	56 (+6)	56.58 %	43.43 %

Table 2 : Filtres appliqués sur les données issues du séquences d'*Arabidopsis thaliana*.

Pour les données porcines, l'application de filtres ne donne pas les mêmes résultats. En effet, le constat est que l'on perd autant de TP que de FP quel que soit le filtre appliqué.

e. Résultats avec graphmap

L'objectif d'utiliser graphmap au lieu de minimap2 pour l'alignement est de comparer les potentielles différences entre les deux outils en termes d'alignement,principalement sur la détection de variants effectués.

En ce qui concerne les alignements sur les ARNr humains avec les données Hek293, une première constatation est que graphmap aligne 769 reads soit 0,06% (contre 0,04% pour minimap2). En revanche l'alignement des données porcines sur cet ARNr n'aligne quant à lui

que 0,03% des reads (contre 0,06% avec minimap2), ce qui constitue une perte de plus de 50% de la sensibilité.

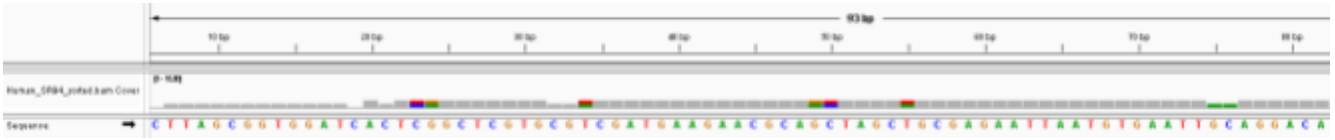


Figure 17: Visualisation de l'alignement des données de séquençage humaines Hek293 avec grahpmap au niveau de la sous-unité 5.8S sur IGV. La composition des bases est montrée selon un code couleur, rouge pour T, marron pour G, bleu pour C et vert pour A.

Une autre constatation, visible figure 17, est qu'un des jeux de données qui ne s'alignait auparavant pas sur cette sous-unité s'aligne avec graphmap. graphmap est plus sensible pour aligner les données humaines sur l'ARNr humain.

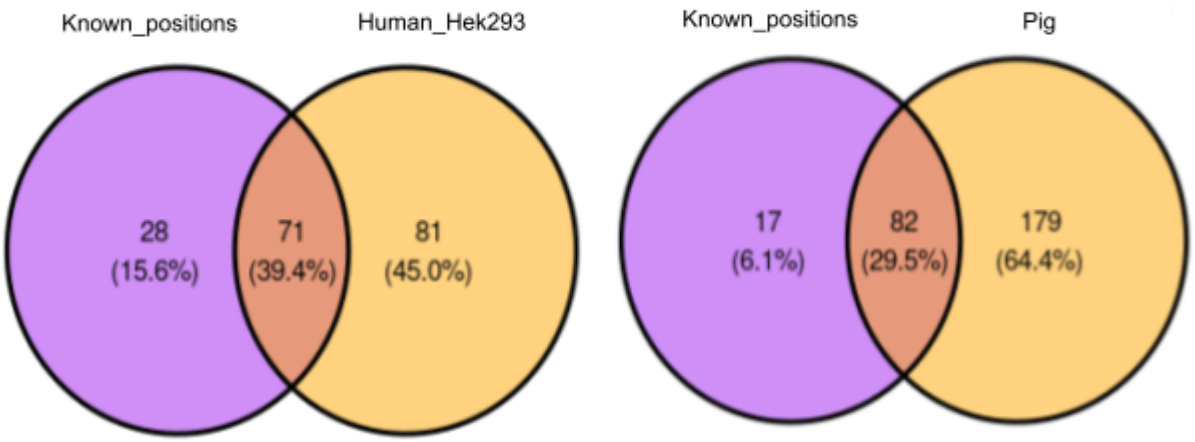


Figure 18 : Comparaison des Pseudouridines connues de la base de données avec les variants prédits par la méthode d'appel de variant pour les cellules Hek293 (à gauche) et pour le porc (à droite) à partir de l'alignement effectué avec GraphMap.

De plus, lors de la nouvelle détection de variants sur ces alignements illustrés figure 18, que ce soit chez les données Hek293 ou celles du porc, il y a une augmentation globale des TP. Pour les données Hek293, il y a 6 TP de plus détectés ce qui correspond à une augmentation de 9,23% et pour les données porcines une augmentation de 18 TP soit 28,125%. Cependant, bien qu'il y ait cette augmentation des TP, l'augmentation des FP est encore plus forte. Pour les données Hek293, 37 nouveaux FP sont prédits soit une augmentation de 84,04% ainsi que 142 nouveaux FP prédits pour les données porcines soit une augmentation de 383,78%. Cela

démontre que, malgré une meilleure sensibilité de l'outil avec l'utilisation de graphmap, la proportion de gain TP et FP n'est pas satisfaisante.

2. Détection de pseudo-uridylations par apprentissage avec Penguin

a. Description globale des événements alignés pour les données Hek293

Afin d'avoir une meilleure compréhension du fonctionnement de l'apprentissage, il est intéressant d'avoir une description du nombre d'événements à une position donnée, le nombre d'événements par read ainsi que le nombre de reads s'alignant à une position de l'ARNr.

Tout d'abord, pour les données Hek293 du fichier « Hek293_eventalign » contenant tous les événements alignés sur l'ARNr de référence. Ce fichier contient un total de 101 256 événements. Un fait important est que certaines positions de l'ARNr n'ont pas d'événement associé. C'est le cas notamment de la sous-unité 5.8S sur laquelle aucun événement ne s'aligne,ainsi que des positions 350-550 du 28S.

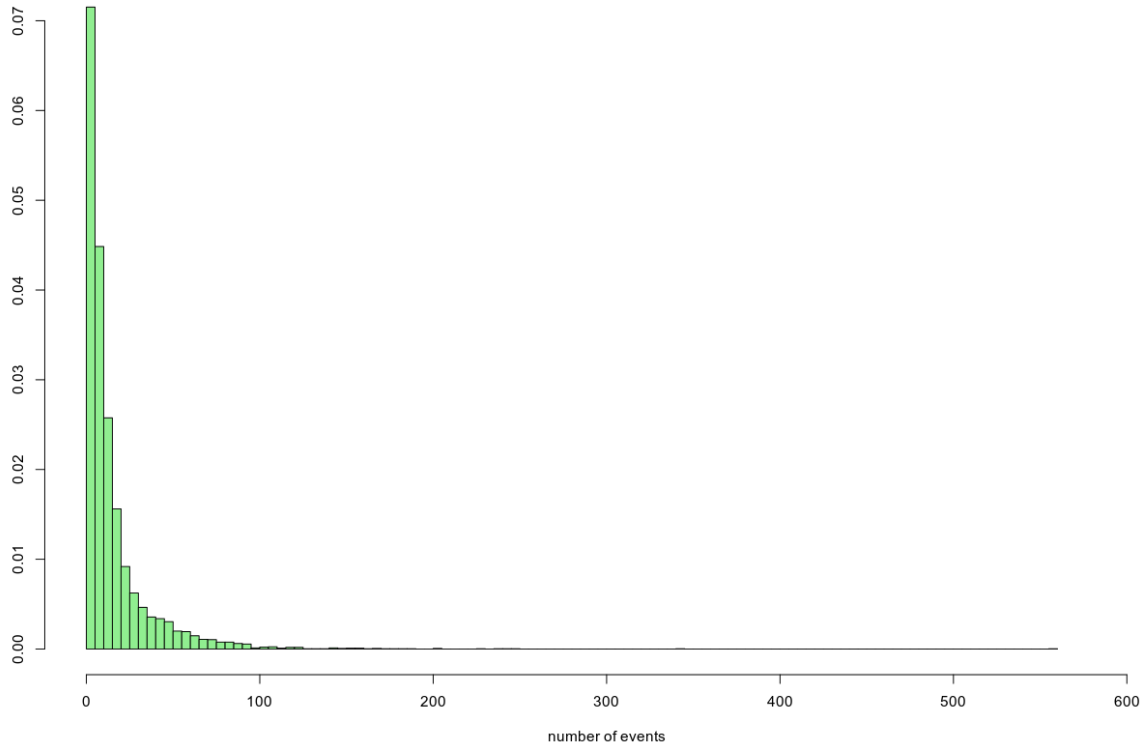


Figure 19 : Distribution du nombre d'événements pour une position donnée.

En s'intéressant aux nombres d'événements associés à une position donnée, la distribution de la figure 19, montre une répartition en forme de L. En moyenne, 9 événements s'alignent à une position donnée. Cependant des pics sont notables pour certaines positions. En effet, le nombre maximum d'événements pour une même position est de 559, qui sont alignés au niveau de la position 3806 de la sous-unité 28S de l'ARNr. Ensuite, pour l'ensemble des reads s'alignant sur les ARNr humains, en regardant uniquement les événements s'alignant sur un site de Pseudouridine, 21 événements en moyenne s'alignent par position avec un pic à 242 événements pour la position 4966 de la sous-unité 28S.

Concernant le nombre d'événements par read, en moyenne un read est associé à 1985 événements avec un maximum atteignant 6 766 événements pour un read. Étonnamment, sur les 513 reads s'alignant sur l'ARNr, seuls 51 reads sont associés à des événements (11 reads pour le 18S et 40 pour le 28S).

Enfin, sur les 40 reads associés à des événements sur le 28S, en moyenne 6 reads s'alignent sur une même position avec un maximum de 29 reads sur la position 4683 du 28S. Or, en moyenne, 9 événements sont associés à une même position. Cela signifie que pour certaines positions il y a plus d'événements que de reads. Normalement un seul événement est associé à la position donnée d'un read. En d'autres termes, lors de la détection des événements,

plusieurs événements ont été détectés par erreur au lieu d'un. Ces événements ont des caractéristiques du signal électrique qui varient mais qui restent toutes plausibles.

b. Prédictions de Penguin

Penguin a été lancé un première fois sur différents jeux de données (voir Section mise en œuvre de Penguin) et les résultats ont été synthétisés dans la table 4.

Train	Test	TP	TN	FP	FN	Predictions
Hek293_18S	Hek293_28S	25 (2)	401	14 (5)	390 (16)	830
Hek293_28S	Hek293_18S	17 (3)	47	3 (3)	33 (7)	100
Hek293_train	Hek293_test	113 (15)	152	5 (4)	44 (10)	314
Arabidopsis_18S	Arabidopsis_25S	86 447 (3)	425 923	16 533 (28)	356 009 (19)	884 912
Arabidopsis_25S	Arabidopsis_18S	6 347 (2)	86 689	5 752 (34)	86 094 (12)	184 882
Arabidopsis_train	Arabidopsis_test	239 397 (25)	219 263	20 299 (80)	165 (1)	479 124
Arabidopsis_train	Hek293_eventalign	219 (14)	400	65 (42)	246 (14)	930
Hek293_eventalign	Arabidopsis_test	291 409 (14)	515 525	25 876 (69)	249 992 (23)	1 082 802

Table 4 : Prédiction avec les jeux de données Hek293 et *Arabidopsis thaliana*. Les nombres correspondent aux prédictions des événements et entre parenthèses à combien de position unique sur l'ARNr de référence cela est associé.

Les tests avec uniquement la sous-unité 5.8S en tant que jeu d'apprentissage et jeu de test n'ont pas été incluse car la prédiction sur le 5.8S donne 0 TP, ce qui doit être lié au fait de n'avoir que deux positions connues de sites de Pseudouridine sur cette sous-unité et par conséquent une quantité d'informations (d'événements) insuffisant pour l'apprentissage et le test. Ce qui est observable dans la table 4, c'est que la séparation aléatoire des fichiers eventalign (Hek293_train, Hek293_test et Arabidopsis_train, Arabidopsis_test) donne de bons résultats avec une précision de 87% et un recall de 84.5% pour Hek293 ainsi qu'une

précision et un recall de 96% pour *Arabidopsis thaliana*. Dans le cas des prédictions entre les différentes sous-unités, la précision et le recall sont plus faibles, se trouvant aux alentours de 50-70%. Dans les cas où la prédiction sur *Arabidopsis thaliana* est faite par l'apprentissage des données Hek293, la précision et le recall augmentent par rapport à précédemment (sans dépasser la prédiction obtenue avec les fichiers séparés aléatoirement) avec 79,5% et 74,5% respectivement. Par contre, cette amélioration de la précision et du recall n'est pas constatée lorsque le jeu d'apprentissage est constitué des données d'*Arabidopsis thaliana* et la prédiction des données Hek293. Malgré ces bons résultats, lorsque l'on regarde le nombre de positions uniques sur l'ARNr auxquelles sont assignés des événements, ce nombre est faible. En effet, pour ces sous-unités de l'ARNr humain ainsi que pour *Arabidopsis thaliana*, le nombre de sites de Pseudouridines connues s'élève à 99. Or, dans ces prédictions, seule une petite partie est prédite. Le fichier Hek293_eventalign contenant tous les événements, on s'attend à ce que plus de positions soient incluses dans la prédiction alors qu'ici 28 (14 dans les TP et 14 dans les FN) positions de sites Pseudouridines le sont. Donc 71 positions ainsi que leurs événements associés ne sont pas pris en compte. De plus, un problème dans la prédiction illustrée table 5 émet la possibilité d'une erreur dans le script Penguin lançant le SVM.

Train	Test	Predicted as TP	Known position
Arabidopsis_train	Hek293_eventalign	18S_54	5.8S_54
Arabidopsis_train	Arabidopsis_test	At_25S_947	At_18S_947
Arabidopsis_train	Arabidopsis_test	At_25S_1310	At_18S_1310
Arabidopsis_train	Arabidopsis_test	At_25S_1701	At_18S_1701

Table 5 : Problème d'association sous-unité/position.

Lors des prédictions, des événements associés à des positions de l'ARNr ont été prédits en tant que TP, alors qu'elles ne devraient pas. Les événements associés à la position 54 de la sous-unité 18S a été prédite vraie positive. Or cette position du 18S ne fait pas partie des sites de Pseudouridines connues. En revanche, la position 54 du 5.8S, en l'occurrence, en fait partie. Cela est aussi visible pour les données d'*Arabidopsis thaliana*.

Lors de la construction du jeu d'apprentissage et de test, la base pour associer si tel ou tel événement est aligné sur une position de Pseudouridine ou non est faite en liant les positions

sur l'ARNr dans le fichier positions_modifications avec les positions sur l'ARNr du fichier eventalign sans prendre en compte sur quelle sous-unité est cette position. Ceci est illustré figure 20, seule une colonne est utilisée (la deuxième, celle de la position sur l'ARNr) pour construire les jeux de données. La première colonne correspondant à la sous-unité n'est pas prise en compte dans les deux fichiers.

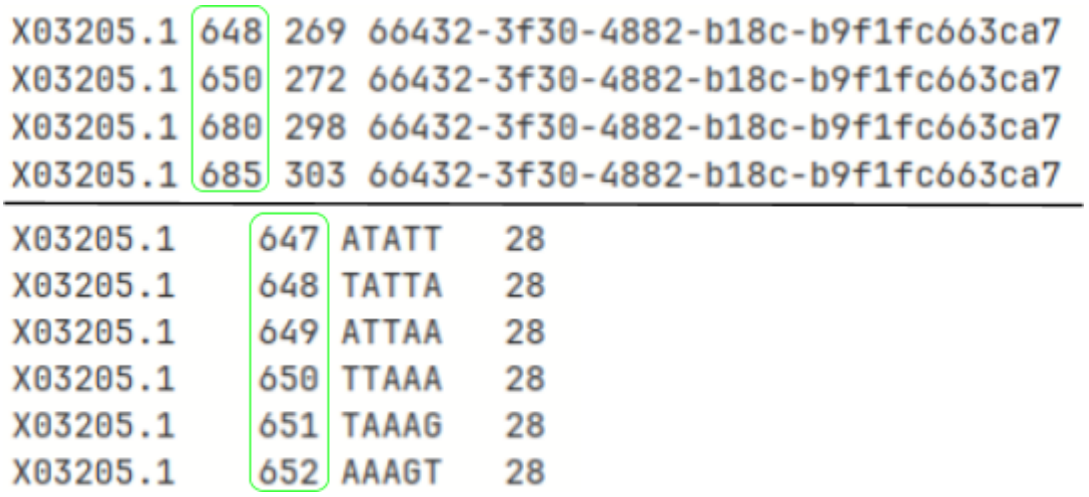


Figure 20 : Extrait du fichier positions_modifications (en haut) et du fichier Hek293_eventalign (en bas) pour les données Hek293. La colonne 1 de positions_modifications est celle de la sous-unité, la colonne 2 correspond à la position sur l'ARNr, la colonne 3 à la position du read qui s'y aligne et la colonne 4 à l'identifiant du read. En vert représente la colonne prise en compte lors de la correspondance entre les deux fichiers pour construire le jeu d'apprentissage et de test.

Sur seule base de la position, malgré le filtrage des k-mers en amont, les caractéristiques de la mauvaise sous-unité pour cette position sont prises en compte. À cause de cela, le jeu d'apprentissage et de test construits ont une association (0 ou 1, c'est-à-dire non Pseudouridine ou Pseudouridine) fausse puisqu'elle est faite sur la base de la mauvaise sous-unité et par conséquent le modèle va être entraîné sur les mauvaises caractéristiques du signal ainsi que du k-mer.

Outre ce problème, nous avons également identifié un autre bug, en lien avec la prise en compte uniquement de la position, qui se base sur le filtrage des k-mers. En effet, comme montré figure 21, le filtrage est fait à la condition du T au milieu, ce qui donne des k-mers du type NNTNN. Comme mentionné précédemment, l'association 0/1 s'effectue sur la base de la position de l'ARNr.

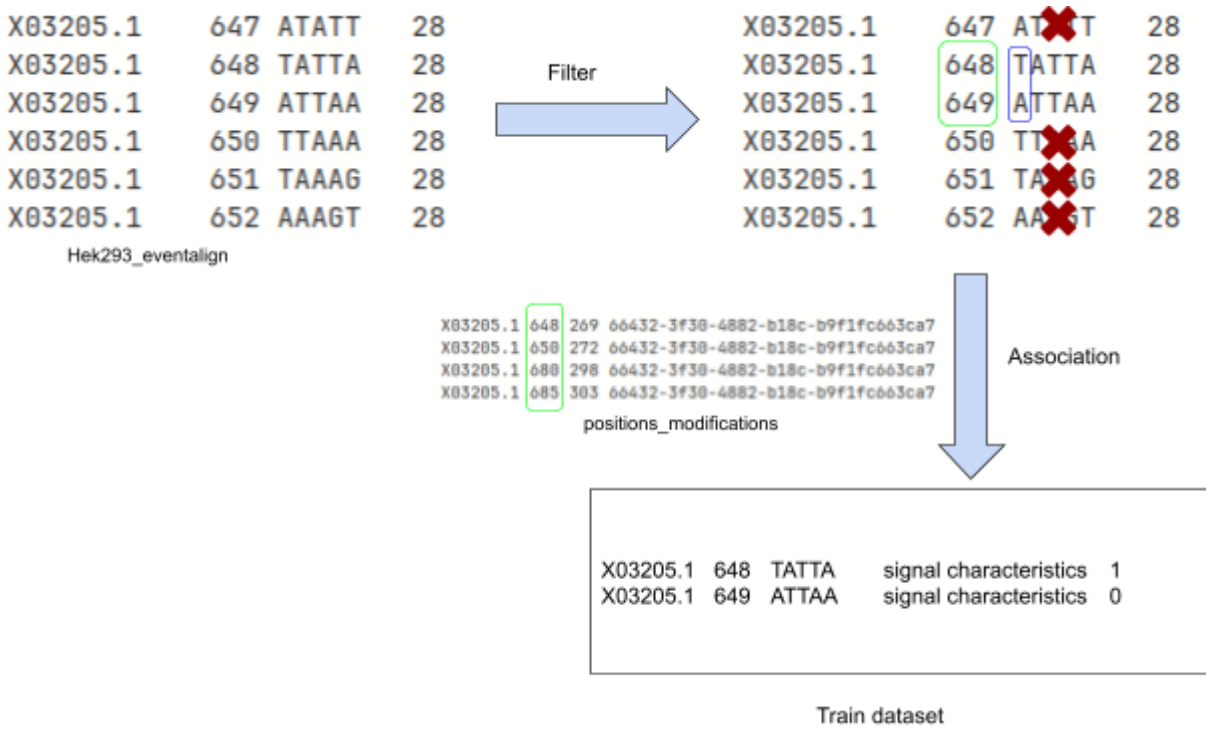


Figure 21: Schéma du filtrage et de l'association des données Hek293. Les croix rouges correspondent aux événements qui ne passent pas le filtre. La base du k-mer correspondant à la position de l'ARNr est entourée en bleu.

Or cette position correspond à la première base du k-mer. Par conséquent les événements liés à des pseudouridylations ont des k-mers de type TNTNN, avec la position de la potentielle Pseudouridine au niveau du premier T du k-mer et non au niveau du T du milieu. Cela a comme conséquence ce qui est illustré table 6.

Sub-unit	position	K-mer	Prediction
25S	**1310**	**A** A T G G	TP
18S	**1310**	T T A A T	Not predicted
18S	**1308**	**G** G T T A	TN

Table 6 : Résultats de prédiction avec les k-mers associés. En gras est la position correspondant à la base du k-mer et en rouge ce qui est connu comme étant un site de Pseudouridine.

La table 6 apporte l'information du k-mer par rapport à la précédente. Ce qui est montré ici, c'est que l'étape de filtrage d'association amène à des prédictions non cohérentes. En effet, ce qui était montré table 5 était le problème d'association entre la sous-unité et la position. Cela entraîne en une prédiction s'effectuant sur un A, la première base du k-mer en tant que TP. Le 1310 du 18S n'est pas prédit car l'étape de filtrage en amont ne l'inclut pas dans les jeux

d'apprentissage. Pour avoir la potentielle Pseudouridine au milieu du k-mer, il faut regarder deux positions en amont de celle-ci, c'est-à-dire au niveau de la position 1308. Cependant cette prédiction s'effectue sur une position correspondant à un G donc est prédit négativement (TN).

Un dernier point est que le signal électrique ne semble pas intervenir dans cette prédiction, et que l'apprentissage et la prédiction ne prennent en compte que les k-mers. C'est visible au niveau des FP et des FN prédits qui ont les mêmes k-mers. Lors de l'apprentissage, si un type de k-mers est plus souvent associé à un événement considéré de pseudouridylation, alors lors du test, ce même type de k-mer va être quasiment systématiquement être prédit positif et vice-versa. Ce processus est induit par la façon dont sont utilisées les données. En effet, tous les événements associées à une position donnée sont étiquetés de la même manière, qu'ils correspondent ou non à un variant selon l'appel de base. Lors de l'apprentissage des données Arabidopsis thaliana, le k-mer "TTTGG" était associé à une position connue de Pseudouridine. Lors de la prédiction sur les données Hek293, parmi les 6 positions de l'ARNr associé à ce k-mer, 4 ont été prédits positivement dont un TP et 3 FP.

c. Prédictions après modification du script

Suite à ces incohérences rencontrées, une proposition de modification du script a été incluse pour corriger les deux premiers problèmes (celle-ci se trouve en Annexes A.2). Les nouvelles prédictions sont présentées table 7.

Train	Test	TP	TN	FP	FN	Predictions
Hek293_train	Hek293_test	677 (68)	6 361	1 193 (178)	83 (21)	8 314
Arabidopsis_ train	Arabidopsis_ test	1 621 771 (81)	219 263	20 299 (80)	84 209 (20)	23 378 662
Arabidopsis_ train	Hek293_ eventalign	680 (44)	11 756	3 328 (234)	824 (53)	16 588
Hek293_ eventalign	Arabidopsis_ test	903 359 (47)	15 966 925	5 714 757 (293)	793 621 (52)	23 378 662

Table 7 : Prédiction avec les jeux de données Hek293 et *Arabidopsis thaliana* après modification du script de Penguin. Les nombres correspondent aux prédictions des événements et entre parenthèses à combien de position unique sur l'ARNr de référence cela est associé.

Ce qui observable, c'est que, d'une part, le nombre d'événements prédits positivement et négativement sont plus important, ce qui est lié aux changements du script, puisque ces changements induisent un passage de 64 combinaisons de k-mers possibles (TNTNN) pour les potentielles Pseudouridines, à 256 (NNTNN) dans la composition du jeu d'apprentissage et du jeu de test. Néanmoins, bien que les positions uniques de l'ARNr associées à des événements soient plus importantes, le nombre de FP et de FN le sont aussi. Ces augmentations sont cohérentes avec le fait que plus de k-mers en apprentissage vont être appris comme « positif » et « négatif », étant donné que Penguin apprend et prédit essentiellement sur les k-mers. Enfin, Penguin prédit les reads de la même manière. C'est-à-dire que parmi tous les événements s'alignant sur des sites de Pseudouridines, certains reads vont avoir ou non cette modification. Or, Penguin ne prend pas ce fait en compte.

IV. Conclusion

Dans le cadre des analyses effectuées dans ce stage, avec les données à disposition, la détection de variants est plus efficace que la méthode d'apprentissage proposée par le logiciel Penguin. Ce dernier aurait pu permettre une approche plus complémentaire et complète grâce à l'information sur le signal électrique. Toutefois, à cause de potentielles erreurs rencontrées lors de la constitution des jeux d'apprentissage et de test, en l'état, Penguin ne permet pas une fiabilité des prédictions. Malgré cela, l'apprentissage reste un atout à exploiter dans la détection de modifications dans les ARNs.

a. Perspective

Nous avons observé de multiples événements pour une position et un read donné, il serait intéressant d'approfondir la raison de ce phénomène et si cela est lié à des modifications particulières. De plus, au cours de mon stage, dans l'optique de prendre en compte le paramètre d'un read modifié (que Penguin ne fait pas), c'est-à-dire à partir de l'alignement minimap2, de voir quel read a un C à la place d'un U (erreur lors de l'appel de base) pour une position de site de Pseudouridine connue sur l'ARNr, j'ai commencé à construire un modèle d'apprentissage qui prend en compte, pour chaque read s'alignant à une position d'un site de pseudouridylation dans l'ARNr, si le read est modifié ou non. L'objectif est de prendre les fichiers d'alignements eventalign en ajoutant une colonne constituant l'étiquette

(Pseudouridine ou non Pseudouridine) afin de construire un jeu d'apprentissage et un jeu de test afin qu'ensuite ils soient directement exploités en entrée d'un RF. Il est même possible d'intégrer une méthode semi-quantitative (Tavakoli *et al*, 2023) dans un modèle d'apprentissage. Enfin d'autres méthodes utilisant l'apprentissage peuvent être utilisées pour prédire les Pseudouridines telles que nanoRMS (Begik *et al*, 2021) ou EpiNano ((Liu *et al*, 2019),https://github.com/novoalab/EpiNano).

b. Conclusion personnelle

D'un point de vue personnel, ce stage m'a permis d'une part, de mettre en application et de consolider mes connaissances en bioinformatiques, ainsi qu'en programmation R et python. D'autre part, j'ai pu acquérir de l'expérience dans le monde professionnel et développer mon autonomie. J'en ai tiré des enseignements notamment au niveau de l'esprit critique vis-à-vis d'un article publié présentant un logiciel. En effet, un logiciel paraissant simple en termes de compréhension et de prise en main à première vue s'est révélé beaucoup plus complexe voire inexact dans sa méthode.

V. Bibliographie

Begik O, Lucas MC, Pryszcz LP, Ramirez JM, Medina R, Milenkovic I, Cruciani S, Liu H, Vieira HGS, Sas-Chen A, *et al* (2021) Quantitative profiling of pseudouridylation dynamics in native RNAs with nanopore sequencing. *Nat Biotechnol* 39: 1278–1291

Breiman L (2001) Random Forests. *Mach Learn* 45: 5–32

Brown JWS, Echeverria M, Qu L-H, Lowe TM, Bachellerie J-P, Hüttenhofer A, Kastenmayer JP, Green PJ, Shaw P & Marshall DF (2003) Plant snoRNA database. *Nucleic Acids Res* 31: 432–435

Chen H-M & Wu S-H (2009) Mining small RNA sequencing data: a new approach to identify small nucleolar RNAs in Arabidopsis. *Nucleic Acids Res* 37: e69

Cortes C & Vapnik V (1995) Support-vector networks. *Mach Learn* 20: 273–297

Ganot P, Bortolin M-L & Kiss T (1997) Site-Specific Pseudouridine Formation in Preribosomal RNA Is Guided by Small Nucleolar RNAs. *Cell* 89: 799–809

Gurney K (1997) An Introduction to Neural Networks 0 ed. CRC Press

Hassan D, Acevedo D, Daulatabad SV, Mir Q & Janga SC (2022) Penguin: A tool for predicting pseudouridine sites in direct RNA nanopore sequencing data. *Methods* 203: 478–487

Kiss DJ, Oláh J, Tóth G, Varga M, Stirling A, Menyhárd DK & Ferenczy GG (2022) The Structure-Derived Mechanism of Box H/ACA Pseudouridine Synthase Offers a Plausible Paradigm for Programmable RNA Editing. *ACS Catal* 12: 2756–2769

Lestrade L & Weber MJ (2006) snoRNA-LBME-db, a comprehensive database of human H/ACA and C/D box snoRNAs. *Nucleic Acids Res* 34: D158–D162

Li H (2011) A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics* 27: 2987–2993

Liu H, Begik O, Lucas MC, Ramirez JM, Mason CE, Wiener D, Schwartz S, Mattick JS, Smith MA & Novoa EM (2019) Accurate detection of m6A RNA modifications in native RNA sequences. *Nat Commun* 10: 4079

Lovejoy AF, Riordan DP & Brown PO (2014) Transcriptome-Wide Mapping of Pseudouridines: Pseudouridine Synthases Modify Specific mRNAs in S. cerevisiae. *PLoS ONE* 9: e110799

Sun L, Xu Y, Bai S, Bai X, Zhu H, Dong H, Wang W, Zhu X, Hao F & Song C-P (2019) Transcriptome-wide analysis of pseudouridylation of mRNA and non-coding RNAs in Arabidopsis. *J Exp Bot* 70: 5089–5600

Tavakoli S, Nabizadeh M, Makhamreh A, Gamper H, McCormick CA, Rezapour NK, Hou Y-M, Wanunu M & Rouhanifard SH (2023) Semi-quantitative detection of pseudouridine modifications and type I/II hypermodifications in human mRNAs using direct long-read sequencing. *Nat Commun* 14: 334

What is Oxford Nanopore Technology (ONT) sequencing? *Yourgenome · Sci Website*

VI. Annexes

At_25S_653	TCTTG	
At_25S_654	CTTGA	
At_25S_675	TCTGA	
At_25S_1129	ATTTT	
At_25S_1440	TCTTG	
At_25S_1472	ACTTT	
At_25S_1577	CTTCG	position modifiée chez l'homme
At_25S_1842	CATCA	
At_25S_1849	TCTCC	
At_25S_1884	TGTAG	
At_25S_2248	ACTAT	position modifiée chez l'homme et la levure
At_25S_2250	TATGA	position modifiée chez l'homme et la levure
At_25S_2284	TCTAA	
At_25S_2304	AATGG	position modifiée chez l'homme et la levure
At_25S_2318	ATTCC	position modifiée chez l'homme
At_25S_2411	ACTCT	
At_25S_2643	TGTCC	
At_25S_2674	TCTCG	
At_25S_2808	GATAA	
At_25S_2872	TCTTC	
At_25S_2873	CTTCC	
At_25S_2911	ATTGT	
At_25S_2914	GTTCA	
At_25S_2944	GTTTA	position modifiée chez l'homme
At_25S_3290	CTTAA	
At_18S_37	TCTCA	position modifiée chez l'homme
At_18S_103	ATTAA	
At_18S_122	GTTTG	
At_18S_123	TTTGA	
At_18S_382	ATTCC	position modifiée chez l'homme
At_18S_465	GGTAG	
At_18S_605	ATTTA	
At_18S_842	GCTTC	
At_18S_1176	CCTGC	position modifiée chez l'homme
At_18S_1182	GCTTA	position modifiée chez l'homme et la levure
At_18S_1261	TCTAT	
At_18S_1263	TATGG	
At_18S_1291	GTTGG	position modifiée chez l'homme et la levure
At_18S_1381	CTTCT	position modifiée chez l'homme
At_18S_1550	GATCA	
At_18S_1584	ATTGC	

A.1 : Conservation des positions prédites en tant que Faux positifs par la détection de variants pour *A.thaliana*. La position indiquée correspond au k-mer du milieu (le troisième k-mer).

```
def filter_df(df1,df2,chrom):
    df_pseu = pd.DataFrame()
    df_U = pd.DataFrame()
    for el in chrom :
        df1_sub = df1[df1.iloc[:,0] == el]
        df2_sub = df2[df2["contig"] == el]
        x_sub = list(set(df1_sub.iloc[:,1]-2).intersection(set(df2_sub.iloc[:,1])) )
        df_pseu_sub = df2_sub[df2_sub['position'].isin(x_sub)]
        df_pseu = df_pseu.append(df_pseu_sub)
        df_U_sub = df2_sub[~df2_sub['position'].isin(x_sub)]
        df_U = df_U.append(df_U_sub)

    return df_pseu, df_U

chrom_train = list(set(df1.iloc[:,0]))
chrom_test = list(set(df3.iloc[:,0]))
```

A.2 : Modifications du script SVM_validate.py de Penguin (https://github.com/AnnabBru/Penguin)