UNIVERSITÉ DU QUÉBEC À MONTRÉAL

MÉTHODES EFFICACES POUR L'EXTRACTION D'ATTRIBUTS ET LA CLASSIFICATION DE SÉQUENCES GÉNOMIQUES VIRALES BASÉES SUR UNE APPROCHE INDÉPENDANTE DE L'ALIGNEMENT

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

PAR

DYLAN LEBATTEUX

JUILLET 2019

REMERCIEMENTS

Je tiens dans un premier temps à remercier mon père pour son soutien général sans faille, sans lequel je n'aurais pu accomplir cette maîtrise. Il est et restera pour moi un exemple et je suis fier d'être son fils.

Dans un deuxième temps, je souhaite exprimer ma gratitude envers mon directeur de recherche, le professeur Abdoulaye Baniré Diallo, pour l'ingéniosité avec laquelle il a su diriger ma maîtrise ainsi que pour la pertinence et la sagesse de ses conseils. J'espère avoir l'occasion d'effectuer mon doctorat sous sa supervision.

Dans un troisième temps, je remercie les membres de mon laboratoire, Mohamed Amine Remita, pour son professionnalisme ainsi que pour les connaissances et l'expérience précieuses qu'il m'a partagées. Golrokh Kiani, pour la disponibilité et la générosité dont elle a fait preuve à mon égard. Julie, pour la rigueur et le dynamisme qu'elle transmet. Ainsi que Steve, Malick, Bruno et Faten, pour les échanges enrichissants que nous avons eu l'occasion d'entretenir.

Je tiens à remercier ma femme, Sara, de m'avoir accompagné et soutenu durant ces deux dernières années dans mes projets. J'aspire ainsi, pouvoir continuer à avancer dans la vie à ses côtés.

Enfin, je tiens à exprimer mes sincères remerciements à toutes les personnes qui ont contribué, de près ou de loin, au bon déroulement de mon projet de maîtrise.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

RÉSUMÉ

Les avancées technologiques des dernières années dans le séquençage des biomolécules ont eu pour conséquence la génération d'immenses quantités de séquences et de données biologiques. Dans les domaines de la virologie et l'épidémiologie exploitant la biologie moléculaire, l'avènement de ces quantités massives de données ont apporté de nouveaux défis aux disciplines de l'analyse des séquences biologiques et de leur classification. En effet, la classification des pathogènes appartenant aux virus émergents et ré-émergents présente des intérêts majeurs au sein des études taxonomiques, de la génomique fonctionnelle, de l'interaction hôte-pathogène, ainsi que dans la prévention et le traitement des maladies. Elle consiste à assigner une séquence donnée à son groupe apparenté de séquences connues partageant des traits et caractéristiques similaires. Cependant, les méthodes de classification actuelles sont souvent confrontées à de nombreuses contraintes : performance globale de prédiction, dépendance à l'alignement impliquant des difficultés face aux pathogènes à forte variation génomique, spécificité des méthodes à certains types de virus, temps et coût de traitement ou encore interprétabilité des décisions. Dans ce mémoire, nous introduisons CASTOR-KRFE, une méthode indépendante de l'alignement basée sur l'apprentissage automatique. Cette dernière détecte les sous-séquences nucléotidiques discriminantes au sein de séquences pathogènes connues dans l'objectif de classifier précisément celles encore inconnues. Nous avons évalué notre approche sur de nombreux jeux de données constitués, couvrant les différents groupes de virus. Enfin, CASTOR-KRFE a été comparée aux prédicteurs spécialisés du virus de l'immunodéficience humaine ainsi qu'à des méthodes populaires et récentes dans les domaines de la classification virale et de l'extraction d'attributs basés sur les k-mers (sous-séquences nucléotidiques de longueur k). Notre nouvelle méthode sera prochainement incluse sur la plateforme web CASTOR, disponible à cette adresse : `http://castor.bioinfo.uqam.ca`.

Mots-clés : Classification des virus, Séquences nucléotidiques, Apprentissage automatique, Extraction d'attributs

PUBLICATIONS

Durant ce projet de maîtrise, nous avons développé CASTOR-KRFE, qui est une nouvelle méthode indépendante de l'alignement. Elle permet d'extraire à partir de séquences virales connues des attributs basés sur les k-mers. Ces attributs sont ensuite combinés à des algorithmes d'apprentissage supervisé afin de constituer des modèles prédictifs pour les séquences génomiques virales encore inconnues.

Cette nouvelle approche nous a donné l'occasion de réaliser plusieurs publications. Nous avons publié nos résultats préliminaires à travers une affiche à la conférence *RECOMB* en avril 2018 à Paris. Il s'en est suivi deux articles de conférences où nous avons présenté le concept et les algorithmes de base : *Joint ICML and IJCAI Workshop on Computational Biology* en juillet 2018 à Stockholm, et *Rencontres de la Société Francophone de Classification* en septembre 2018 à Paris, où j'ai réalisé des présentations orales de mon projet. Enfin, nous avons soumis dans *Journal of Computational Biology* une version étendue et complète, qui a été acceptée en décembre 2018.

Dans ce projet, j'ai conçu l'approche CASTOR-KRFE, réalisé les évaluations ainsi que la majorité des ensembles des données de ces dernières. Les coauteurs m'ont assisté dans les étapes de rédaction des articles ainsi que dans la mise en place des évaluations. Mohamed Amine Remita m'a également apporté son aide lors de l'implémentation finale de CASTOR-KRFE.

Liste des publications :

1. **Dylan Lebatteux**, Mohamed Amine Remita, and Abdoulaye Baniré Diallo. "CASTOR-KRFE : an alignment-free method to extract discriminant genomic subsequences and its application to diverse and complex virus groups classification." In *The 22nd Annual International Conference on Research in Computational Molecular Biology, April 21-24, 2018, Paris.* RECOMB, 2018. *(Présentation par affiche).*
2. **Dylan Lebatteux**, Mohamed Amine Remita, and Abdoulaye Baniré Diallo. "Toward an Alignment-Free Method for Feature Extraction and Accurate Classification of Viral Sequences." In *Joint ICML and IJCAI Workshop on Computational Biology, Stockholm, Sweden, July, 2018. (Présentation orale et par affiche).*
3. **Dylan Lebatteux**, Mohamed Amine Remita, and Abdoulaye Baniré Diallo. "Une méthode sans alignement pour l'extraction d'attributs améliorant la classification des séquences virales." In *XXV èmes Rencontres de la Société Francophone de Classification 5-7 septembre 2018, Paris, France. (Présentation orale).*
4. **Dylan Lebatteux**, Amine M. Remita, and Abdoulaye Baniré Diallo. "Toward an Alignment-Free Method for Feature Extraction and Accurate Classification of Viral Sequences." *Journal of Computational Biology (2019) (Publié).*

INTRODUCTION

Au cours des dernières années, dans le domaine de la bio-informatique, les améliorations des technologies de séquençage d'ADN n'ont cessé de progresser. Ces avancées technologiques ont eu pour conséquence une augmentation sans précédent des données génomiques (Goodwin *et al.*, 2016). Particulièrement dans le champ de recherche de la biologie moléculaire virale, ces immenses quantités de données ont apporté de nouveaux défis dans les disciplines de la classification et de l'analyse des séquences génomiques virales. En effet, la classification des séquences génomiques virales est une pratique fondamentale dans différents domaines de la recherche en microbiologie. Elle consiste à assigner une séquence donnée inconnue à un groupe relatif de séquences connues partageant des traits et caractéristiques similaires. Les défis de telles prédictions peuvent être associés à plusieurs propriétés biologiques virales, incluant les recombinaisons, les taux de mutation, la multiplicité des sous-séquences nucléotidiques discriminantes et la diversité des types et sous-type de virus. En outre, une prédiction précise des séquences génomiques virales peut avoir des impacts majeurs positifs dans de nombreux domaines associés. En médecine, cela peut aider à déterminer la pathogénicité et la virulence des souches virales, à développer des vaccins et à étudier l'épidémiologie microbienne ainsi que la résistance aux médicaments (Struck *et al.*, 2014). En biologie, cela permet d'améliorer les études phylogénétiques et fonctionnelles des virus, la reconnaissance automatique des organismes et la caractérisation des régions génomiques (Van Belkum *et al.*, 2001). Enfin, dans le domaine de la classification de texte, les séquences biologiques répertoriées offrent depuis quelques années un support de qualité pour le développement et l'évaluation de nouveaux algorithmes (Xing *et al.*, 2010). Plu-

sieurs méthodes ont été mises en application à travers de nombreux outils de classification pour les séquences virales. Cependant, certaines contraintes persistent encore : performance, spécificité des méthodes à certains types de virus, complexité spatiale et temporelle des algorithmes ou encore interprétabilité des décisions. À travers ce document, nous exposerons CASTOR-KRFE, une méthode permettant d'extraire au sein de séquences génomiques virales des ensembles minimaux d'attributs basés sur les sous-séquences nucléotidiques discriminantes de longueur k (k-mers). Ces attributs, compréhensibles à l'échelle humaine, sont utilisés pour construire des modèles de prédiction qui tentent de maximiser les performances de classification des nouvelles instances inconnues. Afin d'évaluer cette approche, de nombreux ensembles de données viraux ont été collectés et constitués depuis les bases de données de séquences biologiques virales disponibles. Enfin, CASTOR-KRFE a été comparée à des méthodes populaires et récentes dans les domaines de la classification virale et de l'extraction d'attributs basés sur les k-mers.

CHAPITRE I

NOTIONS PRÉLIMINAIRES DE BIO-INFORMATIQUE

1.1 Définition de la *bio-informatique*

Le terme *bio-informatique* a fait son apparition la première fois dans une publication scientifique (Hesper et Hogeweg, 1970). La *bio-informatique* se présente comme un domaine de recherche multi-disciplinaire incluant des sciences telles que la biologie, l'informatique, les mathématiques, la médecine, la physique et la chimie.

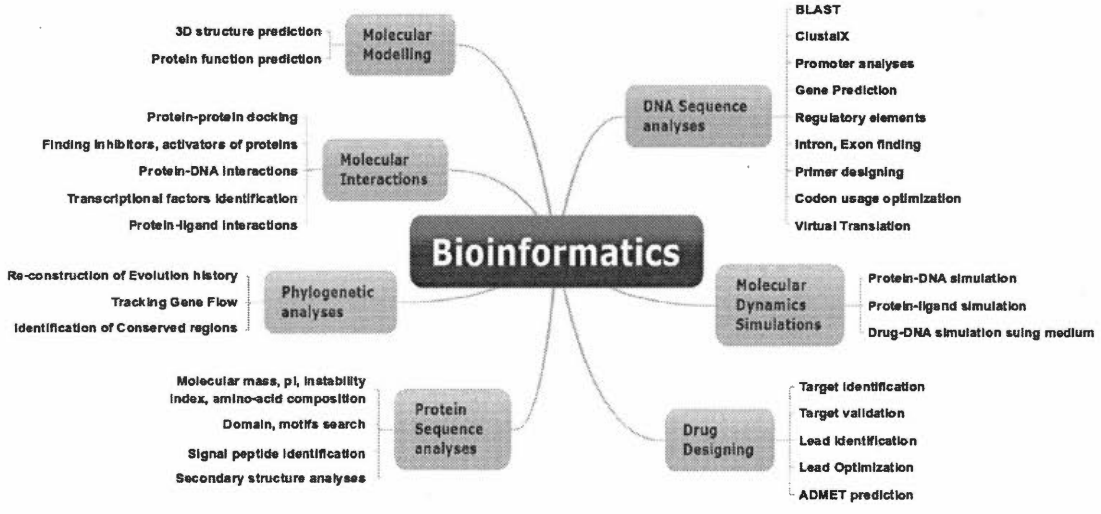

Figure 1.1 Domaines de la *bio-informatique* et outils associés. Source : (Mehmood *et al.*, 2014)

L'association de ces disciplines diverses a pour objectif central la résolution des problèmes posés par la biologie. Selon Luscombe *et al.* (2001), la *bio-informatique* se représente par trois grandes fonctions. Dans sa forme la plus basique, la première tâche est la collecte et l'organisation des données biologiques afin de les rendre facilement accessibles et exploitables par les chercheurs. Le deuxième objectif est le développement d'outils et des ressources qui permettent l'analyse et l'exploitation de ces données. Enfin, la troisième fonction est l'utilisation de ces outils afin d'extraire des informations biologiquement significatives à partir des données et de les interpréter. Dans Mehmood *et al.* (2014), les auteurs nous illustrent le domaine de la *bio-informatique*, les divers champs de recherche associés, ainsi que les types d'outils qui y ont été développés (Figure : 1.1).

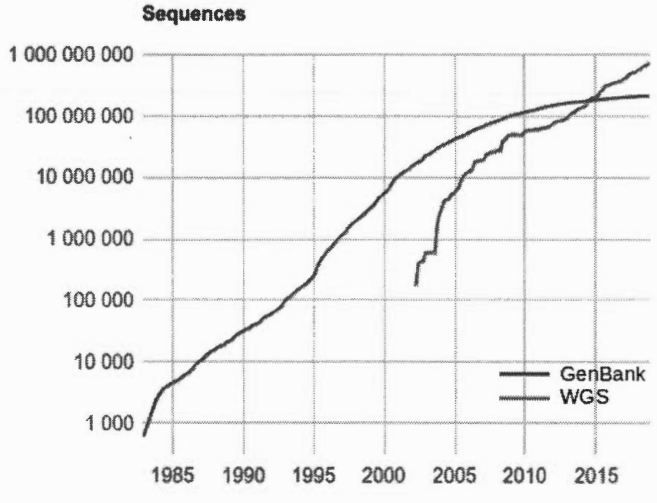

Figure 1.2 Evolution du nombre de séquences depuis les bases de données du NCBI

Figure illustrant l'augmentation du nombre de séquences biologiques disponibles sur les bases de données de GenBank et WGS du NCBI (National Center for Biotechnology Information). Statistiques disponibles à l'adresse suivante : `https://www.ncbi.nlm.nih.gov/genbank/statistics/`

Depuis quelques années, la *bio-informatique* connait une très forte expansion. Cela s'explique notamment par les avancées technologiques dans les outils de séquençages (Goodwin *et al.*, 2016) qui ont amenés un véritable déferlement de données de séquences biologiques (Figure : 1.2). Tout ceci ne cesse donc de faire croitre les challenges de la *bio-informatique.*

1.2 Acides nucléiques

En biologie, les acides nucléiques, tels que l'acide désoxyribonucléique (ADN) ou l'acide ribonucléique (ARN), sont des macromolécules (molécules constituées de la répétition de nombreuses sous-unités) dont les composants de base sont les nucléotides.

1.2.1 Nucléotides

Les nucléotides sont composés de trois éléments de base (Figure : 1.3) qui sont :

1. Une base nucléique (ou base azotée), elles sont séparées en deux types : puriques et pyrimidiques.

2. Un ose (Ribose pour l'ARN et Désoxyribose pour l'ADN) à cinq atomes de carbone (Pentose).

3. Un à trois groupes phosphates.

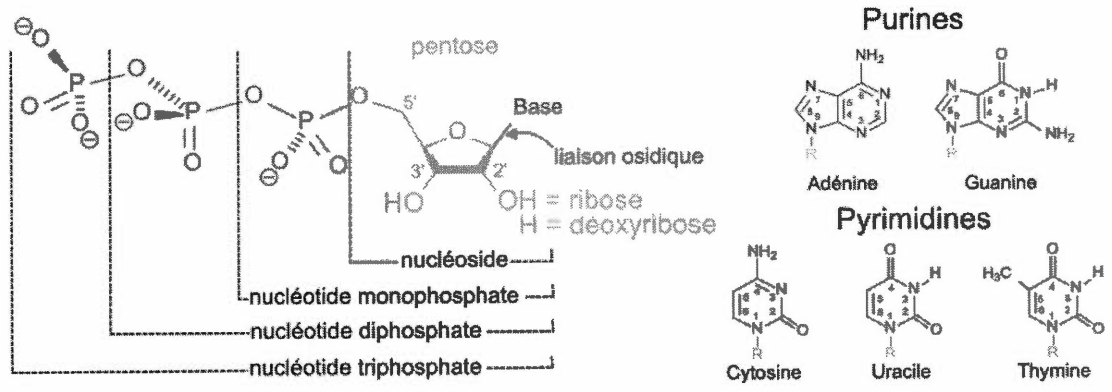

Figure 1.3 Structure des différents nucléotides. Source : `https://fr.wikipedia.org/wiki/Nucléotide`

1.2.2 Acide désoxyribonucléique (ADN)

L'ADN héberge l'ensemble de l'information génétique d'un organisme. Il contient des gènes codants (transcrits en ARN messagers), des gènes non codants (transcrits en non codants ou éléments transposables) ainsi que des régions non transcrites. Il est formé de deux brins antiparallèles (orientés en sens opposé) et enroulés l'un autour de l'autre (sous forme d'une double hélice). L'ADN est constitué de quatre désoxyribonucléotides qui se différencient par leurs bases nucléiques :

1. Le désoxyadénosine monophosphate (dAMP), dont la base nucléique est l'adénine (A)
2. La désoxycytidine monophosphate (dCMP), dont la base nucléique est la cytosine (C)
3. La désoxyguanosine monophosphate (dGMP), dont la base nucléique est la guanine (G)
4. La désoxythymidine monophosphate (dTMP), dont la base nucléique est la thymine (T)

1.2.3 Acide ribonucléique (ARN)

Les ARNs se divisent en deux catégories qui sont les ARNs codants et les ARNs non codants. L'ARN codant, qui est aussi appelé ARN messager (ARNm), est une copie simple brin de l'ADN qui sera traduite en protéine. Les ARNs non codants, quant à eux, sont impliqués dans différents mécanismes cellulaires tels que la traduction protéique (ARN ribosomique (ARNr) et ARN de transfert (ARNt)), la catalyse des réactions chimiques (Ribozyme) ou encore la régulation de l'expression génique (ARN interférent (ARNi)). L'ARN est aussi constitué de quatre ribonucléotides qui se différencient par leurs bases nucléiques :

1. L'adénosine monophosphate (AMP), dont la base nucléique est l'adénine (A)
2. La cytidine monophosphate (CMP), dont la base nucléique est la cytosine (C)
3. La guanosine monophosphate (GMP), dont la base nucléique est la guanine (G)
4. L'uridine monophosphate (UMP), dont la base nucléique est l'uracile (U)

1.2.4 Structure primaire

La représentation linaire des nucléotides du début à la fin d'un acide nucléique est appelée structure primaire (Figure : 1.4). Dans cette structure primaire, les nucléotides sont représentés par les lettres faisant référence à leur base nucléique. Les lettres 'A', 'C', 'G' et 'T' sont utilisées pour représenter les séquences d'ADN. Les séquences d'ARN quant à elles sont illustrées par le même alphabet à l'exception du 'T' qui est remplacé par le 'U'.

```
>gb:CY044090:3-1412|H1N1NA|Influenza A Virus
ATGAATCCAAACCAAAAGATAATAACCATTGGTTCGGTCTGTATGACAATTGGAATGGCT
AACTTAATATTACAAATTGGAAACATAATCTCAATATGGATTAGCCACTCAATTCAACTT
GGGAATCAAAATCAGATTGAAACATGCAATCAAAGCGTCATTACTTATGAAAACAACACT
TGGGTAAATCAGACATATGTTAACATCAGCAACACCAACTTTGCTGCTGGACAGTCAGTG
GTTTCCGTGAAATTAGCGGGCAATTCCTCTCTCTGCCCTGTTAGTGGATGGGCTATATAC
AGTAAAGACAACAGTATAAGAATCGGTTCCAAGGGGGATGTGTTTGTCATAAGGGAACCA
TTCATATCATGCTCCCCCTTGGAATGCAGAACCTTCTTCTTGACTCAAGGGGCCTTGCTA
AATGACAAACATTCCAATGGAACCATTAAAGACAGGAGCCCATATCGAACCCTAATGAGC
TGTCCTATTGGTGAAGTTCCCTCTCCATACAACTCAAGATTTGAGTCAGTCGCTTGGTCA
GCAAGTGCTTGTCATGATGGCATCAATTGGCTAACAATTGGAATTTCTGGCCCAGACAAT
GGGGCAGTGGCTGTGTTAAAGTACAACGGCATAATAACAGACACTATCAAGAGTTGGAGA
AACAATATATTGAGAACACAAGAGTCTGAATGTGCATGTGTAAATGGTTCTTGCTTTACT
GTAATGACCGATGGACCAAGTGATGGACAGGCCTCATACAAGATCTTCAGAATAGAAAAG
GGAAAGATAGTCAAATCAGTCGAAATGAATGCCCCTAATTATCACTATGAGGAATGCTCC
TGTTATCCTGATTCTAGTGAAATCACATGTGTGTGCAGGGATAACTGGCATGGCTCGAAT
CGACCGTGGGTGTCTTTCAACCAGAATCTGGAATATCAGATAGGATACATATGCAGTGGG
ATTTTCGGAGACAATCCACGCCCTAATGATAAGACAGGCAGTTGTGGTCCAGTATCGTCT
AATGGAGCAAATGGAGTAAAAGGATTTTCATTCAAATACGGCAATGGTGTTTGGATAGGG
AGAACTAAAAGCATTAGTTCAAGAAACGGTTTTGAGATGATTTGGGATCCGAACGGATGG
ACTGGGACAGACAATAACTTCTCAATAAAGCAAGATATCGTAGGAATAAATGAGTGGTCA
GGATATAGCGGGAGTTTTGTTCAGCATCCAGAACTAACAGGGCTGGATTGTATAAGACCT
TGCTTCTGGGTTGAACTAATCAGAGGGCGACCCAAAGAGAACACAATCTGGACTAGCGGG
AGCAGCATATCCTTTTGTGGTGTAAACAGTGACACTGTGGGTTGGTCTTGGCCAGACGGT
GCTGAGTTGCCATTTACCATTGACAAGTAA
```

Figure 1.4 Structure primaire de protéine HA du virus Influenza de type A

1.2.5 Sous-séquences nucléotidiques (k-mers)

Un k-mer est un motif défini comme une sous-séquence superposée d'une séquence s avec une longueur de k nucléotides. Par exemple, soit la séquence s définie par la chaîne de caractères : "ACGTTGCA". Alors, le motif m définit par la chaîne de caractères : "TGCA" est un k-mer de longueur $k = 4 \in s$. Un k-mer peut aussi être *dégénéré*. C'est-à-dire qu'un ou plusieurs caractères de sa chaîne de caractères peuvent prendre plusieurs formes. Par exemple, un k-mer définit par :

"ACT{A,G}", implique deux k-mers possibles qui sont "ACTA" et "ACTG". Le motif "ACT{A,G}" sera représenté dans la structure primaire sous la forme "ACTR" où R = {A,G}".

1.3 Processus de séquençage

En bio-informatique, le séquençage désigne le processus qui permet de déterminer la structure primaire d'une biomolécule (molécule présente dans un organisme vivant et qui participe à son métabolisme ainsi qu'à son entretien), c'est-à-dire l'ordre linéaire du début jusqu'à la fin des composants qui la constitue (les nucléotides pour les acides nucléiques et les acides aminés pour les protéines).

1.4 Alignement de séquences

L'alignement de séquences est une méthode de représentation d'un ensemble de séquences. Dans cette représentation, les séquences sous la forme de leur structure primaire sont superposées les une au-dessus des autres en fonction de leurs régions homologues (Figure : 1.5). Les alignements sont réalisés par des programmes informatiques dont l'objectif est de maximiser les zones de concordances des composants (nucléotides ou acides aminés) des différentes séquences. Afin d'aligner au mieux les séquences, les programmes d'alignement peuvent insérer des espacements appelés "*gaps*". Ces derniers correspondent à des insertions ou des délétions de nucléotides ou d'acides aminés au sein des séquences biologiques.

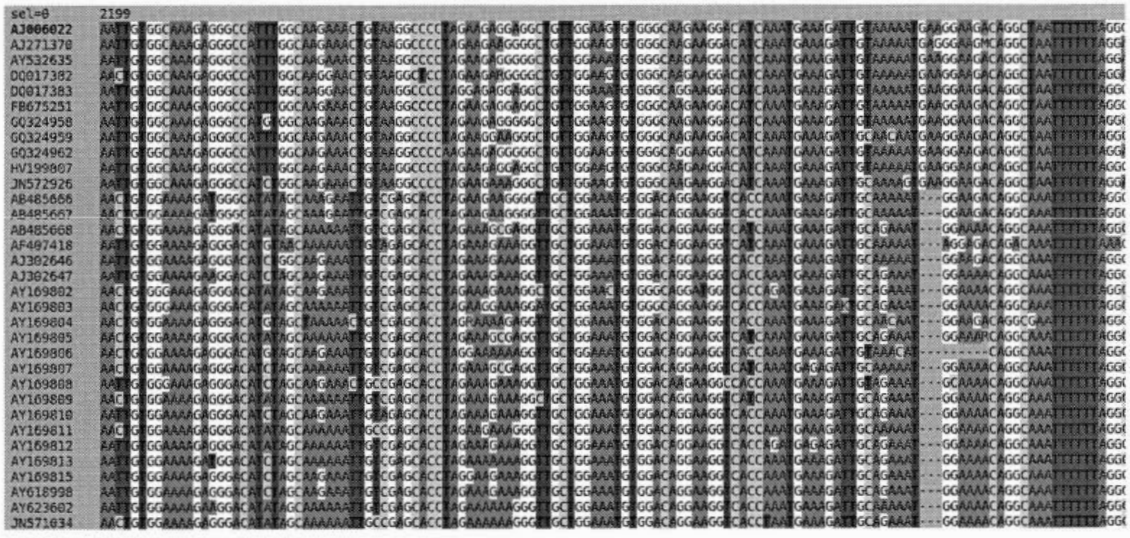

Figure 1.5 Exemple d'alignement de séquences

La figure illustre une partie d'un alignement de plusieurs séquences du virus de l'immunodéficience humaine de type 1 (VIH-1). Cette alignement a été réalisé avec l'algorithme Clustal-W (Thompson et al., *1994) à l'aide de l'application Seaview (Gouy* et al., *2009).*

1.5 Virus biologiques

Les virus sont de petits agents biologiques (diamètre se mesurant à l'échelle du nanomètre) qui infectent les cellules d'autres organismes et exploitent leur machinerie moléculaire afin de pouvoir se répliquer. Tous les virus ont pour structure de base un acide nucléique (ARN ou ADN) inclut dans une capside protéique. Malgré cette base commune, plusieurs schémas de classification ont été proposés afin de les différencier (Mahmoudabadi et Phillips, 2018) (Figure : 1.6). Une première méthode a été réalisée par David Baltimore (lauréat du prix Nobel de physiologie ou médecine en 1975). Celle-ci sépare les virus en sept grands groupes en se basant sur le mode de production de l'ARNm viral (Baltimore, 1971).

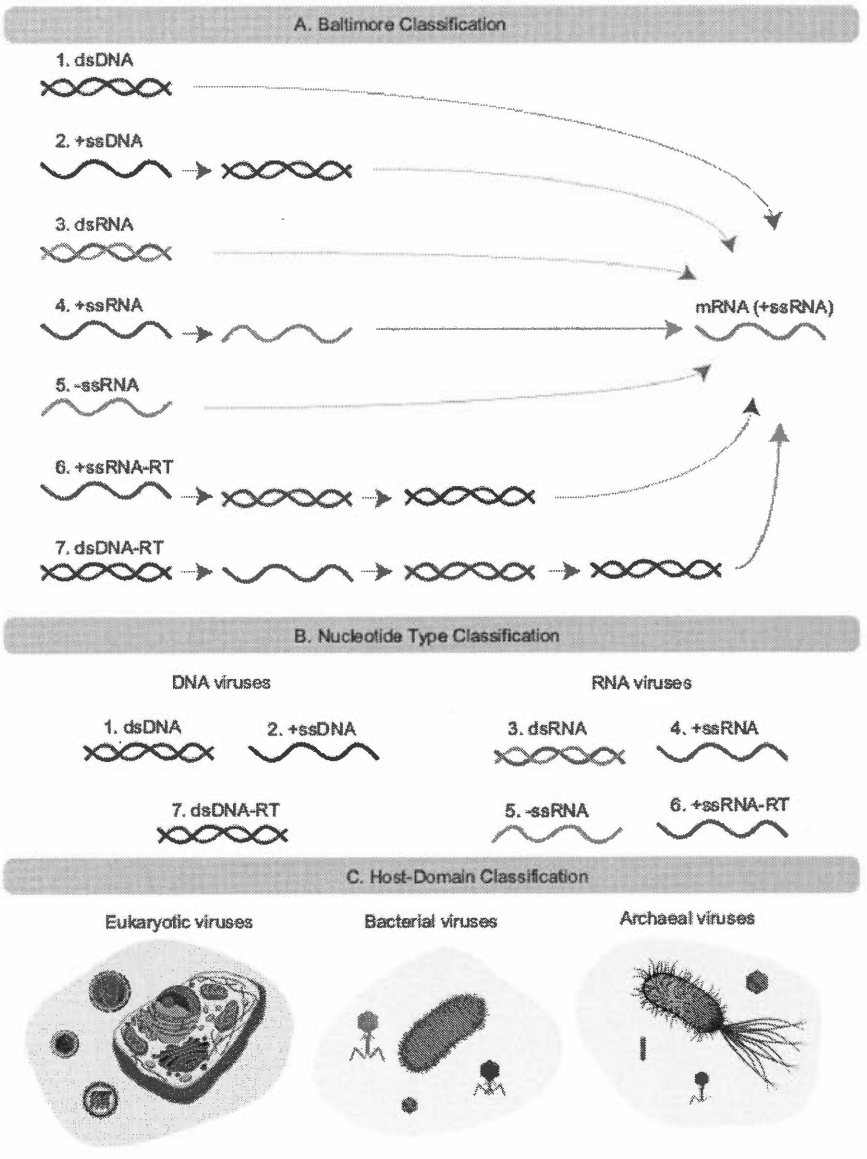

Figure 1.6 Divers systèmes de classification virale. Source : (Mahmoudabadi et Phillips, 2018)

Figure représentant divers systèmes de classification des virus. Légende : DNA = ADN, RNA = ARN, m = messager, ds = double brin, ss = simple brin et RT = transcriptase inverse (enzyme qui transcrit l'information génétique des virus de l'ARN en ADN).

Une deuxième approche basée sur les nucléotides, divise les virus en fonction de leur matériel génomique en virus à ADN et à ARN. Un troisième type de classification regroupe quant à lui les virus en fonction du domaine de l'hôte qu'ils infectent. Trois groupes caractérisent ce type de classification : les virus eucaryotes, bactériens et archéologiques. Même si les spécificités propres à chaque virus qui permettent de le classifier en plusieurs groupes, il existe un consensus concernant leur cycle de réplication :

1. L'attachement : Des protéines du virus effectuent une liaison avec des récepteurs spécifiques se situant sur la membrane cellulaire de la cellule hôte.
2. La pénétration : Mécanisme de pénétration du virus à l'intérieur de la cellule hôte.
3. La décapsidation : Libération du génome viral enveloppé dans une capside protéique.
4. La réplication : Réplication du génome viral et production des protéines virales.
5. L'assemblage : Assemblage et maturation des virus au sein des cellules infectées.
6. La libération : Les nouveaux virions formés sont libérés à l'extérieur de la cellule.

1.6 Principales modifications du génome viral

L'un des facteurs principaux faisant de l'étude des virus une tâche complexe, est la grande instabilité de leur génomes. En effet, ces derniers sont souvent exposés à de nombreuses variations génétiques (Duffy *et al.*, 2008).

1.6.1 Mutations

De par leur rythme de réplication rapide impliquant un grand nombre de particules virales produites depuis les cellules infectées, les virus présentent un plus grand potentiel que les organismes cellulaires à générer des mutations dans un court laps de temps. Ces mutations peuvent être une substitution, dans la situation où une base est remplacée par une autre, une insertion dans le cas où un nucléotide est ajouté dans la séquence, ou encore une délétion, lorsqu'un nucléotide est supprimé de la séquence (Figure : 1.7). Ces types de mutations peuvent parfois affecter les phénotypes des virus (augmentation de la létalité, adaptation à la température, résistances aux antiviraux, accroissement des chances de mutation, etc.).

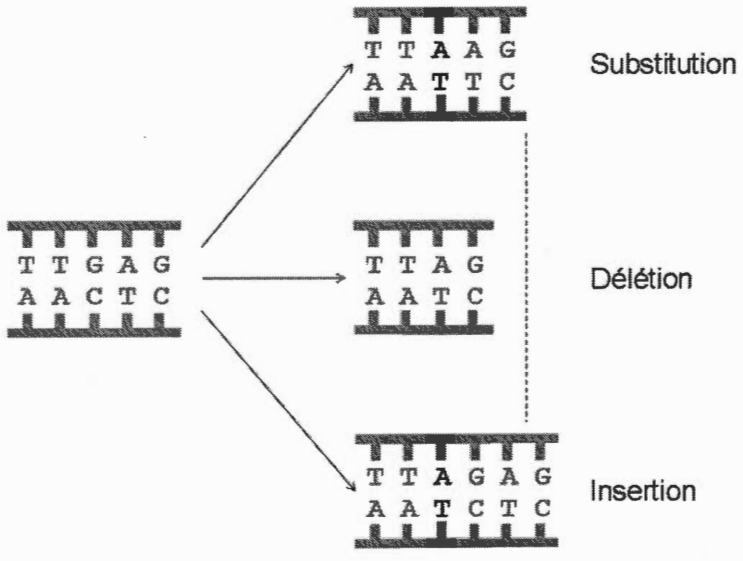

Figure 1.7 Principaux types de mutations génétiques

1.6.2 Recombinaison génétique

Dans le cas où plusieurs particules virales infectent une même cellule hôte, des échanges génétiques entre virus peuvent avoir lieu. Les génomes des différents virus peuvent se rencontrer et éventuellement se fusionner (Figure : 1.8). Ce processus, appelé recombinaison, implique la cassure d'un brin d'acide nucléique, l'échange d'acide nucléique entre deux génomes viraux et la religation de la cassure du brin d'acide nucléique.

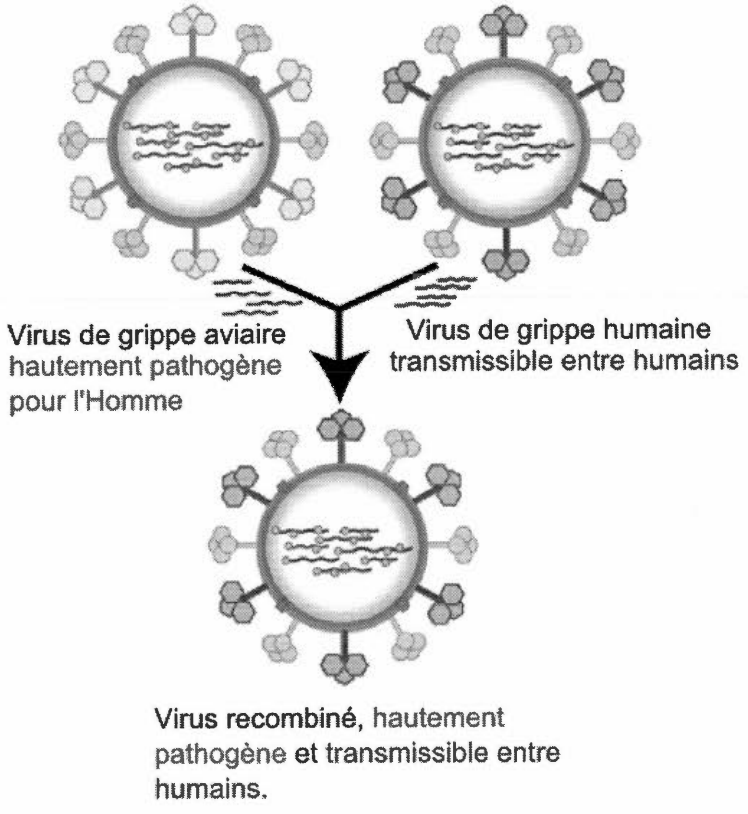

Figure 1.8 Illustration d'une recombinaison génétique chez le virus Influenza. Source : `https://fr.m.wikipedia.org/wiki/Recombinaison_virale`

CHAPITRE II

CONCEPTS D'APPRENTISSAGE AUTOMATIQUE

2.1 Apprentissage automatique en *bio-informatique*

L'apprentissage automatique (*Machine Learning*) est un domaine de recherche à l'intersection des statistiques, de l'intelligence artificielle et de l'informatique qui consiste à extraire des connaissances depuis des données (Müller *et al.*, 2016). Actuellement, l'utilisation des méthodes d'apprentissage automatique est devenue omniprésente dans la vie quotidienne, en passant par les systèmes de recommandation pour le commerce, les systèmes d'aide décisionnelle des entreprises, les outils de détection de fraude ou encore les applications pour l'identification de maladies. Depuis quelques années, avec la forte croissance des données biologiques disponibles, l'apprentissage machine a suscité de nombreux intérêts en *bio-informatique* dans des domaines tels que la génomique, la protéomique, l'étude de l'évolution, ou encore l'analyse des données issues du séquençage (Figure : 2.1).L'apprentissage automatique peut être divisé en deux catégories qui sont : l'apprentissage supervisé et l'apprentissage non supervisé (nous ne traiterons pas de l'apprentissage par renforcement dans ce mémoire). L'apprentissage supervisé se scinde en deux grands types de problèmes, appelés classification et régression.

Dans la classification, l'objectif est de prédire une étiquette de classe. Cette étiquette est un choix parmi une liste prédéfinie de possibilités. Par exemple, si nous

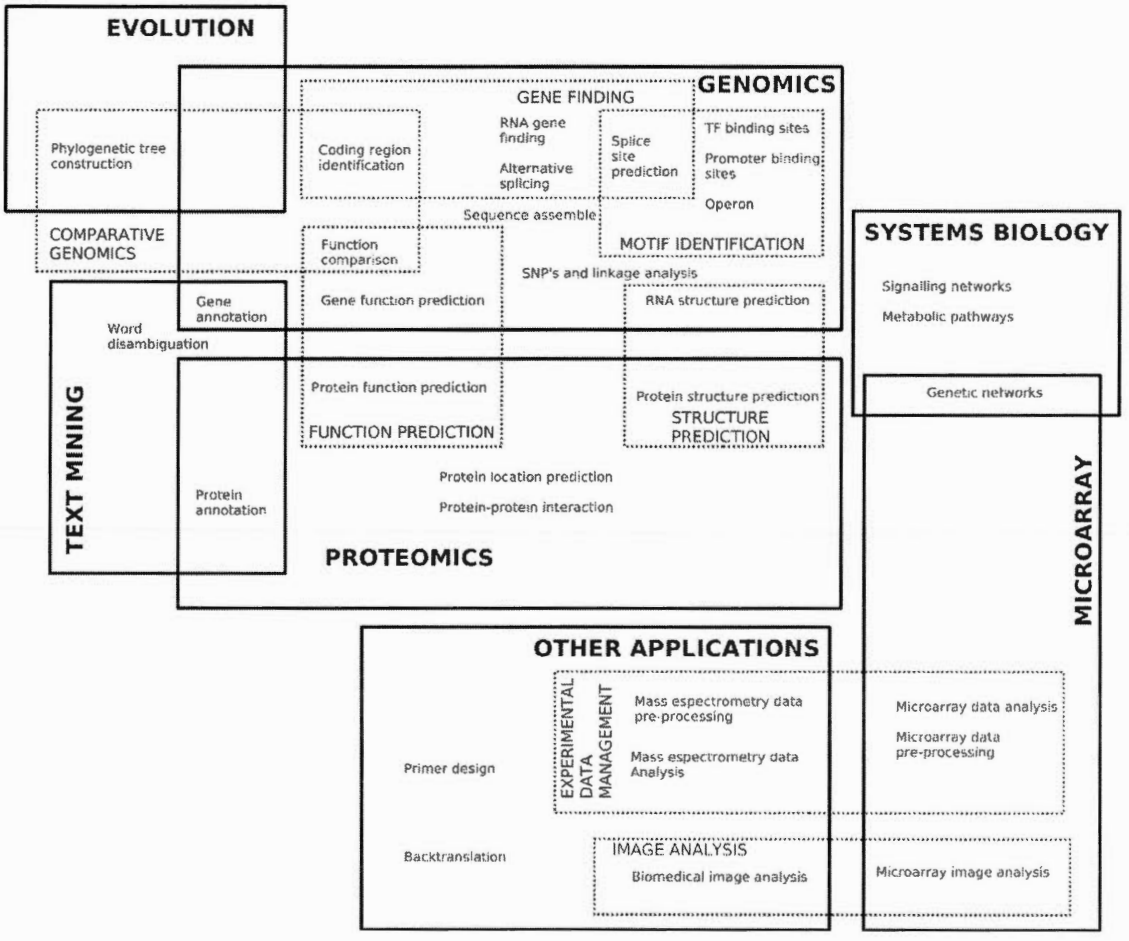

Figure 2.1 Domaines d'application de l'apprentissage automatique en *bio-informatique*. Source : (Larranaga *et al.*, 2006)

cherchons à prédire le groupe d'une instance de VIH-1, les étiquettes prédéfinies possibles seront : Groupe O, Groupe M, Groupe N ou Groupe P.

Concernant les tâches de régression, ici l'objectif est de prédire une valeur numérique qui peut être un nombre continu ou bien à virgule. Dans ce cas, il n'y a pas de liste d'étiquettes prédéfinies. Par exemple, nous pouvons chercher à prédire la probabilité qu'une mutation ait lieu sur le nucléotide n d'une séquence s.

Pour l'apprentissage non supervisé, nous pouvons distinguer des techniques de transformation des données et des méthodes de regroupement (*clustering*). Les techniques de transformation des données ont pour objectif de créer une nouvelle représentation des données qui facilite la compréhension pour les humains ou d'autres algorithmes d'apprentissage machine. Une application courante des transformations non supervisées est la réduction de la dimensionnalité. Elle prend en entrée des données sous une représentation à grande dimension, composée de nombreux attributs, afin de trouver une nouvelle manière de les représenter, qui résument les attributs essentiels avec moins de composants. Une application populaire est l'analyse des composantes principales (Jolliffe, 2011) qui est très utilisée pour visualiser les données représentées par de nombreux attributs, à travers seulement deux ou trois dimensions.

Enfin, les algorithmes de *clustering* cherchent à partitionner les données en groupes distincts d'éléments similaires appelés *clusters*. Par exemple, si nous appliquons un algorithme de *clustering* sur notre cas précédent du VIH-1, il y a de forte probabilité que quatre *clusters* se forment et correspondent respectivement aux classes de virus O, M, N et P. Ce type de méthode peut aussi être utilisé pour la détection de nouvelles classes (dans notre exemple, par la formation d'un cinquième *cluster* contenant plusieurs instances) ou encore pour la détection de valeurs aberrantes (appelées *outliers*) (dans notre exemple, il s'agirait d'une valeur isolée et éloignée

de tous les autres *clusters*).

2.2 Pré-traitement et mise à l'échelle des données

En apprentissage machine, certains algorithmes, tels que les machines à vecteurs de support (SVM), sont très sensibles à l'échelle des données (Hsu *et al.*, 2003). Par conséquent, une pratique courante consiste à ajuster les attributs de façon à ce que la représentation des données soit plus appropriée pour l'application de ces algorithmes. De plus, comme démontré dans (van den Berg *et al.*, 2006), ces méthodes de pré-traitements présentent des impacts positifs pour l'exploitation des données biologiques à travers les outils de la *bio-informatique*. La plupart du temps, il s'agit d'un simple changement d'échelle des valeurs des attributs et d'un décalage des données. La figure 2.2 illustre les effets de différentes méthodes de pré-traitement et de mise à l'échelle sur un jeu de données synthétique.

La première méthode, *StandardScaler* (Figure : 2.2.b), fait en sorte que pour l'ensemble des valeurs des attributs, la moyenne soit de 0 et la variance égale à 1. Cette méthode permet de centrer les valeurs sur une même échelle, mais ne garantit pas de valeurs minimales et maximales particulières.

La deuxième technique présentée est le *MinMaxScaler* (Figure : 2.2.c), qui décale les données de telle sorte que toutes les valeurs des attributs se situent exactement entre une borne minimum (min) et une borne maximum (max) données en entrée. Cela signifie que toutes les données sont contenues dans le rectangle créé par l'axe des x entre x_{min} et x_{max} et l'axe des y entre y_{min} et y_{max}.

La troisième fonction de pré-traitement illustrée est le *RobustScaler* (Figure : 2.2.d) . Ce dernier fonctionne de la même manière que le *StandardScaler*, dans le sens où il assure des propriétés statistiques qui garantissent que les valeurs des attributs soient à la même échelle. Cependant, le *RobustScaler* utilise la médiane

et les quartiles au lieu de la moyenne et de la variance. Les conséquences sont que le *RobustScaler* ignore les points de données qui sont très différents des autres (comme les erreurs de mesure). Ces points de données également appelés valeurs aberrantes (ou *outliers*) peuvent causer des problèmes pour d'autres techniques de mise à l'échelle.

Enfin, la dernière méthode utilisée dans le graphique (Figure : 2.2.e) est le *Normalizer* qui effectue un type de redimensionnement différent des autres fonctions de pré-traitement. Il met à l'échelle chaque point de données de façon à ce que le vecteur des attributs ait une longueur euclidienne égale à 1. En d'autres termes, il projette un point de données sur un cercle (ou une sphère, dans le cas de dimensions supérieures) avec un rayon de 1. Cette normalisation est souvent utilisée lorsque seule la direction des données compte et non la longueur du vecteur des attributs.

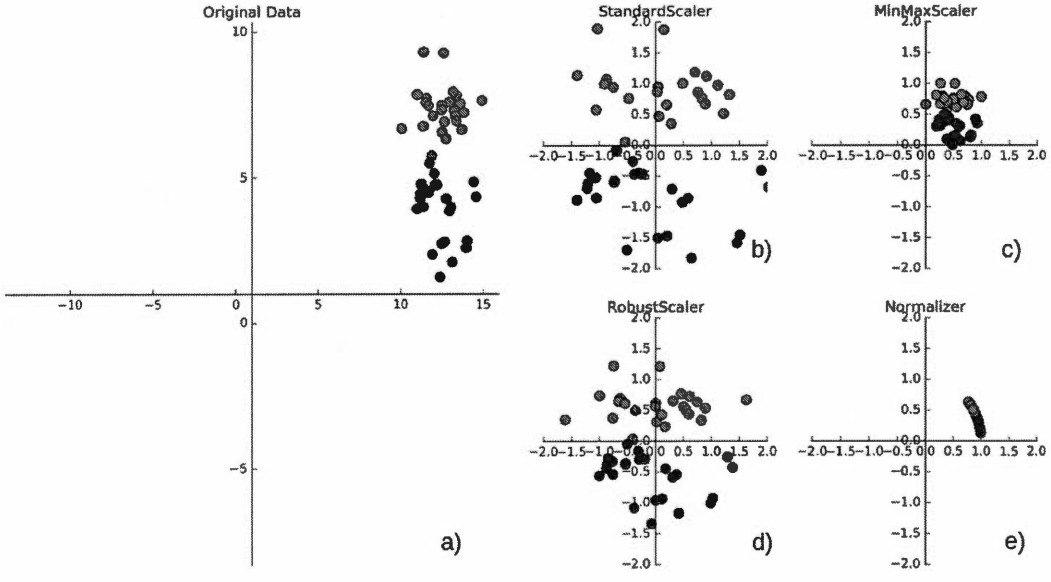

Figure 2.2 Différentes techniques de pré-traitement et de redimensionnement d'un ensemble de données. Source : (Müller *et al.*, 2016)

La figure illustre à gauche un premier graphique d'un ensemble de données synthétiques de classification. Ce dernier est composé de deux classes (rouge et bleu) incluant deux attributs représentés par l'axe des ordonnées et des abscisses. Les quatre graphiques à droite représentent différentes manières de transformer les données sur des échelles plus standards à l'aide d'outils de la librairie Scikit-learn (Pedregosa et al., *2011).*

2.3 Méthodes de sélection d'attributs

L'augmentation constante des données à traiter a rendu les méthodes de sélection d'attributs pratiquement indispensables en tant que stratégie de pré-traitement pour les problèmes de fouille de données et d'apprentissage machine. Ces méthodes de sélection d'attributs ont pour objectifs la construction de modèles plus simples et plus compréhensibles, l'amélioration des performances durant la phase d'exploration de données ou d'apprentissage machine ainsi que la préparation de données claires et compréhensibles. De manière générale (Li *et al.*, 2017), ou en *bio-informatique* (Saeys *et al.*, 2007), les méthodes de sélection d'attributs sont divisées en trois grandes catégories :

1. Méthodes de type *Filter*
2. Méthodes de type *Wrapper*
3. Méthodes de type *Embedded*

2.3.1 Méthodes de type *Filter*

Les techniques de type *Filter* évaluent la pertinence des attributs en examinant seulement les propriétés intrinsèques des données. Dans la majorité des cas, un score de pertinence des attributs est calculé et ceux avec les scores les plus faibles

sont supprimés. Les avantages de ce type de techniques est qu'elles sont simples et rapides à calculer, et qu'elles s'adaptent à de grandes dimensions. Cependant, l'inconvénient courant des méthodes de type *Filter* est qu'elles ignorent l'interaction avec l'algorithme de classification et que la plupart des techniques proposées sont univariées telles que `chi2` (Liu et Setiono, 1995), `ANOVA` (Jafari et Azuaje, 2006) ou encore t-`test` (Jafari et Azuaje, 2006). Malgré cela, ces dernières sont parmi les approches les plus utilisées dans les études sur les micro-puces à ADN. Une méthode univariée signifie qu'elle considère chaque attribut séparément, ignorant ainsi les dépendances entre ces derniers, ce qui peut conduire à des performances inférieures par rapport à d'autres types de techniques. Afin de surmonter le problème de l'ignorance des dépendances des attributs, un certain nombre de techniques de type *Filter* multivariées ont été introduites telles que *Markov blanket filter* (Koller et Sahami, 1996), *Correlation-based feature selection* (Hall, 1999) ou encore *Bivariate* (Bø et Jonassen, 2002), visant à incorporer dans une certaine mesure les dépendances entre les attributs.

2.3.2 Méthodes de type *Wrapper*

Le deuxième type d'approche (*Wrapper*) intègre dans sa recherche de sous-ensemble d'attributs l'hypothèse d'un modèle. Il effectue une procédure de recherche dans l'espace des sous-ensembles d'attributs possibles afin de générer et d'évaluer divers sous-ensembles d'attributs. L'évaluation d'un sous-ensemble spécifique d'attributs est obtenu par l'entrainement et le test d'un modèle de classification, rendant ainsi cette approche adaptée à un algorithme de classification spécifique. Pour rechercher l'espace de tous les sous-ensembles d'attributs, un algorithme de recherche est "enroulé" autour du modèle de classification. Cependant, comme l'espace des sous-ensembles d'attributs croît de façon exponentielle dû au nombre de ces derniers, des méthodes de recherche heuristiques sont utilisées pour guider la recherche d'un

sous-ensemble optimal. Parmi les approches de type *Wrapper*, nous pouvons citer des algorithmes génétiques (Holland, 1975), *Randomized hill climbing* (Skalak, 1994) ou encore des algorithmes d'estimation de distribution (Blanco *et al.*, 2004). Les avantages des approches *Wrapper* est qu'elles incluent l'interaction entre la recherche de sous-ensembles d'attributs et la sélection de modèles, ainsi que la possibilité de prendre en compte les dépendances entre les attributs. L'inconvénient courant est qu'elles présentent un risque plus important de surapprentissage (ou *overfitting*) que les techniques de filtrage et qu'elles impliquent parfois un coût de calcul élevé.

2.3.3 Méthodes de type *Embedded*

Le dernier grand groupe de techniques de sélection d'attributs est nommé *Embedded.* Avec ce type d'approche, la recherche d'un sous-ensemble optimal d'attributs est intégrée à la construction du modèle de classification. Tout comme les approches *Wrapper*, les approches *Embedded* sont donc spécifiques à un algorithme d'apprentissage. Les approches *Embedded* telles que les méthodes de sélection d'attributs utilisant les vecteurs de pondération de *SVM* (Guyon *et al.*, 2002; Jong *et al.*, 2004) ou encore celles basées sur le gain d'information des attributs établi par *Random Forest* (Díaz-Uriarte et De Andres, 2006), ont l'avantage d'inclure l'interaction avec le modèle de classification. Elles offrent donc les performances des méthodes de type *Wrapper*, tout en étant beaucoup moins intensives en terme de calcul. En effet, ces dernières n'ont pas besoin d'évaluer les ensembles de caractéristiques de façon itérative.

2.4 Algorithmes d'apprentissage supervisé

Dans le domaine supervisé de l'apprentissage automatique, il existe de nombreux algorithmes capables d'apprendre sur des données afin de réaliser par la suite des prédictions. À travers cette section, plusieurs approches supervisées ainsi que quelques-uns de leurs avantages et inconvénients seront présentés.

2.4.1 Méthode des k plus proches voisins

Un premier type d'algorithme et sûrement l'un des plus simples est celui des k plus proches voisins (k-NN ou KNN), de l'anglais *k-nearest neighbor* (Cover et Hart, 1967). Dans un cadre de classification, k-NN dispose d'une base de données d'apprentissage constituée de n instances accompagnées de leurs classes associées. Afin d'estimer la classe d'appartenance d'un nouvel objet x, la méthode des k plus proches voisins va se baser de façon identique sur les k échantillons d'apprentissage les plus proches de la nouvelle entrée x, selon une métrique distance telle que la distance euclidienne (2.1) ou de Minkowski (2.2).

$$Distance\ euclidienne = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.1}$$

$$Distance\ de\ Minkowski = \left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{1/p} \tag{2.2}$$

En résumé, un nouvel objet x est classifié selon le résultat majoritaire des statistiques de classes d'appartenances de ses k plus proches voisins. Par exemple, si $k = 1$, alors x est assigné à la classe d'appartenance de son plus proche voisin. De manière générale, ce type d'algorithme possède une faculté d'apprentissage rapide et de bonnes performances de classification. Cependant, de par son fonctionne-

ment, k-NN reste très sensible aux ensembles de données contenants des attributs non pertinents ou interdépendants (Kotsiantis *et al.*, 2007).

2.4.2 Arbre de décision

Un deuxième type d'algorithme communément utilisé est l'arbre de décision (Quinlan, 2014). L'apprentissage par arbre de décision repose sur l'utilisation d'un arbre de décision comme modèle prédictif. Dans ce type de structure, chaque nœud interne correspond à un attribut, chaque arête menant à un nœud fils représente un ensemble de valeurs qu'un attribut peut prendre et chaque feuille (nœud terminal de l'arbre) contient une classe (Figure : 2.3).

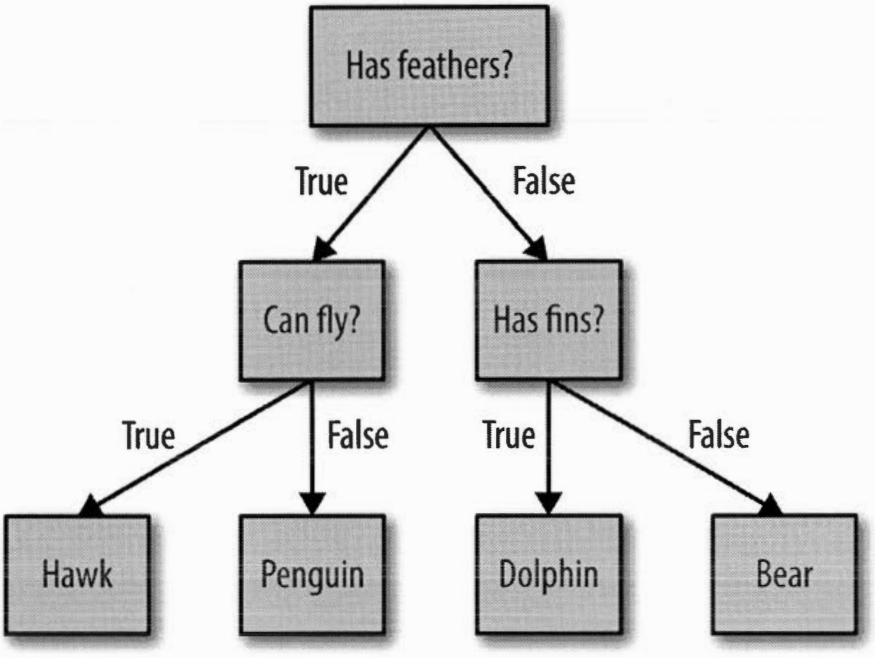

Figure 2.3 Exemple d'arbre de décision. Source : (Müller *et al.*, 2016)

La figure représente un arbre de décision formé à partir de trois attributs binaires menant à quatre classes potentielles.

Les arbres de décision classent les instances inconnues en les triant en fonction des valeurs de leurs attributs. L'architecture de ces derniers est établie à travers un classement et une sélection des meilleurs attributs, définis par des mesures comme l'entropie (Hunt *et al.*, 1966) (2.3) et l'indice de Gini (Breiman, 2017) (2.4).

Soit un nœud n et $p(j|n)$ la fréquence relative de la classe j au nœud n.

$$Entropie(n) = -\sum_{j} p(j|n) \log p(j|n) \tag{2.3}$$

$$Gini(n) = 1 - \sum_{j} p(j|n)^2 \tag{2.4}$$

Il résulte de ce type d'algorithme une capacité d'apprentissage et de prédiction rapide avec de manière générale de bonnes performances de prédiction. Un avantage supplémentaire se trouve dans la facilité d'interprétation de leur décision (Kotsiantis *et al.*, 2007). Cependant, les arbres de décision s'exposent facilement à des risques de surapprentissage (Müller *et al.*, 2016).

2.4.3 Forêts d'arbres décisionnels

L'algorithme des forêts d'arbres décisionnels (ou forêts aléatoires de l'anglais *Random forest classifier*) a été introduit par Breiman (2001). *Random forest* fait partie des méthodes ensemblistes les plus performantes qui ont pour fonction de combiner plusieurs modèles d'apprentissage machine afin de créer des modèles plus puissants. Concrètement, une forêt aléatoire se constitue d'une collection d'arbres de décision, où chaque arbre est légèrement différent des autres (Figure : 2.4). L'idée principale est que chaque arbre de manière individuelle peut fournir un travail correct de prédiction, mais tel qu'évoqué précédemment, il serait trop adapté à une partie des données.

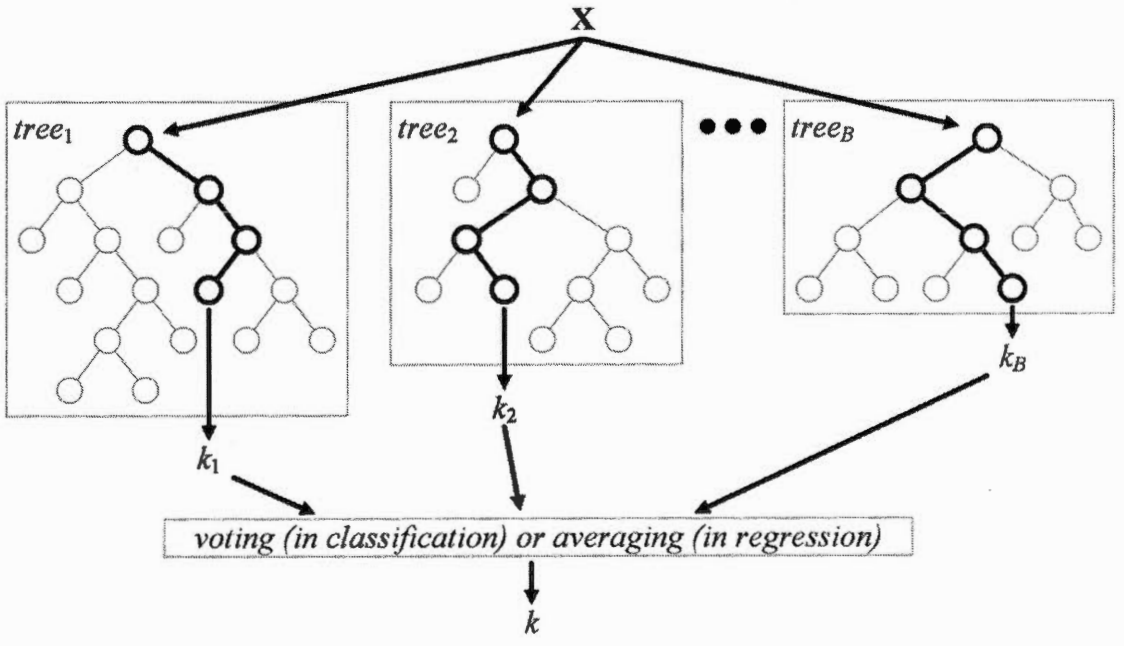

Figure 2.4 Architecture d'un modèle de forêt d'arbres décisionnels. Source : (Verikas *et al.*, 2016)

En construisant beaucoup d'arbres qui fonctionnent et s'adaptent de différentes façons, il est donc possible, en faisant la moyenne de leurs résultats, d'obtenir un modèle plus performant moins sujet au surapprentissage. La principale contrainte des forêts d'arbres décisionnels est qu'ils perdent la facilité d'interprétation de la prédiction de l'arbre de décision unique (Müller *et al.*, 2016).

2.4.4 Classifieurs bayésiens

Parmi les algorithmes de classification supervisée populaires, existent aussi les classifieurs bayésiens (Friedman *et al.*, 1997). Un classifieur naïf de Bayes est un algorithme probabiliste basé sur l'application du théorème de Bayes (2.5) avec une hypothèse naïve, c'est-à-dire que les variables explicatives Xi sont supposées

indépendantes conditionnellement à la classe cible C.

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \tag{2.5}$$

où $P(A|B)$ désigne la probabilité conditionnelle de A sachant B.

Les grandes étapes peuvent se résumer de la manière suivante. On détermine un ensemble d'apprentissage. On calcule les probabilités a priori de chaque classe (en comptant par exemple leurs proportions relatives dans l'ensemble d'apprentissage). On applique ensuite la règle de Bayes pour obtenir la probabilité à posteriori des classes en fonction des valeurs des attributs. Puis, on choisit la classe la plus probable. Ce type d'algorithme présente l'avantage d'être très rapide en terme d'apprentissage et de classification, tout en offrant une bonne tolérance au bruit et au risque de surapprentissage. Cependant, leurs performances restent inférieures par rapport à des approches telles que les *SVM*, les réseaux de neurones ou encore les *Random Forest*(Kotsiantis *et al.*, 2007). Cela peut s'expliquer par l'aspect "naïf" qui ne tient pas compte des dépendances entres les attributs.

2.4.5 Machine à vecteurs de support

Les machines à vecteurs de support ou séparateurs à vaste marge, de l'anglais *support vector machine* (SVM), représentent un ensemble de techniques d'apprentissage supervisé ayant pour objectif de résoudre des problèmes de discrimination (déterminer la classe d'appartenance d'un objet) et de régression (prédire la valeur numérique d'une variable). Dans un cadre de classification binaire, les SVM cherchent à séparer les instances en deux classes (Figure : 2.5). Pour cela, ils utilisent un hyperplan qui se base sur des instances essentielles nommées vecteurs de support ainsi que sur des marges définies par ces derniers (Boser *et al.*, 1992).

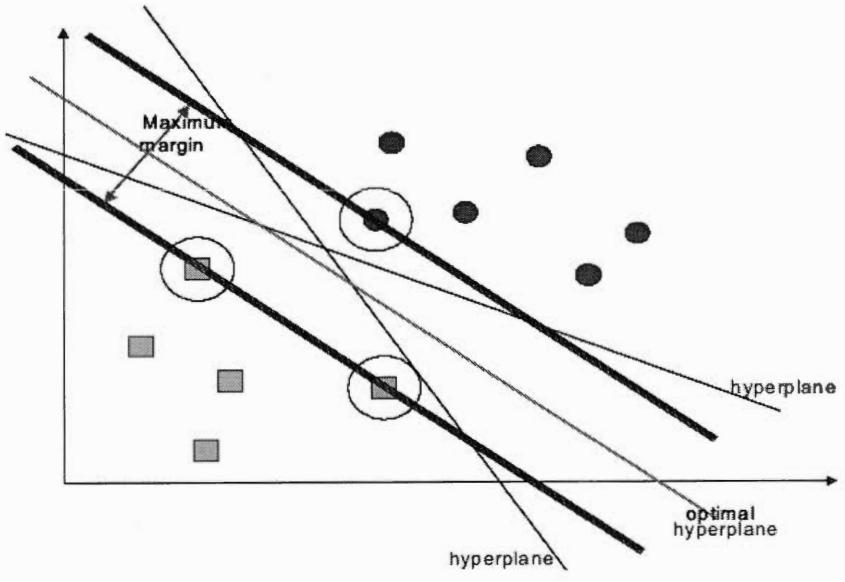

Figure 2.5 Exemple de séparation par machine à vecteurs de support. Source : (Kotsiantis *et al.*, 2007)

La figure représente une séparation par une approche SVM. Dans cette figure, les instances de deux classes sont séparées par un hyperplan optimal qui maximise la marge de séparation.

Cependant, il existe une infinité d'hyperplans séparateurs qui ont un taux d'erreur de classification nul sur des instances d'entraînement. Néanmoins, ces hyperplans ne possèdent pas une performance égale sur des objets inconnus. C'est pour cette raison que les SVM cherchent l'hyperplan avec une marge maximale qui minimise le risque empirique de mauvaises clasifications (Boser *et al.*, 1992), (Vapnik et Chervonenkis, 2015). Les SVM sont parmi les approches permettant d'obtenir les meilleures performances de prédiction. Malgré une phase d'entrainement lente, ils fournissent une capacité à traiter avec des attributs non pertinents et une bonne rapidité de prédiction une fois le modèle établi (Kotsiantis *et al.*, 2007).

2.4.6 Perceptron Multicouche

Le perceptron multicouche, de l'anglais *Multilayer perceptron* (MLP), fait partie des grandes familles des réseaux de neurones. Il est organisé en plusieurs couches au sein desquelles une information circule de la couche d'entrée vers la couche de sortie. Il s'agit d'un réseau à propagation directe (*feedforward*). La première couche est reliée aux entrées. Les couches suivantes sont ensuite reliées aux sorties des couches précédentes. La dernière couche produit les sorties du perceptron multicouche. Les sorties des autres couches ne sont pas visibles à l'extérieur du réseau et sont pour cette raison appelées couches cachées. Chaque neurone N_i de perceptron réalise un produit scalaire entre son vecteur d'entrées X et un vecteur de paramètres W appelé poids. Il y ajoute ensuite un biais b et utilise une fonction d'activation f pour déterminer sa sortie.

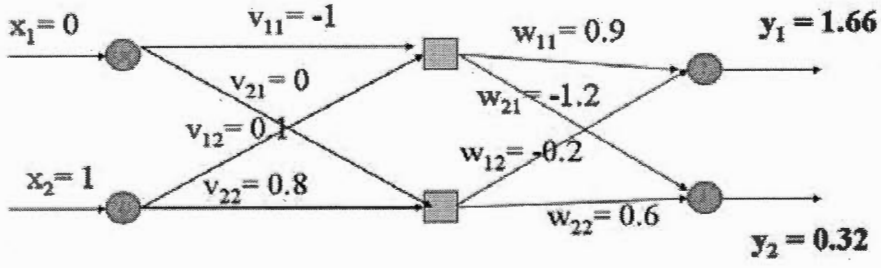

Figure 2.6 Exemple de réseau de neurones. Source : (Kotsiantis *et al.*, 2007)

Les réseaux de neurones font partie des approches permettant d'atteindre les meilleures performances de prédiction. Cependant, il s'avère souvent difficile voir pratiquement impossible d'interpréter leur décision. Ces derniers présentent une forte exposition au surapprentissage (Kotsiantis *et al.*, 2007). De plus, la phase de construction du modèle est très lente et ils sont, contrairement au SVM, très sensibles aux attributs sans importance.

2.5 Évaluation de l'apprentissage

Dans l'objectif de savoir si un modèle de classification est performant, c'est-à-dire s'il est capable de classifier avec un bon score un ensemble de données indépendant des instances d'entrainement, il est nécessaire d'utiliser des méthodes ainsi que des métriques d'évaluation.

2.5.1 Méthodes d'évaluation

Validation (Entraînement / Test) : méthode qui consiste à diviser un ensemble de données initial en deux sous-ensembles disjoints. L'un des sous-ensemble est utilisé pour entrainer un modèle d'apprentissage (*training set*) et l'autre sert de base de test pour évaluer la performance du modèle (*testing set*).

Validation croisée k : méthode statistique d'évaluation des performances plus stable et plus approfondie que la validation (entraînement / test). Elle consiste à diviser un ensemble de données en k partitions disjointes approximativement similaires. Des processus d'entraînement et de test sont réalisés k fois de manière alternative. Pour chaque itération k-1 partitions sont utilisées pour former un modèle d'entrainement et la dernière partition est utilisée comme base de test (Figure : 2.7).

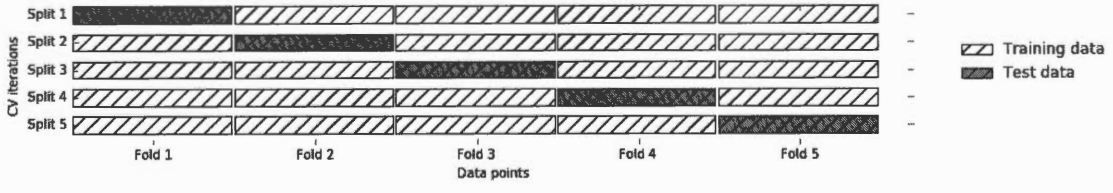

Figure 2.7 Schéma de validation croisée 5. Source : (Müller *et al.*, 2016)

Bootstrap : méthode qui réalise un échantillonnage avec remise à partir d'un ensemble de données initial. Cet échantillonnage est répété n fois. Pour chaque itération i, un bootstrap $boot_i$ est formé sur lequel un modèle est entrainé et un ensemble de test $test_i$ est construit pour évaluer le modèle d'entrainement. Avec ce type d'approche, il est donc possible d'avoir des objets similaires présents à la fois dans l'ensemble d'entrainement et dans l'ensemble de test.

Jackknife : méthode similaire au *Boostrap* à l'exception du fait qu'il n'y a pas de remise effectuée durant les phases d'échantillonnage des ensembles d'entrainement et de test de chaque itération. Avec ce type d'approche, il n'est donc pas possible d'avoir des instances similaires dans l'ensemble entrainement et de test.

2.5.2 Métriques d'évaluation des performances

Afin d'illustrer plusieurs métriques d'évaluation des performances de classification, supposons un ensemble de données divisé en deux classes : une classe True et une classe False étiquetées $+$ et $-$ respectivement. Les différentes prédictions que pourrait effectuer un modèle sur ce type de données seraient :

1. Correctes pour les instances $+$ prédites comme $+$ (Vrais positifs)
2. Correctes pour les instances $-$ prédites comme $-$ (Vrais négatifs)
3. Incorrectes pour les instances $+$ prédites comme $-$ (Faux positifs)
4. Incorrectes pour les instances $-$ prédites comme $+$ (Faux négatifs)

L'ensemble des possibilités de prédiction peuvent être représentées sous forme de matrice de confusion (Tableau : 2.1).

Tableau 2.1 Matrice de confusion pour un cas de classification binaire

		Classes prédites	
		+	−
Classes réelles	+	Vrais positifs (VP)	Faux négatifs (FN)
	−	Faux positifs (FP)	Vrais négatifs (VN)

Depuis cette matrice de confusion, plusieurs métriques de performance peuvent être calculées :

1. **Taux de faux positifs (TFP)** : proportion des exemples négatifs incorrectement classés.

$$TFP = \frac{FP}{VN + FP} \tag{2.6}$$

2. **Taux de faux négatifs (TFN)** : proportion des exemples positifs incorrectement classés.

$$TFN = \frac{FN}{VP + FN} \tag{2.7}$$

3. **Taux de vrais négatifs (TVN)** : proportion des exemples négatifs correctement classifiés, connu aussi sous le nom de *spécificité*.

$$TVN = \frac{VN}{VN + FP} \tag{2.8}$$

4. **Rappel** : proportion des exemples positifs correctement prédits. Il est appelé aussi *Sensibilité* ou taux de vrais positifs (TVP).

$$Rappel = \frac{VP}{VP + FN} \tag{2.9}$$

5. **Précision** : fraction des exemples positifs correctement classifiés par rapport à tous les exemples classés positifs par le modèle.

$$Precision = \frac{VP}{VP + FP} \tag{2.10}$$

6. **Exactitude** : de anglais *Accuracy*, désigne la proportion des prédictions correctes effectuées par le modèle.

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN} \tag{2.11}$$

7. **F-mesure** : moyenne harmonique entre le Rappel et la Précision.

$$F - mesure = 2 * \frac{Precision * Rappel}{Precision + Rappel} \tag{2.12}$$

CHAPITRE III

ÉTAT DE L'ART ET PROBLÉMATIQUE

3.1 Méthodes de classification basées sur l'alignement

En bio-informatique, dans le domaine de la classification des séquences génomiques, plusieurs méthodes ont déjà été mises en application. Celles-ci peuvent être divisées en deux catégories distinctes que sont les méthodes basées sur l'alignement (Edgar et Batzoglou, 2006) et les méthodes indépendantes de l'alignement (Vinga et Almeida, 2003).

Parmi les méthodes basées sur l'alignement, nous pouvons citer dans un premier temps les outils généralistes pour la recherche de similarité tels que BLAST (Altschul *et al.*, 1997) et USEARCH (Edgar, 2010). Dans un deuxième temps, nous pouvons mentionner les méthodes par paires basées sur le calcul de distance tel que PASC (Bao *et al.*, 2014) ou encore DEmARC (Lauber et Gorbalenya, 2012). D'autres types d'approches telle que Pplacer (Matsen *et al.*, 2010), utilisent quant à elles des arbres phylogénétiques construits à partir d'ensembles de séquences connues. L'objectif est d'insérer les séquences encore inconnues dans un arbre, en les alignant avec les séquences de ce dernier. Toutefois, ce type d'approches basées sur la phylogénie reste souvent très spécialisé à certaines espèces. Ce qui est le cas de REGA (De Oliveira *et al.*, 2005; Alcantara *et al.*, 2009) et SCUEAL (Pond *et al.*, 2009), spécifique au virus de l'immunodéficience humaine de type

1 (VIH-1). À notre connaissance, la méthode la plus récente et preformante proposant une réponse partielle à notre problématique (Section : 3.5) est MISSEL (Multiple SubSequences Extractor for cLassification) (Fiscon *et al.*, 2016). MISSEL est une approche supervisée dépendante de l'alignement basée sur un algorithme génétique. Elle permet d'extraire des sous-séquences discriminantes au sein d'ensembles de séquences connues afin de construire des modèles de prédiction facilement interprétables à l'échelle humaine pour classifier des nouvelles séquences encore inconnues. Leur algorithme a été évalué sur plusieurs jeux de données viraux (Influenza virus, Polyoma virus et Rhino virus) obtenant des résultats de prédiction allant de 90% à 100% d'*exactitude*.

3.2 Limitations des approches basées sur l'alignement

Cependant, ces méthodes initiales s'appuyant sur l'alignement font face à de nombreuses contraintes. Un des premiers obstacles de ce type de méthodes est que leur complexité algorithmique implique des temps et coût de calcul qui peuvent se révéler insurmontable sur certains groupes de génomes complets (Bonham-Carter *et al.*, 2013). Trouver l'alignement multiple optimal entre n séquences a été classé parmis les problèmes NP-difficile. En effet, pour aligner n séquences de longueur l, il en résulte une complexité algorithmique $O(l^n)$. Ce qui devient inapplicable dès que $n > 5$ et $l \approx 100$. Deuxièmement, les algorithmes d'alignement supposent que les séquences homologues se composent d'une série de segments de séquence agencés de façon linéaire et plus ou moins conservés. Or, cette hypothèse, nommée *colinéarité*, n'est pas toujours représentative de la réalité (Zielezinski *et al.*, 2017). En effet, dans notre cas portant sur les génomes viraux, ces derniers sont exposés à de grandes variations génétiques en raison de leur taux de mutations élevés, leurs fréquentes recombinaisons génétiques ou encore de leurs transferts horizontaux de gènes (Duffy *et al.*, 2008). Enfin, le fait d'effectuer un alignement nécessite souvent

d'ajuster plusieurs paramètres (matrices de substitution, pénalités d'écart, seuils pour les paramètres statistiques ...) qui sont dépendant de connaissances a priori sur l'évolution des séquences comparées. L'ajustement de ces paramètres est donc parfois arbitraire et nécessite une approche par essais et erreurs (Zielezinski *et al.*, 2017). De plus, de nombreuses expériences ont montré que de petites variations au niveau de ces paramètres peuvent affecter grandement la qualité de l'alignement (Wong *et al.*, 2008). Ces nombreuses limitations nous mènent à nous orienter vers les méthodes dites *Alignment-free*, se détachant de l'alignement.

3.3 Méthodes indépendantes de l'alignement et attributs basés sur les k-mers

Afin de faire face aux nombreux problèmes des méthodes basées sur l'alignement, les méthodes sans alignement sont devenues une alternative dans la comparaison et la classification des séquences. Elles transforment les séquences biologiques en vecteurs d'attributs (Vinga, 2014) pour calculer une distance, construire un modèle phylogénétique ou d'apprentissage machine (Xing *et al.*, 2010) (Figure : 3.1). Les méthodes sans alignement ont été initiées par Blaisdell (1986) qui propose une approche pour comparer des séquences par un calcul de similarité. Les méthodes plus récentes se basent sur l'utilisation des statistiques liées à la composition des séquences et les corrélations nucléotidiques telles que COMET (Struck *et al.*, 2014), Fangorn Forest (F2) (Silva *et al.*, 2017b), (Wen *et al.*, 2014), (Yu *et al.*, 2013) et (Liu *et al.*, 2008). D'autres approches utilisent l'information positionnelle (Yu *et al.*, 2010a), (Deng *et al.*, 2011) ou encore le gain d'information (Li *et al.*, 2001) et (Wang, 2011). Récemment, notre laboratoire de recherche a développé une méthode de classification virale (Remita *et al.*, 2017) utilisant l'apprentissage automatique combiné à des signatures biologiques (Williams, 1989) pour constituer des attributs. Afin d'obtenir une classification précise en utilisant une approche sans alignement, il est en effet essentiel de pouvoir identifier

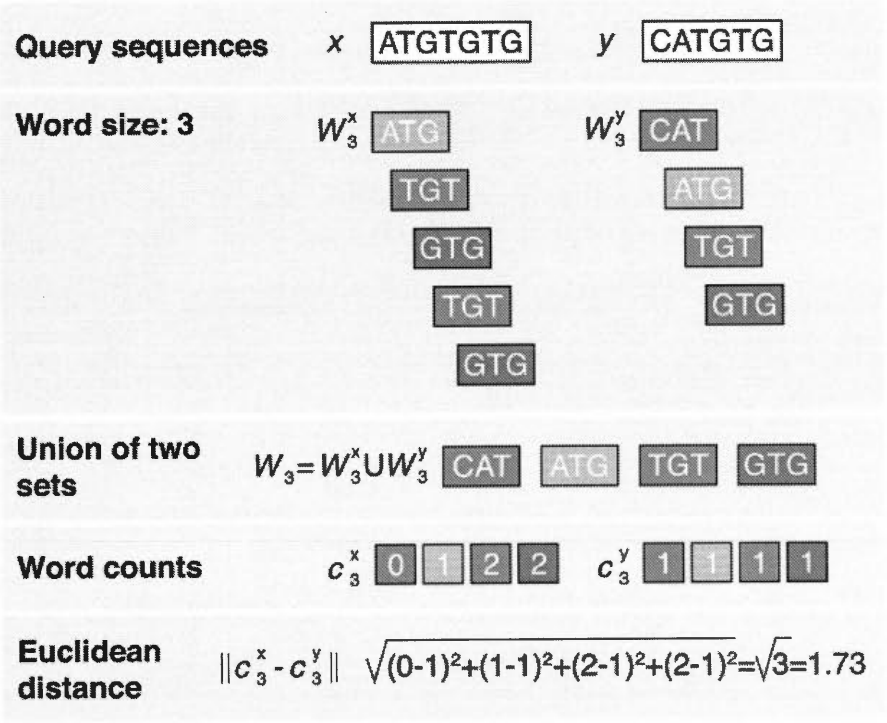

Figure 3.1 Exemple d'une méthode de classification indépendante de l'alignement. Source : (Zielezinski *et al.*, 2017)

La figure présente un calcul de la distance euclidienne entre deux séquences d'ADN (x et y) basé sur les k-mers. Dans le cas présent, si x était une séquence de référence, un score de distance faible avec y, impliquerait une forte probabilité pour que y soit associé à la classe d'appartenance de x.

des attributs pertinents. A date, les k-mers (sous-séquences nucléotidiques de longueur k) semblent constituer des attributs de qualité pour la classification et la comparaison de séquences génomiques (Bonham-Carter *et al.*, 2013). Leur utilisation a montré ses performances dans la classification de séquences à travers les méthodes de construction d'arbres phylogénétiques (Blaisdell, 1986), (Blaisdell, 1989), (Sims *et al.*, 2009), (Yu *et al.*, 2010b) et (Kolekar *et al.*, 2012). D'autres approches basées sur des calculs de distance ont aussi utilisé ce type d'attributs (Liu *et al.*, 2011), (Otu et Sayood, 2003) et (Nalbantoglu *et al.*, 2011). Ou encore à travers de nombreuses méthodes statistiques (Ulitsky *et al.*, 2006) (Arnau *et al.*, 2008), (Reinert *et al.*, 2009) ou de composition des séquences nucléotidiques (Lu *et al.*, 2008), (Chan *et al.*, 2012), (Sims *et al.*, 2009) et (Soares *et al.*, 2012). Encore actuellement, les k-mers constituent un élément central des méthodes de classification métagénomique les plus populaires tels que Kraken (Wood et Salzberg, 2014), CLARK (Ounit *et al.*, 2015) et VirFinder (Ren *et al.*, 2017).

3.4 Contraintes liées à l'utilisation des k-mers et outils d'extraction de motifs discriminants

Comme nous avons pu le développer dans la section 3.3, les k-mers présentent un potentiel certain pour constituer des attributs de qualité pour les méthodes de classification et de comparaison de séquence indépendante de l'alignement. Cependant, l'utilisation de ce type d'attributs est liée à certaines contraintes. Premièrement, dû à leur nombre de possibilité (4^k, avec 4 étant le nombre d'éléments de l'alphabet de nucléotide (A, C, G et T) et k étant la longueur des k-mers à explorer), leur utilisation peut présenter des limites quant à la mémoire nécessaire pour leur utilisation (Wood et Salzberg, 2014), (Ounit *et al.*, 2015). Deuxièmement, l'identification d'une longueur k de motifs appropriée à un problème de classification reste un défi toujours d'actualité (Zhang *et al.*, 2017). Il en découle de manière

logique que l'identification des motifs discriminants associé à des espcèces spécifiques de virus réprésentent une tâche complexe. L'extraction de motifs au sein de séquence d'ADN a déjà suscité de nombreux intérets (Sandve et Drabløs, 2006). Dans ce domaine, MEME (Bailey *et al.*, 2009) se présente comme la suite d'outils de référence. Leur plateforme web (`http://meme-suite.org/`) offre un grand nombre d'applications spécialisées dans la découverte et l'analyse de motifs de séquences représentant des attributs d'intérêt. De plus, leur programme principal dispose d'un mode discriminatif (Bailey *et al.*, 2010). Celui-ci permet d'extraire des motifs au sein d'un ensemble A de séquences (ensemble *primaire*), afin de discriminer par rapport à un ensemble de séquences B (ensemble de *contrôle*). Pour cela, l'algorithme va extraire les motifs dont la probabilité d'apparition est à la fois maximisée dans les séquences de l'ensemble A et à la fois minimisée dans les séquences de l'ensemble B. Ce type d'algorithme se présente comme un moyen efficace d'extraire des attributs discriminants basés sur les k-mers afin de construire des modèles de prédiction. Enfin, l'approche la plus récente permettant de répondre aussi à ce problème d'extraction de motifs (k-mers) discriminants au sein de séquences génomiques est l'algorithme de MISSEL (Fiscon *et al.*, 2016), qui a déjà été présenté antérieurement (Section : 3.1).

3.5 Définition de la problématique de recherche

Au vu de la littérature mentionnée antérieurement, nous posons la problématique de recherche suivante :

Etant donné un ensemble S de séquences génomiques virales étiquetées par leurs classes, comment identifier au sein de S un ensemble d'attributs discriminants F basés sur les k-mers ? Cet ensemble F devra dans un premier temps permettre de maximiser les performances de classifications des instances inconnues en terme de *F-mesure* pondérée. Dans un deuxième temps, l'ensemble F devra être minimal

afin de réduire la complexité du modèle de prédiction et de faciliter la compréhension de ses décisions à l'échelle humaine. Enfin ; la longueur k des k-mers identifiés pour constituer les attributs devra être optimale dans le sens où elle maximisera les deux contraintes précédentes.

CHAPITRE IV

MATÉRIEL ET MÉTHODE

4.1 Algorithme CASTOR-KRFE

4.1.1 Objectifs de l'algotihme

À travers ce projet, nous présentons une nouvelle méthode, CASTOR-KRFE (*K*-mers extraction by Recursive Feature Elimination). Cette approche permet d'extraire un ensemble d'attributs à partir de séquences génomiques virales connues afin de prédire les nouvelles séquences encore non identifiées. Avec cette dernière, nous prévoyons d'atteindre trois objectifs principaux :

1. Extraire un ensemble de sous-séquences (k-mers) discriminantes afin de former un jeu d'attributs permettant de maximiser les performances de classification virale.

2. Faire en sorte que le jeu d'attributs constitué soit minimal afin de faciliter le modèle et de rendre ses prédictions plus facilement compréhensible pour les humains.

3. Identifier la longueur optimale de k des sous-séquences extraites, qui est associé à chaque type de classification.

4.1.2 Étapes majeures de CASTOR-KRFE

L'algorithme de CASTOR-KRFE peut se diviser en plusieurs composantes majeures :

1. Vectorisation des séquences connues à partir d'attributs basés sur leur composition en k-mers.
2. Mise à l'échelle et pré-traitement de la matrice attributs formée antérieurement.
3. Sélection et évaluation d'ensembles de jeu d'attributs basés sur une élimination récursive.
4. Identification des k-mers représentant l'ensemble d'attributs optimal en fonction des entrées et paramètres associés.
5. Construction d'un modèle de prédiction à partir des attributs identifiés.

4.1.3 Description détaillée de l'algorithme

L'algorithme CASTOR-KRFE prend deux entrées principales :

— S, un ensemble de n séquences nucléotidiques connues, de telle manière que $S = \{s_1,\ s_2,\ \ldots\ s_{n-1},\ s_n\}$, où s_i est une chaîne de caractères formée d'un alphabet fini $\Omega = \{A, C, G, T\}$;

— y, un vecteur des étiquettes correspondant aux classes de chaque séquence $\in S$

À celles-ci s'ajoute différents paramètres associés tels que k_{min} et k_{max} se définissant de la façon suivante : soit k_{min} et k_{max} respectivement les longueurs minimales et maximales des k-mers et K l'ensemble des longueurs à explorer où $K = \{k_{min},\ k_{min}+1\ \ldots\ k_{max}-1,\ k_{max}\}$.

Un k-mer est défini comme une sous-séquence superposée d'une séquence s_i avec une longueur de k nucléotides. Le nombre de k-mers possible est théoriquement borné par 4^k (4 correspond à la cardinalité de Ω).
D'autres paramètres (f_{min} et f_{max}) relatifs au nombre minimum et maximum d'attributs potentiels à extraire, ainsi qu'un seuil T de performance à conserver lors de la réduction du nombre d'attributs pour constituer l'ensemble final minimal sont aussi à renseigner.

Dans l'algorithme 1, pour chaque $k \in K$, l'ensemble des k-mers présents dans S sont extraits et une matrice d'occurrences X est calculée. X contient, pour chaque séquence $s_i \in S$, le nombre d'occurrences des k-mers. Par la suite, un pré-traitement réalisant une mise à l'échelle min-max entre 0 et 1 à la matrice X est appliquée. L'algorithme CASTOR-KRFE analysera ensuite si le nombre actuel d'attributs (correspondant au nombre de colonnes de la matrice X) est supérieur au nombre maximum d'attributs fixé en paramètre (f_{max}). Si tel est le cas, une sélection préliminaire utilisant l'élimination récursive des attributs basée sur les séparateurs à vaste marge (SVM-RFE) (Guyon *et al.*, 2002) sera appliquée à la matrice attributs X.

SVM-RFE construit de manière itérative un modèle et effectue une sélection des attributs les plus discriminants en fonction d'un classement de ces derniers. Ce classement est basé sur une pondération qui est associée aux attributs. Celle-ci est déterminée par le classifieur lors de son entraînement. À chaque itération, un certain nombre d'attributs (ceux dont le classement est le moins bon) sont éliminés jusqu'à atteindre le nombre d'attributs voulu (f_{min}). Par défaut, le pas d'élimination à chaque itération du SVM-RFE préliminaire est fixé à 10% du nombre d'attributs total. Cette valeur permet d'éliminer rapidement une majorité des attributs non pertinents afin de ne conserver que f_{max} attributs.

Algorithm 1: CASTOR-KRFE : Extracteur d'attributs

Input : S : séquences nucléotidiques étiquetées,

k_{min} : longueur minimum de k,

k_{max} : longueur maximum de k,

f_{min} : nombre minimum d'attributs,

f_{max} : nombre maximum d'attributs,

Output: f_{list} liste des ensembles d'attributs potentiels,

s_{list} : liste des scores associés à f_{list}

Begin
  $f_{list} \leftarrow \emptyset$
  $s_{list} \leftarrow \emptyset$
  foreach $k \in [k_{min} \ldots k_{max}]$ **do**
    $D \leftarrow k$-mers $\in S$
    foreach $s_i \in S$ **do**
      $X \leftarrow$ ocurrences de chaque k-mers $\in D$
    end
    $X \leftarrow$ **MinMaxScaler**(0, 1)
    if *nombre d'attributs de* $X > f_{max}$ **then**
      $X \leftarrow$ SVM-RFE jusqu'à f_{max} attributs
    foreach $f \in [f_{min} \ldots f_{max}]$ **do**
      $X \leftarrow$ SVM-RFE jusqu'à f attributs
      $f_{list} \leftarrow$ attributs de X
      $s_{list} \leftarrow$ Score de validation croisée (X)
    end
  end
End

Une fois cette étape préliminaire terminée, pour chaque nombre d'attributs f allant de f_{max} à f_{min}, SVM-RFE sera une nouvelle fois appliqué afin de réduire X à f attributs en utilisant cette fois-ci un pas d'élimination de 1. Chaque ensemble d'attributs formé sera évalué à l'aide d'une validation croisée stratifiée, d'une métrique de performance (*F-measure* pondérée) et d'un classifieur (SVM). Enfin, les ensembles d'attributs ainsi que leurs scores de performance associés sont sauvegardés. CASTOR-KRFE utilise un SVM à noyau linéaire. Celui-ci ne possède qu'un seul paramètre C (pénalité du terme d'erreur) significatif à ajuster (Hsu *et al.*, 2003), dont nous avons laissé la valeur 1 par défaut pour l'ensemble de nos évaluations.

Dans la dernière partie (algorithme 2), la liste des scores de performance (s_{list}) associée à la liste des ensembles d'attributs optimaux potentiels f_{list} sont analysées. Cette analyse a pour objectif d'extraire l'ensemble optimal d'attributs basé sur les k-mers ainsi que la longueur k optimale des sous-séquences, en satisfaisant deux conditions :

1. L'ensemble optimal d'attributs doit avoir un score de *F-measure* supérieur au score maximum de la liste multiplié par le seuil de performance T (pourcentage de performance à maintenir par rapport au meilleur score de s_{list}).
2. L'ensemble optimal doit contenir le plus petit nombre d'attributs possible tout en remplissant la première condition.

Enfin, l'ensemble final sélectionné sera associé à un algorithme d'apprentissage supervisé afin de construire un modèle de prédiction.

4.1.4 Implémentation de CASTOR-KRFE

L'algorithme CASTOR-KRFE a été implémenté dans le langage de programmation Python 3.6 (`https://www.python.org/`). La phase de traitement sur les sé-

Algorithm 2: CASTOR-KRFE : Identificateur d'un ensemble optimal d'attributs et constructeur de modèle de prédiction

Input : f_{list} liste des ensembles d'attributs potentiels,

s_{list} : liste des scores associés à f_{list},

T : seuil de performance à conserver

Algorithm : algorithme d'apprentissage supervisé

Output: k_{length} : longueur optimal de k,

f_{set} : ensemble optimal de k-mers,

model : modèle de prédiction basé sur f_{set}

```
1  Begin
2      best_score ← max(s_list)
3      f_set ← ensemble d'attributs ∈ f_list associé au best_score
4      n ← longueur de f_set
5      k_length ← longueur des k-mers ∈ f_set
6      foreach f, s ∈ f_list, s_list do
7          if s >= best_score * T and longueur de f < n then
8              f_set ← f
9              n ← longueur de f
10             k_length ← longueur des k-mers ∈ f
11     end
12     model ← (Algorithm, f_set)
13 End
```

quences nucléotidiques a été assistée à l'aide de la bibliothèque Biopython (Cock *et al.*, 2009), et la partie apprentissage automatique, quant à elle, fut réalisée grâce à l'API Scikit-Learn (Pedregosa *et al.*, 2011). Le code source de CASTOR-KRFE est disponible à l'adresse suivante : `https://github.com/T3ZUK4/CASTOR_KRFE`.

4.2 Choix des méthodes utilisées

4.2.1 Technique de pré-traitement

Pour le choix de la méthode de pré-taitement, la mise à l'échelle min-max entre 0 et 1 a été choisie pour plusieurs raisons. Premièrement, comme évoqué dans la section 2.2, cette méthode permet de recentrer la matrice attributs entre de nouvelles bornes sans en modifier réellement la représentation des données. Deuxièmement, ce type de mise à l'échelle est recommandé lorsqu'il est associé à l'utilisation de SVM (Hsu *et al.*, 2003), car celui-ci permet d'éviter la prédominance d'attributs à plus grandes échelles numériques sur celles à plus petites et diminue les difficultés de calcul lors des étapes qui précèdent. Troisièmement, lors d'une comparaison avec d'autres méthodes de mise à l'échelle des données, l'approche min-max a démontré qu'une fois appliquée elle permettait d'obtenir des meilleures performances de prédiction par les classifieurs qu'avec d'autres méthodes de normalisation (Al Shalabi *et al.*, 2006).

4.2.2 Algorithme d'apprentissage supervisé

Concernant le choix de SVM en tant qu'algorithme de classification supervisé, celui-ci a été fondé sur plusieurs éléments. Le premier a été l'analyse comparative des algorithmes de classification supervisée réalisée par (Kotsiantis *et al.*, 2007). Cette dernière mettait en avant la capacité des SVM à obtenir généralement les meilleures performances de prédiction. De plus, dans un cadre similaire à celui

de notre recherche portant sur la prédiction de séquences génomiques virales, SVM a encore une fois surpassé de manière générale les performances des autres classifieurs (Remita *et al.*, 2017). Afin de valider définitivement notre choix, nous avons réalisé entre différents algorithmes de classification supervisée notre propre étude comparative sur nos ensembles de données virales (Figure : 4.2.2), où SVM c'est une fois de plus imposé. Les algorithmes évalués ainsi que leurs paramètres principaux étaient :

1. *SVM* : kernel = linear, C = 1.0
   Informations complémentaires : `https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC`
2. *Random Forest* : criterion = gini, number_estimators = 100, max_depth = None, max_features = n_features
   Informations complémentaires : `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier`
3. *Multi-layer Perceptron* : alpha : 0.0001, hidden_layer_sizes = 200, activation = 'relu' (fonction d'unité linéaire rectifiée), solver = 'adam' (optimiseur basé sur le gradient stochastique)
   Informations complémentaires : `https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier`
4. *Multinomial Naive Bayes* : alpha = 1.0
   Informations complémentaires : `https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html`
5. *k-nearest neighbors* : n_neighbors = 5, metric = 'minkowski'
   Informations complémentaires : `https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html`

Les différents paramètres ont été fixés à l'aide de l'algorithme GridSearchCV afin d'identifier les combinaisons qui maximisaient les performances.

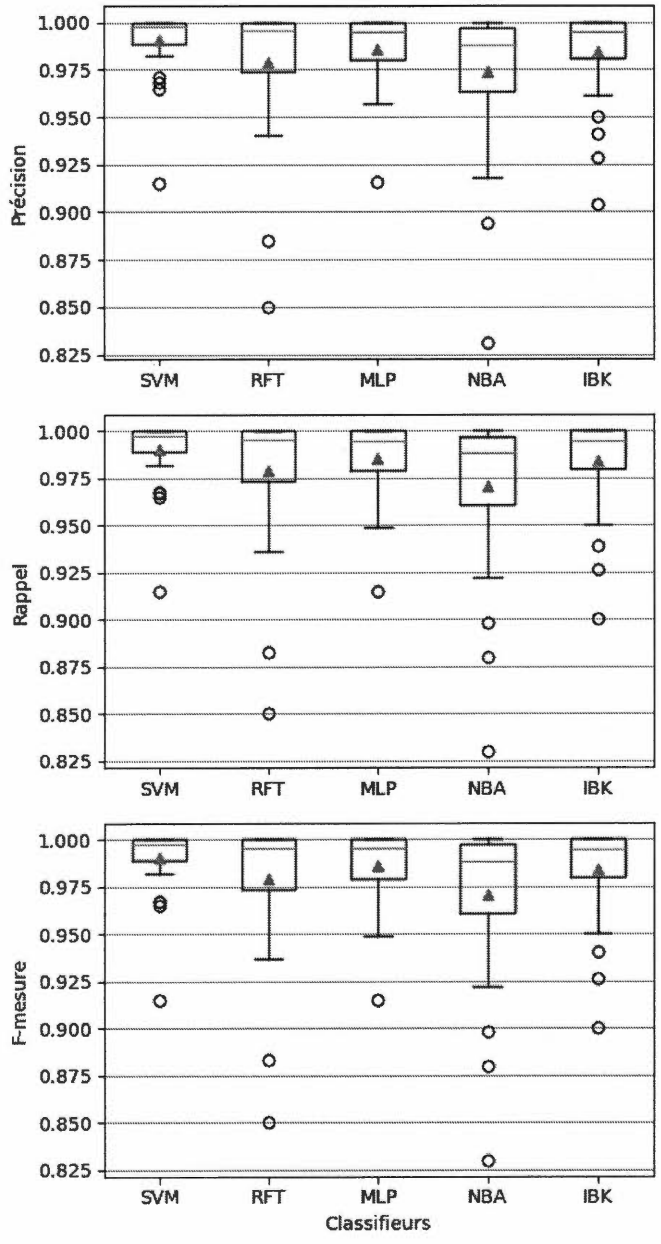

Figure 4.1 Comparaison des performances des algorithmes de classification supervisées sur données génomiques virales

La figure illustre les distributions de différentes métriques de performance (Précision, Rappel et F-mesure) pour chaque algorithme de classification supervisée. Les résultats sont obtenus depuis des évaluations par validation croisée 10 portant sur 16 ensembles de données virales (Section : 4.3). Pour chaque ensemble de données les k-mers de longueur $k = 5$ ont été choisi comme attributs. Légende : SVM = Support vector machine, RFT = Random forest, MLP = Multilayer perceptron, NBA = Multinomial Naive Bayes, IBK = k nearest neighbor, ligne orange = médiane et triangle vert = moyenne.

4.2.3 Méthode de sélection des attributs

Pour la méthode de sélection des attributs, notre décision s'est orientée vers l'élimination récursive des attributs basée sur les séparateurs à vaste marge (SVM-RFE) (Guyon *et al.*, 2002). Tel que mentionné dans la section 2.3, cette approche de type *Embbeded* possède l'avantage d'intégrer le modèle dans sa recherche d'attributs tout en étant moins coûteuse que les méthodes de type $Wrapper$. De plus, SVM-RFE a démontré de très bonnes performances sur des données à grande dimension telle que la sélection de gènes (Guyon *et al.*, 2002). Ces résultats laissent présager une bonne capacité de SVM-RFE dans l'identification des k-mers discriminants au sein des séquences virales. Afin de confirmer nos hypothèses, nous avons comme pour les algorithmes de classification supervisée réalisé une comparaison des méthodes de sélection d'attributs sur nos ensembles de données virales (Figure : 4.2). Dans cette évaluation, SVM-RFE obtient les meilleures performances. Il est ensuite suivi par les approches de type *Wrapper*, puis les méthodes *Filter* multivariées et univariées. Ces résultats semblent donc en accord avec les éléments évoqués dans la littérature de la section 2.3.

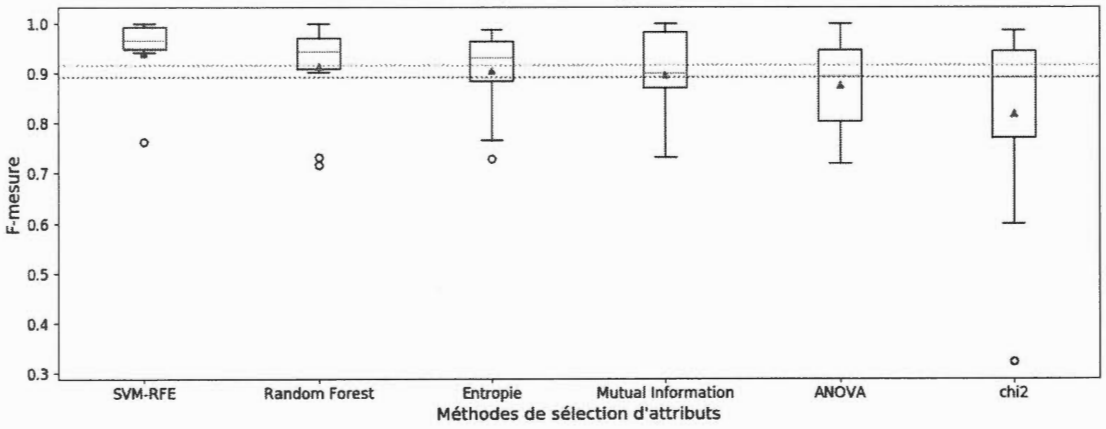

Figure 4.2 Comparaison des performances des méthodes de sélection d'attributs

La figure illustre une évaluation des performances de différentes méthodes de sélection d'attributs. L'évaluation consistait à extraire les 50 meilleurs attributs basés sur les k-mers de longueur k = [3-10]. Pour chaque méthode des modèles de prédiction ont été formés à l'aide d'un SVM et des attributs sélectionnés. Les modèles ont ensuite été évalué par une validation croisée 10. Les boxplots illustrent les distributions des F-mesure obtenues à partir de 16 ensembles de données virales (Section : 4.3). Les lignes complètes oranges et les triangles verts correspondent respectivement aux valeurs médiane et moyenne de chaque méthode. Les lignes pointillées oranges et vertes représentent respectivement les valeurs médiane et moyenne de toutes les méthodes.

4.3 Jeux de données virales

4.3.1 Présentation des virus étudiés

Notre approche CASTOR-KRFE, a été appliqué sur un large ensemble de séquences génomiques virales couvrant les sept groupes majeurs de la classification Baltimore (Baltimore, 1971) (Figure :1.6). Parmi ces jeux de données hétérogènes,

nous avons couvert des virus affectant actuellement la population mondiale tels que le virus Ebola, le virus de l'immunodéficience humaine (VIH-1), le virus de l'hépatite C (VHC) ou encore le virus de la Dengue. L'identification rapide et précise de ce type de pathogènes peut avoir des impacts importants pour leurs études, ainsi que pour la préservation et la protection de la vie et de la santé humaine.

Aujourd'hui encore, une contamination par le virus Ebola expose à un taux de létalité variant de 30 % à 90 %, en fonction de son type associé (Baize *et al.*, 2014). Quant au virus de la Dengue, il représente une maladie ré-émergente qui est présente dans les régions tropicales et sous-tropicales de l'Afrique, de l'Asie, de l'Amérique et du Pacifique (Vaughn *et al.*, 2000). Chaque année, nous estimons à plus de 100 millions le nombre de personnes infectées par ce virus (Halstead, 1988). L'infection par le virus de la Dengue est associée aux syndrômes de la fièvre et du choc hémorragique qui sont l'une des principales causes de morbidité et de mortalité pédiatrique en Asie tropicale (sangkawibha *et al.*, 1984). Certains de ces virus ré-émergents sont aussi très complexes, ce qui est le cas pour VIH-1 et VHC. Leur complexité est le résultat de fortes probabilités de mutation dans leurs génomes, ce qui implique au sein de ces groupes de virus de nombreux sous-types et recombinants. Concernant HCV, ce dernier possède plusieurs génotype (6 confirmés actuellement), dont ils diffèrent les uns des autres d'environ 30% à l'échelle des nucléotides. De plus, chaque génotypes se décompose en multiple sous-types (20 confirmés au total) qui divergent eux-mêmes entre eux de 20 % au niveau de leur composition nucléotidique (Simmonds *et al.*, 2005).

La classification du VIH-1 représente elle aussi un intérêt et une compléxité majeurs. Le groupe M prédominant du VIH-1 est divisé en 13 sous-types. La variation génétique à l'intérieur d'un sous-type peut être de 15% à 20%, tandis que la variation entre les sous-types est habituellement de 25% à 35% (Taylor

et al., 2008). Un taux de mutation et de recombinaison très élevés, représentent les deux caractéristiques essentielles à son cycle de réplication et lui permettent une propagation continue à travers les populations dans le monde entier, entraînant une pandémie d'une complexité génétique et géographique sans précédent (Taylor *et al.*, 2008). À l'heure actuelle, plus de 90 classes de recombinants dérivés des 13 sous-types de base ont vu le jour sur la base de données de *Los Alamos HIV* (`http://www.hiv.lanl.gov/`).

D'autres virus d'intérêts avec des impacts à grande échelle sur la population mondiale ont été incorporés dans nos études. L'un d'entre eux est le virus Influenza, aussi communément appelé virus de la grippe, qui réprésente d'importants centres d'intérêts épidémiologiques et cliniques. C'est d'ailleurs pour ces raisons qu'une surveillance épidémiologique et virologique renforcée de la grippe en Europe a récemment été mise en place (Fiscon *et al.*, 2016). Divers autres virus tels que le virus Polyoma et le Rhino virus humain ont été inclus parmi nos jeux de données. Ces derniers suscitent encore beaucoup de questions sur leur évolution, leur réaction, leur contribution aux maladies ou encore la sévérité clinique de certaines de leurs espèces (Pierangeli *et al.*, 2013). Enfin, nous avons étudié les virus de la famille des Geminiviridae, qui sont responsables de l'infection de nombreuses cultures et causent d'importantes pertes économiques mondiales (Silva *et al.*, 2017a). L'ensemble des jeux de données virales et leurs caractéristiques associées sont disponibles dans le tableau 4.1.

4.3.2 Formation des ensembles de données virales

Tableau 4.1 Ensembles de données virales

Jeux de données	Groupes	Virus	Fragments	Nb Nucléotides	Classifications	Nb Instances	Nb Classes
HPVGENCG	I (dsDNA)	Human papillomavirus	CG	7610	Genus	125 [27-50]	3
HPVSPECG	I (dsDNA)	Human papillomavirus	CG	7905	Species	118 [7-25]	8
POLSPEVP1	I (dsDNA)	Polyomavirus	VP1	1065	Species	121 [1-26]	13
POLSPEVP2	I (dsDNA)	Polyomavirus	VP2	726	Species	121 [1-26]	13
POLSPEVP3	I (dsDNA)	Polyomavirus	VP3	588	Species	115 [1-25]	13
POLSPEST	I (dsDNA)	Polyomavirus	ST	519	Species	127 [1-28]	13
POLSPELT	I (dsDNA)	Polyomavirus	LT	828	Species	127 [1-26]	13
GEMGENCG	II (ssDNA)	Geminiviridae	CG	2870	Genus	299 [25-50]	7
ROTSPECG	III (dsRNA)	Rotavirus	CG	1264	Species	146 [46-50]	3
RHISPECG	IV ((+)ssRNA)	Rhino virus	VP4/2	369	Species	1316 [209-752]	3
DENSPECG	IV ((+)ssRNA)	Dengue virus	CG	10655	Species	250 [50-50]	4
HCVGENCG	IV ((+)ssRNA)	Hepatitis C virus	CG	9538	Genotypes	284 [17-80]	6
HCVSUBCG	IV ((+)ssRNA)	Hepatitis C virus	CG	9538	Subtypes	284 [4-25]	18
INSUBFNA	V ((-)ssRNA)	Influenza virus	NA	1410	Subtypes	10715 [4716-5999]	2
INFSUBHA	V ((-)ssRNA)	Influenza virus	HA	1701	Subtypes	10925 [4715-6110]	2
INFSUBMP	V ((-)ssRNA)	Influenza virus	MP	756	Subtypes	21421 [9427-11994]	2
EBOSPECG	V ((-)ssRNA)	Ebola virus	CG	18982	Species	88 [4-31]	5
HIVGRPCG	VI (ssRNA-RT)	HIV-1	CG	9164	Groups	76 [4-32]	4
HIVSUBCG	VI (ssRNA-RT)	HIV-1 group M	CG	7905	Subtypes	597 [10-50]	18
HIVSUBPOL	VI (ssRNA-RT)	HIV-1 group M	pol	1211	Subtypes	1352 [33-50]	28
HBVGENCG	VII (dsDNA-RT)	Hepatitis B virus	CG	3189	Genotypes	230 [21-30]	8

Ce tableau présente les informations relatives aux ensemble de données collectées (nom de l'ensemble de données, groupe d'affiliation des virus, taxonomie, fragments de séquence utilisés, longueur moyenne des séquences, le type de classification, nombre d'instances [min -max] et enfin nombre de classes). Légende : DNA = ADN, RNA = ARN, m = messager, ds = double brin, ss = simple brin, RT = transcriptase inverse, CG = génome complet, VPn = protéine virale numéro n, ST = petit antigène tumoral , LT = grand antigène tumoral, HA = Hemagglutinin, NA = Neuraminidase, MP = protéine M1 + protéine M2 et pol = fragment pol.

Les ensembles de données sur les virus de l'immunodéficience humaine (HIV-GRPCG, HIVSUBCG et HIVSUBPOL), les virus du papillome humain (HPV-GENCG et HPVSPECG) et les virus de l'hépatite B (HBVGENCG) ont été collectés directement depuis la base de données CASTOR (Remita *et al.*, 2017). Les séquences génomiques des polyomavirus (POLSPEVP1, POLSPEVP2, POLS-PEVP3, POLSPEVP3, POLSPEST et POLSPELT), des virus de la grippe (IN-SUBFNA, INFSUBHA et INFSUBMP) et des rhinovirus (RHISPECG) ont été obtenues à partir des ressources supplémentaires de l'article de MISSEL (Fiscon *et al.*, 2016). Les ensembles de données sur les virus de la Dengue (DENSPECG), les virus Ebola (EBOSPECG) et les rotavirus (ROTSPECG) ont été constitués à partir de la base de données Virus Pathogen Ressource (VIPR) (Pickett *et al.*, 2011). Les données sur le virus de l'hépatite C (HCVGGENCG et HCVSUBCG) ont été recueillies à partir de la base de données de VIPR et de Los Alamos HCV (Kuiken *et al.*, 2004). Enfin, le jeu de données Geminiviridae (GEMGENCG) a été construit à partir de la base de données Geminivirus (Silva *et al.*, 2017a).

4.3.3 Création de bases de données de prédiction

En addition à cela, nous avons collecté et formé de grands ensembles de données depuis les bases de données virales *open data* disponibles. L'objectif était de pouvoir constituer des ensembles de test consistants pour les modèles de prédiction générés par CASTOR-KRFE. Les premiers ensembles de données formés ont été pour les virus les plus complexes (VIH-1 et VHC).

Concernant le VIH-1, nous avons collecté l'ensemble des génomes complets (3778 séquences) et l'ensemble des fragments *pol* (119 005 séquences) depuis la base de données de *Los Alamos HIV*.

Pour le VHC, nous avons créé notre ensemble de test depuis l'union des bases de

données de VIPR (Pickett *et al.*, 2011) et de *Los Alamos HCV* (Kuiken *et al.*, 2004). Il en a résulté un jeu de données de 3455 génomes complets confirmés pouvant être prédits à l'échelle des génotypes et des sous-types.

Un ensemble de données supplémentaire, portant sur le virus de la Dengue a aussi été constitué. Cet ensemble est formé de 4938 génomes complets collectés depuis la base de données de *NCBI Virus Variation Resource* (Hatcher *et al.*, 2016).

Enfin, un dernier ensemble de test concernant le virus Ebola a été réalisé. Ce dernier est composé de 2045 génomes complets collectés depuis les bases de données de VIPR (Pickett *et al.*, 2011), de NCBI Virus Variation Resource (Hatcher *et al.*, 2016) et de *Los Alamos Ebola Hemorrhagic Fever Virus* (Kuiken *et al.*, 2011).

CHAPITRE V

ÉVALUATION

5.1 Évaluation préliminaire de CASTOR-KRFE sur données simulées

5.1.1 Génération des jeux de séquences simulées et mise en place de l'évaluation

La première évaluation que nous avons mis en place portait sur des jeux de séquences simulées. Ce type d'étude, dans un cadre contrôlé, nous permet d'analyser la capacité de notre approche à extraire le jeu minimal d'attributs basé sur les k-mers permettant de discriminer les différentes classes d'un ensemble de séquences. Nous avons donc généré plusieurs ensembles de séquences simulées par la procédure suivante :

Soit S un ensemble de n séquences de longueur l, suivant une distribution uniforme des caractères (A, C, G, T). L'ensemble S est divisé en C classes au nombre d'instances I équivalentes. Pour chaque classe $c \in C$: On génère un motif m aléatoire de longueur k. On introduit ensuite m à une position aléatoire de chaque séquence $s \in c$.

La figure 5.1 illustre la procédure de génération des séquences simulées. Puis, nous avons appliqué CASTOR-KRFE sur chaque ensemble S pour identifier les motifs m discriminants de chaque classe c, la longueur optimale k associée à m et enfin construire et évaluer un modèle de prédiction à partir des motifs extraits.

Le tableau 5.1 résume les diverses expériences réalisées, les paramètres associés à ces dernières, ainsi que les résultats obtenus.

Concernant les paramètres relatifs aux ensembles de séquences simulées, nous avons fait varier les valeurs telles que la longueur des séquences $l = [1000, 5000, 10\,000]$, la longueur des motifs discriminants $k = [10, 15, 20]$ et le nombre de classes $C = [2, 10, 20]$. Enfin, les paramètres utilisés par CASTOR-KRFE sont $T = 0.99$, $k_{min} = 1$, $k_{max} = 25$, $f_{min} = 1$ et $f_{max} = 50$.

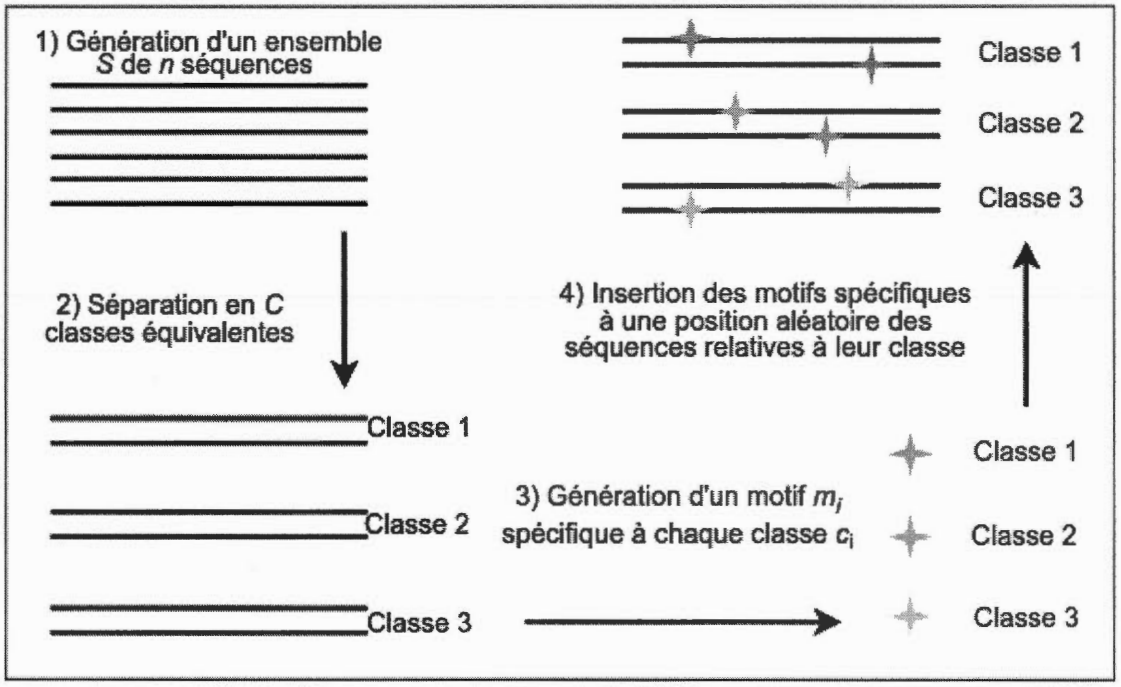

Figure 5.1 Procédure de génération d'un ensemble de séquences simulées

5.1.2 Analyse des résultats de l'évaluation sur les ensembles de séquences simulées

En se focalisant dans un premier temps sur les colonnes "k" et "k identifié" du tableau 5.1, correspondant respectivement à la longueur des motifs discriminants

Tableau 5.1 Expériences et résultats de l'évaluation portant sur les ensembles de données simulées

Longueur des séquences	k	Nombre de classes	Nombre de seq/clas	Nombre de k-mers identifiés	k identifié	*F-mesure*
1000	10	2	100	1	10	1,000
		10	20	9	10	1,000
		20	10	19	10	0,985
	15	2	100	1	15	1,000
		10	20	9	15	1,000
		20	10	19	15	1,000
	20	2	100	1	20	1,000
		10	20	9	20	1,000
		20	10	19	20	1,000
5000	10	2	100	1	10	1,000
		10	20	9	10	0,985
		20	10	19	10	0,935
	15	2	100	1	15	1,000
		10	20	9	15	1,000
		20	10	19	15	1,000
	20	2	100	1	20	1,000
		10	20	9	20	1,000
		20	10	19	20	1,000
10000	10	2	100	1	10	0,995
		10	20	9	10	0,950
		20	10	19	10	0,920
	15	2	100	1	15	1,000
		10	20	9	15	1,000
		20	10	19	15	1,000
	20	2	100	1	20	1,000
		10	20	9	20	1,000
		20	10	19	20	1,000

Ce tableau présente les évaluations et résultats obtenus à partir des ensembles de séquences simulées. Les quatre premières colonnes font respectivement référence à la longueur l des séquences, la longueur k des motifs discriminants introduits, le nombre de classe C de l'ensemble de séquence et enfin le nombre n de séquences par classe. Les trois dernières colonnes représentent les résultats obtenus par CASTOR-KRFE sur les différents jeux de données. Avec premièrement le nombre de motifs (k-mers) identifiés, deuxièmement la longueur k optimale identifiée, et troisièmement, la F-mesure obtenue à partir de l'évaluation par validation croisée 10 du modèle de prédiction formé depuis les motifs extraits.

introduits dans les séquences et la longueur optimale des motifs identifiés par CASTOR-KRFE, nous pouvons voir que les valeurs sont identiques et que notre algorithme a donc été en mesure d'identifier la longueur exacte des motifs discriminants introduits pour chaque cas de figure.

Si nous regardons dans un deuxième temps la colonne "Nombre de k-mers identifiés", nous pouvons voir que les valeurs de cette colonne correspondent à celle de la colonne "Nombre de classes" moins 1. Après une vérification, nous avons confirmé que les motifs identifiés par CASTOR-KRFE appartenaient dans 100% des cas aux motifs discriminants insérés dans les séquences simulées. Par conséquent, pour chaque ensemble de séquences, CASTOR-KRFE a extrait l'ensemble de motifs discriminants en en excluant un dernier. En effet, dans sa recherche de l'ensemble minimum d'attributs, CASTOR-KRFE a considéré que chaque classe était caractérisée par son motif discriminant et que la dernière classe était représentée par une absence de motifs. La figure 5.2 illustre un exemple de la fonction de décision prise par l'algorithme CASTOR-KRFE concernant un ensemble de motifs extraits.

Enfin, les résultats de la colonne "*F-mesure*" montrent des résultats moyens de *F-mesure* très proche de 1. Nous pouvons remarquer que cette dernière chute légèrement lorsque la longueur "k" des motifs discriminants est petite et que la longueur l des séquences augmente. Après une analyse des séquences mal classifiées, nous expliquons cela par le fait que ces motifs discriminants de petite taille sont apparus dans des séquences relatives à d'autres classes durant leur génération aléatoire.

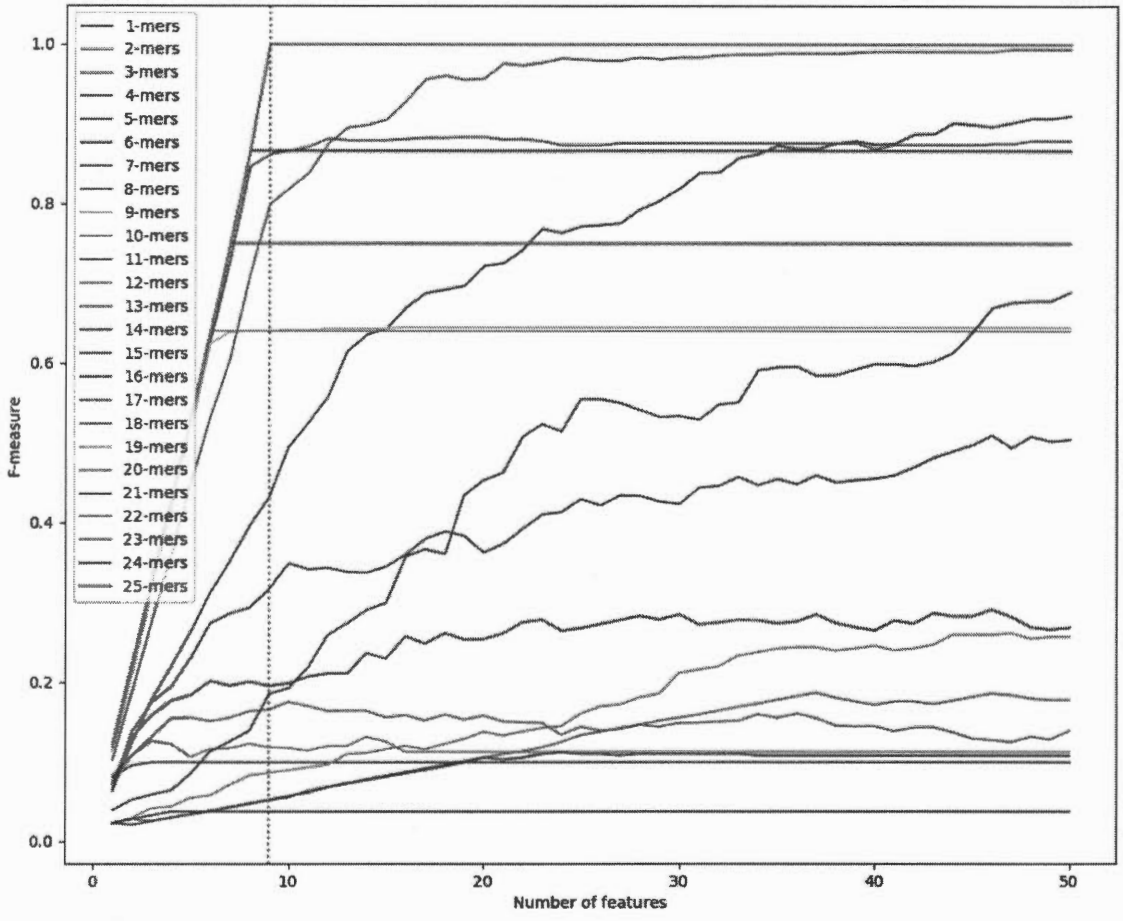

Figure 5.2 Graphe de décision de CASTOR-KRFE

Cette figure représente un exemple de graphe d'analyse de CASTOR-KRFE pour l'identification d'un ensemble d'attributs optimal et minimal. Dans cet exemple qui fait référence à l'expérience de la deuxième ligne du tableau 5.1, CASTOR-KRFE a extrait 9 attributs de longueur $k = 10$. La décision de l'algorithme est représentée par la ligne verticale en pointillé rouge.

5.2 Évaluation sur données virales réelles

5.2.1 Évaluation par partitionnement *Jackknife*

Dans l'objectif de pouvoir étudier de manière pertinente le comportement et les capacités de CASTOR-KRFE, nous avons évalué ce dernier sur un large éventail de données virales réelles. Cette étude porte sur 12 jeux de données virales couvrant les 7 grands groupes de virus selon la classification Baltimore, permettant ainsi d'avoir un environnement hétérogène. Les jeux de données utilisés sont : HBVGENCG pour le virus de l'hépatite B (VHB), HIVGRPCG, HIVSUBCG et HIVSUBPOL pour le virus de l'immunodéficience humaine 1 (VIH-1), HPVGENCG et HPVSPECG pour le virus du papillome humain (VPH), HCVGENCG et HCVSUBCG pour le virus de l'hépatite C (VHC), DENSPECG pour le virus de la Dengue, GEMGENCG, pour les Geminivirus, EBOSPECG pour le virus Ebola et enfin, ROTSPECG pour le Rotavirus.

Le tableau 4.1 offre des informations plus complètes sur les jeux de données utilisés. Nous avons évalué CASTOR-KRFE sur ces données en formant différents ensembles d'entrainement et de test suivant un partitionnement Jackknife. Pour chaque ensemble de données nous avons réalisé 100 itérations. À chaque itération, les données ont été divisées aléatoirement en 80 % (partie entrainement) et 20 % (partie test). CASTOR-KRFE a été appliqué sur les parties entrainement afin d'extraire des attributs et de constituer des modèles de prédiction. La performance de ces modèles a ensuite été évaluée par la prédiction sur les parties test. La métrique de performance utilisée pour cette évaluation est la *F-mesure* pondérée. Les paramètres utilisés par CASTOR-KRFE sont $T = 0.99$, $k_{min} = 1$, $k_{max} = 30$, $f_{min} = 1$ et $f_{max} = 100$. Ces derniers ont été fixés de manière à permettre d'identifier des ensembles minimaux d'attributs, maintenant une bonne performance de classification en terme de *F-mesure* et en couvrant un large champ de longueur de

k-mers. La figure 5.3 illustre les résultats obtenus lors des différentes prédictions de chaque jeu de données.

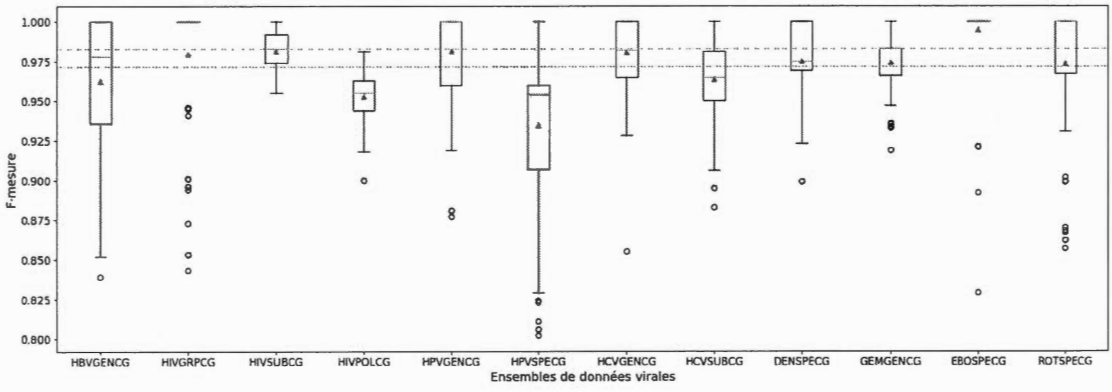

Figure 5.3 Performances de prédiction sur ensembles de données virales

Dans cette figure, pour chaque ensemble de données, un boxplot illustre la distribution de la F-mesure pondérée obtenue sur 100 itérations. Les cercles noirs représentent les valeurs aberrantes. Les lignes complètes oranges et les triangles verts correspondent respectivement aux valeurs médiane et moyenne de chaque ensemble de données. Les lignes pointillées oranges et vertes représentent respectivement les valeurs médiane et moyenne de tous les ensembles de données.

Dans la figure 5.3, nous pouvons voir que pour l'ensemble des prédictions réalisées, la *F-mesure* pondérée moyenne est supérieure à 0,97. De plus, au moins 75% des prédictions ont obtenu un score de *F-mesure* pondérée supérieur à 0,90. Au plus bas, certaines prédictions chutent à 0,80 de *F-mesure*. Nous pouvons expliquer cela par la présence d'un faible nombre d'instances de certaines classes où il est par conséquent difficile de construire des modèles représentatifs de la réalité. Nous pouvons par exemple prendre la classe de Alpha VHP 14 qui n'est présente que 7 fois dans le jeu de données de HPVSPECG. Enfin, les résultats globaux montrent

que CASTOR-KRFE possède une bonne capacité à extraire des attributs discriminants afin d'établir des modèles de prédictions performants.

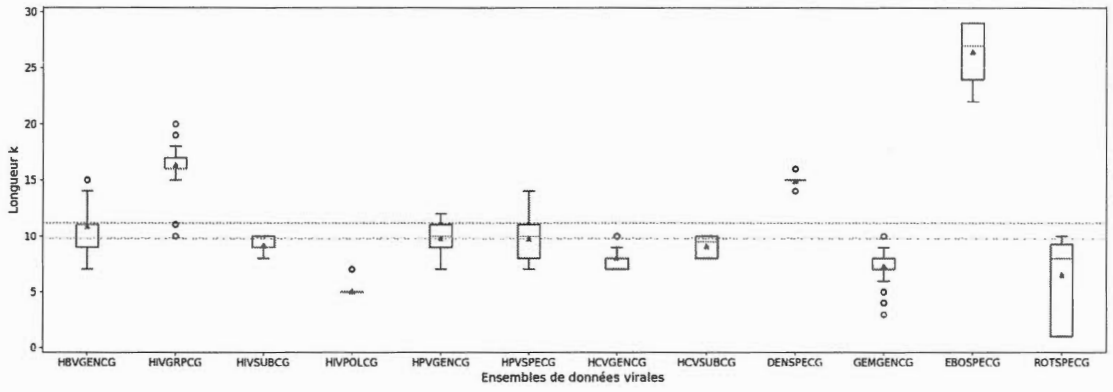

Figure 5.4 Distribution de la longueur k des motifs extraits

Dans cette figure, pour chaque ensemble de données, un boxplot illustre la distribution de la longueur k des motifs extraits sur 100 itérations. Les cercles noirs représentent les valeurs aberrantes. Les lignes complètes oranges et les triangles verts correspondent respectivement aux valeurs médiane et moyenne de chaque ensemble de données. Les lignes pointillées oranges et vertes représentent respectivement les valeurs médiane et moyenne de tous les ensembles de données.

Dans la figure 5.4 traitant de la distribution des longueurs de k des motifs extraits, nous pouvons voir que les k-mers d'une longueur aux alentours de 10 nucléotides permettent d'obtenir de bonnes performances pour différencier les classes de séquences génomiques virales. Nous pouvons également souligner que les ensembles de données composés de courtes séquences avec un grand nombre de classes sont souvent discriminés avec des k-mers de longueur inférieure à 10 (Tableau : 4.1 et Figure : 5.4). Par exemple, l'ensemble de données HIVSUBPOL, composé d'une séquence d'environ 1 300 nucléotides et comprenant 28 classes, a une longueur

discriminante de $k = 5$. Cependant, l'ensemble de données sur le virus Ebola (EBOSPECG), avec une longueur moyenne de séquence d'environ 19 000 nucléotides et cinq classes, implique une discrimination des k-mers de longueur moyenne $k = 26$.

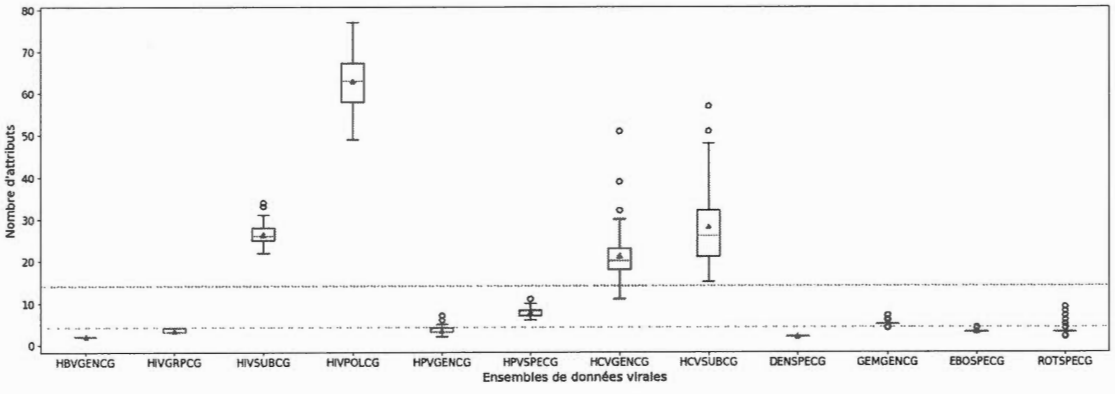

Figure 5.5 Distribution du nombre d'attributs extraits

Dans cette figure, pour chaque ensemble de données, un boxplot illustre la distribution du nombre d'attributs extraits sur 100 itérations. Les cercles noirs représentent les valeurs aberrantes. Les lignes complètes oranges et les triangles verts correspondent respectivement aux valeurs médiane et moyenne de chaque ensemble de données. Les lignes pointillées oranges et vertes représentent respectivement les valeurs médiane et moyenne de tous les ensembles de données.

En accord avec l'information du tableau 4.1, nous pouvons voir que le nombre de k-mers extraits par jeu de données présenté dans la figure 5.5 est souvent proche de son nombre de classes. Cette observation montre que notre méthode tend à rechercher des sous-séquences uniques spécifiques aux groupes de séquences associés à chaque classe. La figure 5.5 révèle également la complexité des ensembles de données HIVSUBCG, HIVSUBPOL, HCVGGENCG et HCVSUBCG. Les virus

de ces jeux de données sont fortement exposés aux mutations, aux recombinaisons et comportent de nombreux sous-types. Cette complexité est illustrée par le besoin d'un nombre plus élevé de motifs pour obtenir une *F-mesure* moyenne supérieure à 0,950. Pour ces ensembles de données complexes, le nombre d'attributs extraits est souvent trois fois plus élevé que leur nombre de classes.

5.2.2 Prediction à partir de bases de données virales

Après avoir évalué la capacité de notre solution à construire des modèles de prédictions performants sur divers jeux de données virales, nous avons voulu tester sa robustesse à une plus grande échelle. Pour cela, nous avons constitué de grands jeux de données virales en unissant les séquences disponibles des bases de données virales. Nous avons ensuite formé des modèles de prédictions à partir de nos jeux de données initiaux (Figure : 4.1) et nous avons réalisé la prédiction des différentes bases de données construites.

Le premier ensemble de données dont nous avons réalisé la prédiction est un jeu de VIH-1 composé de 3778 génomes complets. Le modèle de prédiction pour cet ensemble de données a été formé à partir du jeu de données HIVSUBCG où 74 k-mers de longueur $k = 8$ ont été extraits pour constituer les attributs discriminants. Le deuxième jeu de données portait aussi sur le VIH-1, mais est constitué de 119 005 fragments *pol*. L'ensemble de données HIVSUBPOL a été utilisé pour former le modèle de prédiction. Les attributs sont caractérisés par 96 k-mers de longueur $k = 5$. Les troisième et quatrième bases de données de prédiction formées comportent 3455 séquences de génome complet de HCV. Une prédiction à la fois à l'échelle du génotype et des sous-types ont été réalisées. Le modèle pour prédire les génotypes a été créé à partir de 50 k-mers de longueur $k = 8$ extraits depuis le jeu de données HCVGENCG. Celui pour la prédiction

des sous-type a, quant à lui, été construit à partir de 43 k-mers de longueur $k = 8$ extrait par CASTOR-KRFE depuis HCVSUBCG. Le cinquième ensemble de données que nous avons prédit est celui du virus de la Dengue contenant 4938 génomes complets. Pour cette prédiction, CASTOR-KRFE a réalisé un modèle de prédiction à partir de 2 k-mers de longueur $k = 11$ depuis le jeu de données DENSPECG. Enfin, la dernière base de données prédite est constituée de 2045 génomes complet du virus Ebola, et son modèle de prédiction a été réalisé à partir de 3 k-mers discriminants de longueur $k = 29$ extraits depuis EBOSPECG. Nous précisons qu'aucune des séquences présentes dans les ensembles de test ne faisaient partir des jeux d'entrainement sur lesquels ont été établis les modèles de prédiction. Dans une optique de maximisation des performances de prédiction, les paramètres utilisés par CASTOR-KRFE étaient $T = 1.00$, $k_{min} = 1$, $k_{max} = 30$, $f_{min} = 1$ et $f_{max} = 100$. La figure 5.6 illustre les résultats obtenus lors de la prédiction des différentes bases de données virales.

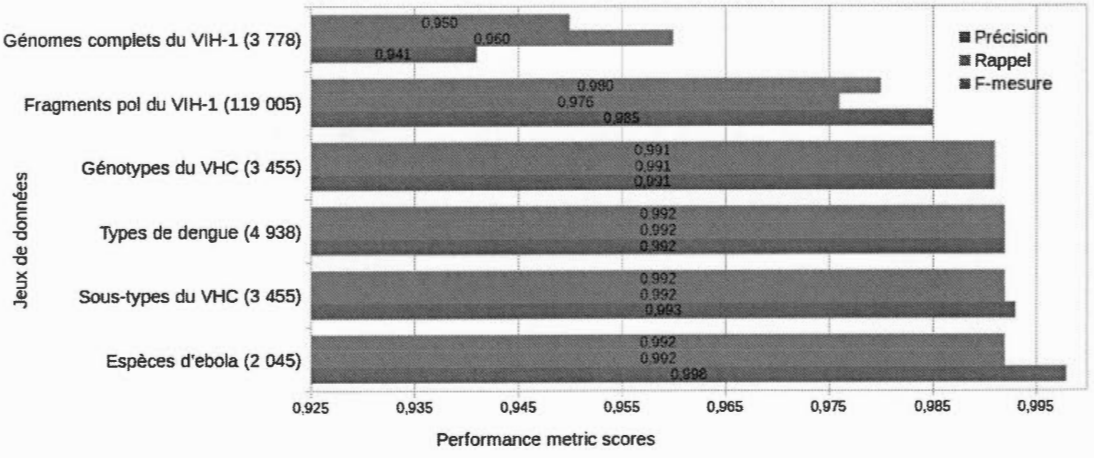

Figure 5.6 Performances de prédiction sur base de données virales

Cette figure illustre les performances prédictives de 6 bases de données virales (génomes complets et fragments pol du VIH-1 pour la classification des sous-types, génomes complets du VHC pour la classification du génotype et du

sous-type, génomes complets du virus de la Dengue pour la classification du type et génomes complets d'Ebola virus pour la classification des espèces). Les valeurs entre parenthèses indiquent le nombre d'instances de chaque ensemble de données. Les barres bleues, rouges et vertes représentent respectivement la Précision, le Rappel et la F-mesure pondérée.

Dans la figure 5.6, nous pouvons voir que les résultats globaux mettent en évidence des scores de performance supérieurs à 0,940 pour les différentes bases de données prédites. Pour l'ensemble de données portant sur les génomes complet de VIH-1, le modèle de prédiction établi par CASTOR-KRFE a permis d'obtenir des scores de 0,941 pour la *Précision*, 0,960 pour le *Rappel* et 0,950 pour la *F-mesure*. Concernant la prédiction des 119 005 fragments *pol* de VIH-1, le modèle prédictif généré par notre approche a obtenu des scores respectifs de 0,985, 0,976 et 0,980 pour la *Précision*, le *Rappel* et la *F-mesure*. Pour la prédiction des bases de données de VHC à l'échelle des génotypes et des sous-types, CASTOR-KRFE a obtenu des scores supérieurs à 0,990 pour l'ensemble des métriques de performances. De la même manière, pour les prédictions des bases de données du virus de la Dengue et de Ebola virus, les modèles prédictifs établis par CASTOR-KRFE, ont permis d'obtenir des scores supérieurs à 0,990 pour la *Précision*, le *Rappel* et la *F-mesure*.

5.3 Comparaison avec MEME (Mode Discriminatif)

5.3.1 Présentation de MEME et de l'étude comparative

Une étude comparative pertinente à mener était celle avec MEME (Bailey *et al.*, 2009). En effet, cette dernière est une plateforme de référence spécialisée dans la découverte et l'analyse de motifs de séquences biologiques représentant des attributs d'intérêts. Pour cette étude, l'outil MEME (Mode Discriminatif) (Bailey *et al.*, 2010) a été sélectionné. Parmi les nombreuses approches que propose la

plateforme MEME, ce dernier se révèle être le plus proche de notre méthode. L'algorithme de MEME mode discriminatif prend en entrée deux ensembles de séquences. Il découvre des motifs qui sont communs aux séquences du premier jeu (primaire) et qui permettent de discriminer par rapport aux séquences du second jeu (contrôle).

Afin d'évaluer et d'exploiter pleinement l'approche de MEME, nous avons mis en place l'expérience suivante pour différents ensembles de données virales.
Pour chaque classe c d'un ensemble de données D, nous sélectionnons toutes les séquences appartenant à la classe c comme ensemble primaire. Puis, toutes les autres séquences de D sont utilisées pour former l'ensemble de contrôle. Enfin, nous appliquons l'algorithme MEME pour extraire les motifs discriminants. Ces étapes sont itérées pour chaque classe c de l'ensemble de données D afin d'extraire les motifs qui discriminent chaque classe des autres. L'ensemble des motifs extraits sont ensuite utilisés pour construire un modèle de prédiction qui sera évalué par validation croisée 10 (Figure : 5.7). Afin d'avoir un ensemble de données varié et hétérogène pour cette évaluation, nous avons sélectionné les douze jeux de données utilisés dans l'évaluation sur données virales réelles (Section : 5.2). Ces derniers ont également été donnés en entrée à CASTOR-KRFE afin qu'il en extrait des motifs discriminants, construise des modèles de prédiction et en évalue les performances dans les mêmes conditions que pour MEME. Pour cette comparaison, les métriques de performances utilisées sont respectivement la *Précision*, le *Rappel* et la *F-mesure*. Concernant les paramètres utilisés, nous avons fixé la longueur minimum (k_{min}) et maximum (k_{max}) des motifs à extraire pour les deux approches respectivement à $k_{min} = 3$ et $k_{max} = 30$. Pour le nombre de motifs à extraire nous avons laissé par défaut à 3 par classes pour MEME. Pour CASTOR-KRFE, nous avons appliqué les paramètres suivants : $T = 0.99$, $f_{min} = 1$ et $f_{max} = 100$.

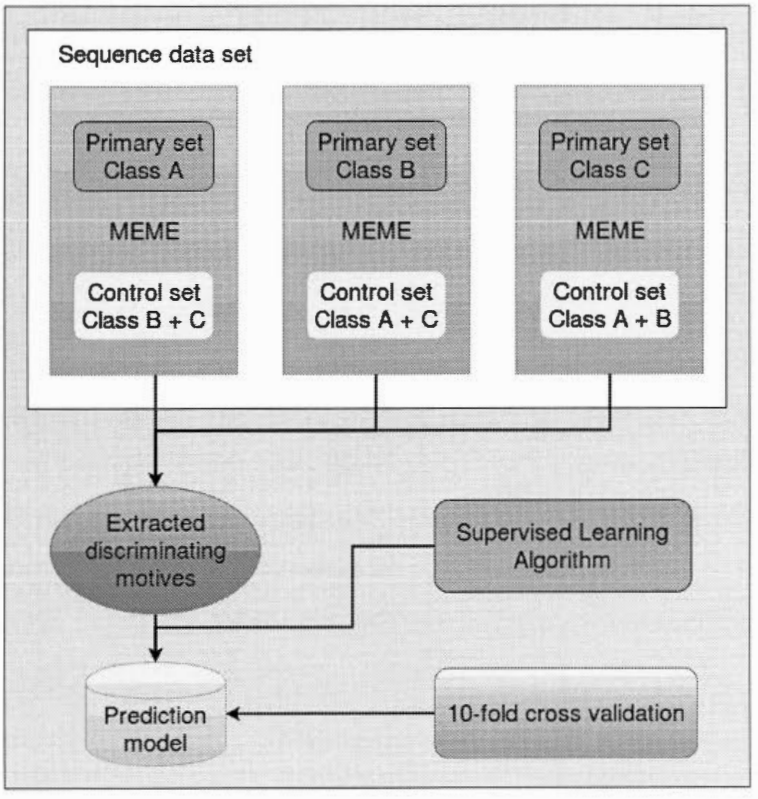

Figure 5.7 Schéma de l'évaluation de MEME

La figure illustre le pipeline d'évaluation de MEME (Mode Discriminatif). Cet exemple montre le cas de figure où nous avons un jeu de données constitué de trois classes. Les ensembles primaires et de contrôles sont dans un premier temps formés, afin d'extraire les motifs discriminants de chaque classe par rapport aux autres à l'aide de MEME. À partir des motifs extraits et d'un algorithme d'apprentissage supervisé, un modèle de prédiction est construit puis évalué par validation croisée 10.

5.3.2 Résultats de la comparaison

Le tableau 5.2 illustre les différentes informations relatives aux résultats obtenus de l'étude comparative entre MEME (Mode Discriminatif) (Bailey *et al.*, 2010) et

CASTOR-KRFE.

Tableau 5.2 Comparaison entre CASTOR-KRFE et MEME (Mode Discriminatif)

	MEME					CASTOR-KRFE				
Jeux de données	**Précision**	**Rappel**	**F-mesure**	**Motifs**	k	**Précision**	**Rappel**	**F-mesure**	**Motifs**	k
HBVGENCG	0,951	0,948	0,949	24	30	1,000	1,000	1,000	5	11
HPVGENCG	0,590	0,624	0,555	9	30	0,977	0,976	0,976	4	8
HPVSPECG	0,839	0,788	0,807	24	30	1,000	1,000	1,000	10	9
HIVGRPCG	0,975	0,974	0,974	12	30	1,000	1,000	1,000	2	17
HIVSUBCG	0,900	0,871	0,878	54	30	0,995	0,995	0,995	25	10
HIVSUBPOL	0,839	0,786	0,802	84	30	0,972	0,972	0,972	61	5
HCVGENCG	0,918	0,877	0,882	18	30	1,000	1,000	1,000	33	6
HCVSUBCG	0,910	0,863	0,875	54	30	0,988	0,986	0,986	56	8
DENSPECG	0,977	0,975	0,975	12	30	0,995	0,995	0,995	2	14
GEMGENCG	0,935	0,890	0,897	21	30	0,997	0,997	0,997	5	7
EBOSPECG	1,000	1,000	1,000	15	30	1,000	1,000	1,000	3	29
ROTSPECG	0,869	0,842	0,834	9	30	1,000	1,000	1,000	3	7
MOYENNE	**0,892**	**0,870**	**0,869**	**28**	**30**	**0,994**	**0,993**	**0,993**	**17**	**11**

Ce tableau présente les résultats de l'étude comparative entre CASTOR-KRFE et MEME (Discriminative Mode). La première colonne fournit les informations relatives aux ensembles de données utilisés. Les autres colonnes montrent respectivement pour les deux méthodes, le nombre de motifs extraits, leur longueur k et les scores des différentes métriques de performance (Précision, Rappel, F-mesure) obtenus lors de l'évaluation par validation croisée 10.

En nous focalisant dans un premier temps sur les scores des métriques de performance obtenus pour les 12 ensembles de données, nous pouvons voir que MEME a obtenu une *Précision* moyenne de 0,892, un *Rappel* moyen de 0,870 et une *F-mesure* pondérée moyenne de 0,869. Concernant CASTOR-KRFE, il améliore les scores de ces métriques d'en moyenne 14%, obtenant respectivement pour la *Précision*, le *Rappel* et la *F-mesure*, 0,994, 0,993 et 0,993. De plus, sur des jeux de données tel que HPVGENCG (classification des genres de HPV virus) les performances de MEME chutent très bas avec une *F-mesure* de 0,555. Contrairement

à CASTOR-KRFE, dont le score le plus bas est obtenu pour la classification des fragments de pol du VIH (HIVSUBPOL), avec une *F-mesure* de 0,972.

Dans un deuxième temps, en regardant le nombre de motifs extraits, nous pouvons voir que CASTOR-KRFE n'a eu besoin en moyenne que de 11 motifs pour obtenir de meilleures performances que MEME qui a eu besoin en moyenne de 28 motifs par jeu de données.

Enfin, en ce qui concerne la longueur k des motifs extraits, nous pouvons observer que CASTOR-KRFE a identifié des k-mers spécifiques et variés en fonction des différents ensembles de données. La plus petite longueur k est obtenue pour l'ensemble de données sur les fragments de pol du VIH-1 (nombre de classes = 28, longueur moyenne de séquence = 1,211) avec un $k = 5$. Le k le plus long est obtenu pour l'ensemble de données du virus Ebola (classes = 5, longueur moyenne de séquence = 18,982) avec $k = 29$. Pour les différents ensembles de données, CASTOR-KRFE a identifié une longueur moyenne de $k = 11$. En revanche, MEME a identifié des motifs dégénérés de longueur $k = 30$ pour tous les ensembles de données. Cela s'explique par le fait que l'algorithme de MEME favorise des motifs longs mais dégénérés pour discriminer les groupes de séquences. Cependant, il est bon de souligner que des motifs courts impliquent une sensibilité moindre aux mutations au sein des séquences nucléotidiques. De manière générale, ces résultats montrent que les motifs extraits par CASTOR-KRFE sont à la fois inférieurs en nombre, ce qui facilite l'interprétation de la classification, sont d'une longueur k plus petite, ce qui permet de former des attributs plus stables face aux mutations, et enfin offrent un meilleur potentiel de discrimination entre les différentes classes de virus.

5.4 Comparaison avec MISSEL

5.4.1 Définition du cadre de l'évaluation

Comme déjà mentionné antérieurement, MISSEL (Fiscon *et al.*, 2016) se présente comme une méthode supervisée capable d'extraire de multiple sous-séquences discriminantes au sein d'un groupe de séquences connues afin d'en séparer et d'en identifier les différentes espèces. Dans leur article (Fiscon *et al.*, 2016), l'algorithme de MISSEL a été appliqué à divers jeux de données du virus Influenza, Polyoma et Rhino. Ces jeux de données viraux ont été divisés en des ensembles d'entrainement et de test. Sur les jeux d'entrainement, l'algorithme de MISSEL a été appliqué afin d'extraire des sous-séquences discriminantes et par la suite former une structure de prédiction. Les ensembles de test ont servi par la suite à évaluer les performances de classification des modèles formés. Nous avons réitéré l'expérience, en l'appliquant également à notre approche et en y incluant nos propres jeux de données complexes de VIH-1 et de VHC. Les jeux de données virales utilisés pour la comparaison sont : RHISPECH pour le virus Rhino, INFSUBHA, INFSUBNA et INFSUBMP pour le virus Influenza, POLSPELT, POLSPEST, POLSPEVP1, POLSPEVP2 et POLSPEVP3 pour le virus Polyoma, HIVGRPCG, HIVSUBCG et HIVSUBPOL pour le virus de l'immunodéficience humaine et enfin HCVGENCG et HCVSUBCG pour le virus de l'hépatite C. Les informations relatives aux différents jeux de données sont disponibles dans le tableau 4.1.

Pour l'expérience, nous avons suivi les mêmes schémas de partitionnement (Entraînement / Test) des données que dans l'article (Fiscon *et al.*, 2016) de MISSEL pour les jeux de données de Influenza, Polyoma et Rhino virus. En ce qui concerne les données de VIH-1 et de VHC, 80 % ont été utilisés en tant que base d'entraînement et 20 % en tant que base de test. De plus, il est important de mentionner

que l'algorithme de MISSEL prend en entrée un jeu de séquences aligné et n'offre en paramètre que la possibilité de modifier les pourcentages d'entraînement et de test. En sortie, MISSEL ne fournit que les résultats obtenus par leurs meilleures solutions identifiées. Comme mentionné précédemment, MISSEL prend en entrée des séquences ayant subi un alignement. Cette étape a donc été réalisée à l'aide de l'algorithme MUSCLE (Edgar, 2004) et du programme (Gouy *et al.*, 2009). Concernant l'évaluation de notre approche, pour chaque jeu de données, nous avons réalisé 100 itérations d'entraînement et de test. Pour chaque itération, les jeux de données ont suivi une sélection aléatoire en respectant les mêmes pourcentages de partitionnement que MISSEL. Les résultats de nos différentes itérations ont donc été comparées aux meilleures performances obtenues par l'algorithme de MISSEL. Les paramètres utilisés par CASTOR-KRFE sont $T = 0,99$, $k_{min} = 1$, $k_{max} = 30$, $f_{min} = 1$ et $f_{max} = 100$. Enfin, la métrique de performance choisie pour cette comparaison est l'*accuracy*. Elle est la seule qui est fournie et utilisée par défaut dans l'approche de MISSEL.

5.4.2 Résultats et discussion

La figure 5.8 montre la comparaison des performances de prédiction entre CASTOR-KRFE et MISSEL sur différents ensembles de données virales. En avant-propos, nous soulignons que les résultats obtenus par MISSEL sur les données de Influenza, Polyoma et Rhinovirus sont en accord avec ceux mentionnés dans leur article (Fiscon *et al.*, 2016).

Dans un premier temps, concernant, les trois premiers jeux de données (RHISPECG, INFSUBHA et INFSUBNA) portant sur les virus Rhino Influenza, les deux approches montrent des performances de prédiction proches des 100 % d'*accuracy*.

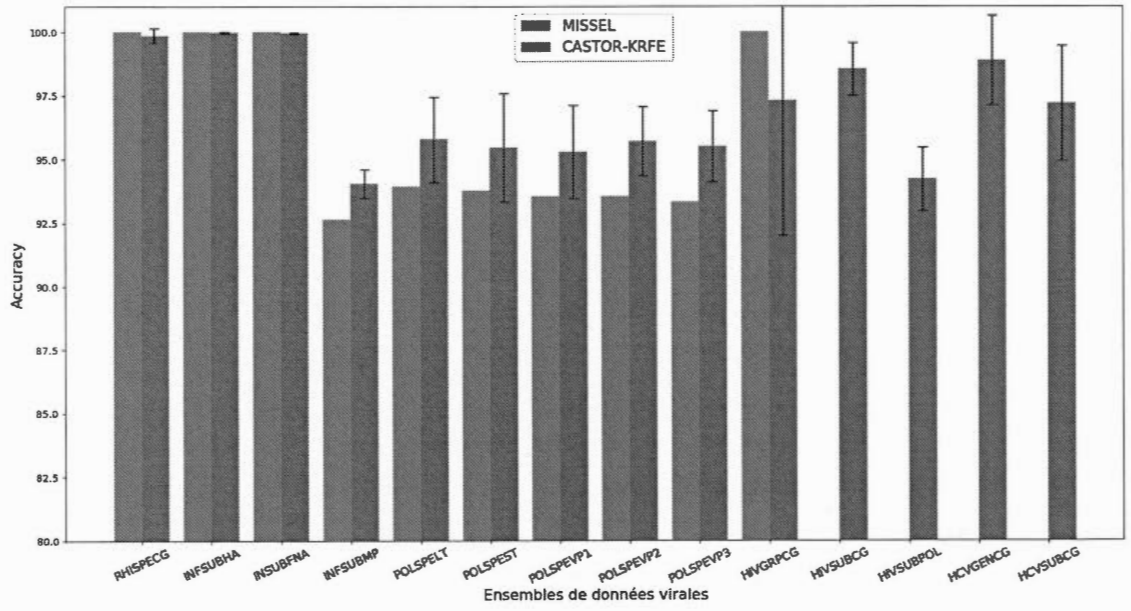

Figure 5.8 Comparaison des performances de prédiction de CASTOR-KRFE avec MISSEL

Cette figure illustre la comparaison des performances de prédiction entre MISSEL et CASTOR-KRFE sur 14 ensembles de données virales. Les barres vertes représentent la performance des meilleurs résultats obtenus par l'algorithme MISSEL. Les barres bleues représentent la performance moyenne obtenue sur 100 itérations par l'algorithme CASTOR-KRFE. Les écarts-types correspondants sont inclus. Les neuf premiers ensembles de données, en partant de la gauche, sont réalisés à partir des données MISSEL. Les cinq derniers ensembles de données, incluant quatre ensembles complexes, font partie de nos propres ensembles de données. Sur les quatre derniers ensembles de données complexes, l'algorithme MISSEL n'a pas réussi à identifier des sous-séquences discriminantes, ce qui explique l'absence de barres.

Dans un deuxième temps, en se focalisant sur les jeux de données (INFSUBMP, POLSPELT , POLSPEST, POLSPEVP1, POLSPEVP2, et POLSPEVP3), por-

tant sur les Influenza Polyoma, nous pouvons constater que les performances de prédiction de CASTOR-KRFE sont en moyenne 3% à 4% supérieures à celle de MISSEL.

Dans un troisième temps, pour le jeux de VIH-1 (HIVGRPCG) ; MISSEL obtient un score de 100% d'*accuracy* et CASTOR-KRFE obtient en moyenne 98% d'*accuracy* sur les 100 prédictions.

Enfin, à propos des quatre derniers jeux de données complexes de VIH-1 et de VHC (HIVSUBCG, HIVSUBPOL, HCVGENCG et HCVSUBCG), l'algoritme de MISSEL n'a pas été en mesure d'identifier des sous-séquences discriminantes, ce qui explique l'absence de résultats.

Ces résultats confirment que les grandes variations génétiques des génomes de virus (Duffy *et al.*, 2008) sont une limitation des algorithmes basés sur un alignement de séquences (Zielezinski *et al.*, 2017). Sur ces différents jeux de données, CASTOR-KRFE a obtenu des pourcentages moyens de bonne classification allant de 94% au minimum pour HIVSUBPOL à environ 99 % pour HCVGENCG.

5.5 Comparaison des performances de classification avec les prédicteurs spécialisés du VIH

5.5.1 Prédiction à partir de la base de données de *Los Alamos HIV-1*

Afin de pouvoir évaluer les performances de notre approche dans un cadre de classification à la fois complexe et spécialisée, nous nous sommes inspiré de l'étude comparative de CASTOR (Remita *et al.*, 2017). Cette étude a été menée contre les prédicteurs experts les plus populaires du VIH-1 (COMET (Struck *et al.*, 2014) et REGA version 2.0 (De Oliveira *et al.*, 2005)) à laquelle nous avons inclus les résultats obtenus par notre solution. La comparaison est basée sur la prédiction de

génomes complets (Figure : 5.9) ainsi que de fragments *pol* (Figure : 5.10) (partie du génome du VIH-1 utilisée par les scientifiques pour déterminer le sous-type) du VIH-1. Elle est divisée en trois sous comparaisons.

La première évalue les performances de chaque méthode sur un échantillonnage formé à partir de 10 % d'instances aléatoires de la base de données de *Los Alamos HIV-1* (`http://www.hiv.lanl.gov/`). Cet échantillonnage permet d'évaluer le comportement des prédicteurs face à des données réalistes comportant des classes inconnues. La deuxième est basée sur un partitionnement des données en fonction des sous-types spécifiques aux modèles d'entraînement de chaque prédicteur. Ce partitionnement permet de se concentrer pour chaque méthode sur leur cadre respectif pour lesquelles elles sont entraînées. Enfin, la troisième est réalisée à partir des sous-types communs. Cette dernière permet ainsi de comparer la performance des différentes méthodes sur l'intersection des sous-types entraînés par leurs modèles.

Concernant les modèles de prédiction, ceux de CASTOR-RFLP (Remita *et al.*, 2017) et de CASTOR-KRFE ont été formés à partir des jeux de données HIVSUBCG (18 classes) pour les génomes complets et HIVSUBPOL (28 classes) pour les fragments *pol* (Tableau : 4.1). Ces deux jeux de données ont été donnés en entrée à notre algorithme CASTOR-KRFE avec les paramètres suivants : $T = 1.00$, $k_{min} = 1$, $k_{max} = 30$, $f_{min} = 1$ et $f_{max} = 100$. CASTOR-KRFE a donné en sortie un modèle de prédiction pour les génomes complets basé sur 74 k-mers de longueur $k = 8$ et un modèle de prédiction pour les fragments *pol* basé sur 96 k-mers de longueur $k = 5$. Les modèles de prédiction de CASTOR-RFLP, quant à eux, sont construits à partir de 100 attributs formés depuis l'information de diverses enzymes de restriction (Remita *et al.*, 2017). Enfin, nous tenons à préciser que les approches de REGA et COMET sont fixes et ne permettent aucun changement sur les données ainsi que les classes d'entrainement. Dans notre cas actuel, les

structures d'entraînement de REGA et COMET sont basées respectivement sur 22 et 55 classes que cela soit pour les génomes complets ou bien les fragments *pol*.

5.5.2 Résultats et discussion

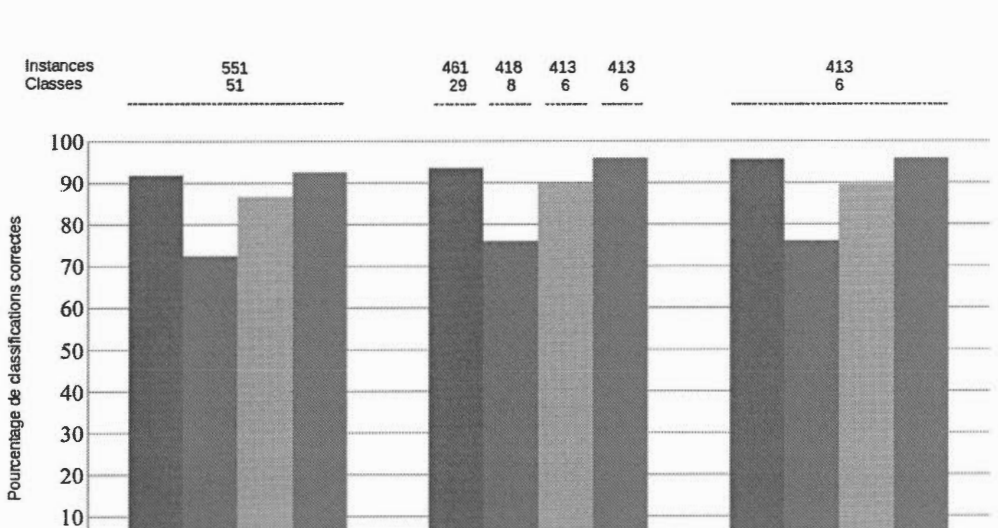

Figure 5.9 Comparaison des performances de prédiction à partir de génomes complets de VIH-1

La figure montre les pourcentages de classification correcte des génomes complets du VIH-1 sur les divers types de prédiction. Le nombre d'instances et de classes associées pour chaque échantillonnage est présenté en haut de la figure. L'échantillonnage complet correspond à 10 % des données de Los Alamos HIV-1 database sélectionnées aléatoirement. Dans le cas de l'échantillonnage de sous-types spécifiques, les prédicteurs sont évalués par rapport à leurs classes entraînées. Dans l'échantillonnage de sous-types communs, les prédicteurs sont évalués en fonction de l'intersection des classes entraînées par l'ensemble des prédicteurs.

La figure 5.9 illustre les performances de prédiction obtenues pour les différentes comparaisons portant sur les génomes complets. Pour le premier échantillonnage complet, REGA obtient les meilleures performances avec 77 % de bonne classification. CASTOR-KRFE obtient un pourcentage de bonne classification de 74% et est suivi par CATOR-RFLP et COMET avec respectivement 72 % et 68 % d'instances correctement prédites. Concernant les évaluations portant sur les sous-type spécifiques et les sous-types communs, notre approche obtient les meilleurs résultats avec plus de 99 % de prédictions correctes. Elle est suivi par REGA et CASTOR-RFLP avec respectivement 98 % et 97 % de bonne classification pour les deux évaluations. Enfin, COMET obtient les résultats les plus faibles avec 81 % de prédictions correctes sur sous-types spécifiques et 87 % sur les sous-types communs.

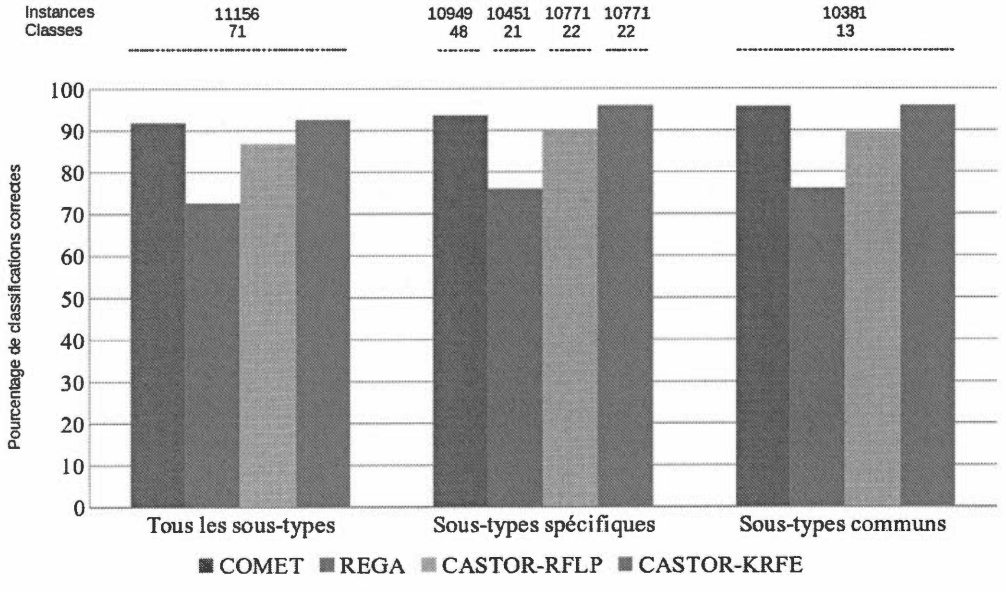

Figure 5.10 Comparaison des performances de prédiction à partir de fragments *pol* de VIH-1

La figure montre les pourcentages de classification correcte des fragments pol

du VIH-1 sur les divers types de prédiction. Le nombre d'instances et de classes associées pour chaque échantillonnage est présenté en haut de la figure. L'échantillonnage complet correspond à 10 % des données de Los Alamos HIV-1 database sélectionnées aléatoirement. Dans le cas de l'échantillonnage de sous-types spécifiques, les prédicteurs sont évalués par rapport à leurs classes entraînées. Dans l'échantillonnage de sous-types communs, les prédicteurs sont évalués en fonction de l'intersection des classes entraînées par l'ensemble des prédicteurs.

La figure 5.9 montre les performances de prédiction obtenues pour les diverses évaluations portant sur les fragments *pol*. Sur ces évaluations, CASTOR-KRFE obtient les meilleures performances avec 91 % d'instance correctement prédites sur l'échantillonage complet, 96 % sur les sous-types spécifiques et 97 % sur les sous-types communs. COMET, qui avait les résultats les plus bas sur les génomes complets, se montre très performant sur les fragments *pol* avec des performances à environ moins de 2 % de celles de CASTOR-KRFE. Pour ces expériences, CASTOR-RFLP obtient les troisièmes meilleures performances avec 87 % de bonnes instances classifiées sur l'échantillonage complet et 89 % pour les deux autres évaluations. Enfin, REGA qui avait obtenu des scores de prédiction élevés sur les génomes complets voit ses performances chuter pour la classification des fragments *pol*, obtenant des pourcentages de bonne classification entre 71 % et 76 % pour les différentes évaluations.

CHAPITRE VI

CONCLUSION ET PERSPECTIVES

Durant ce projet de maîtrise illustré à travers ce mémoire, de nombreuses compétences dans les domaines de l'informatique, de l'apprentissage automatique et de la biologie ont été acquises. L'association de ces connaissances nous a permis de concevoir CASTOR-KRFE, une nouvelle méthode indépendante de l'alignement, permettant d'extraire des attributs basés sur les sous-séquences nucléotidiques discriminantes afin d'établir des modèles de prédiction permettant de classifier les séquences génomiques virales encore inconnues.

Afin d'évaluer de manière complète et pertinente notre approche, nous avons constitué de nombreux ensembles de données depuis les bases de données virales disponibles. Sur ces derniers, nous avons réalisé diverses évaluations et comparaisons. Sur une large étendue de jeux de données virales, CASTOR-KRFE a pu extraire des ensembles d'attributs discriminants basés sur les k-mers afin de constituer des modèles de prédiction robustes qui ont été évalués à grande échelle sur plusieurs bases de données virales. Les résultats obtenus lors de la prédiction de ces grands ensembles de données, incluant des virus complexes tels que le virus de l'immunodéficience humaine (VIH) et le virus de l'hépatite C (VHC), attestent de l'efficacité des modèles prédictifs établis par CASTOR-KRFE. Notre approche a démontré à travers des études comparatives ses capacités à extraire des

motifs discriminants permettant d'obtenir des performances de prédiction supérieures à des méthodes populaires telle que MEME (Mode Discriminatif) (Bailey *et al.*, 2010), ou encore face à des approches récentes telle que MISSEL (Fiscon *et al.*, 2016). CASTOR-KRFE a aussi montré des performances de classification globales meilleures que les prédicteurs spécialisés du VIH-1 (REGA (De Oliveira *et al.*, 2005) et COMET (Struck *et al.*, 2014)), à la fois sur les génomes complets et sur les fragments *pol*.

Concernant les sous-séquences extraites par CASTOR-KRFE, de futurs analyses de ces dernières pourraient potentiellement aider à révéler des informations biologiques significatives. Ces ensembles d'attributs dérivés de k-mers pourraient aussi être utilisés pour construire des modèles de prédiction facilement compréhensibles par l'homme tels que des classificateurs à base de règles. Ce type d'approche pourrait aussi être utilisé afin de constituer des bases de données réduites des motifs discriminants dans les domaines de la virométrie et de la métagénomique. De plus, elle peut également être un atout dans le processus d'identification de biomarqueurs discriminants pour des classes spécifiques de virus.

Tel que spécifié dans (Zhang *et al.*, 2017), l'identification d'une longueur de k-mer optimale dans une méthode sans alignement est un processus difficile et important, qui reste un problème ouvert. Des méthodes comme (Wu *et al.*, 2005), (Chikhi et Medvedev, 2013) ainsi que CASTOR-KRFE, qui ont l'intention de s'attaquer à ce problème, proposent une longueur unique "optimale" pour les k-mers identifiés. Nous envisageons par la suite de faire évoluer notre approche en ajoutant une couche supplémentaire qui pourrait permettre de construire un ensemble optimal de k-mers de longueurs différentes. Ainsi, la nouvelle version pourrait obtenir une classification plus précise et pourrait représenter les propriétés biologiques de k-mers d'une meilleure façon.

Enfin, nous soulignons que l'ensemble des résultats obtenus durant notre travail de recherche nous ont mené à plusieurs publications scientifiques (voir section : PUBLICATIONS), incluant une version étendue dans *Journal of Computational Biology*, (Lebatteux *et al.*, 2019). L'algorithme CASTOR-KRFE sera prochainement inclut à la plateforme web CASTOR (Remita *et al.*, 2017) disponible à l'adresse suivante : (`http://castor.bioinfo.uqam.ca/`)

RÉFÉRENCES

Al Shalabi, L., Shaaban, Z. et Kasasbeh, B. (2006). Data mining : A preprocessing engine. *Journal of Computer Science*, *2*(9), 735–739.

Alcantara, L. C. J., Cassol, S., Libin, P., Deforche, K., Pybus, O. G., Van Ranst, M., Galvao-Castro, B., Vandamme, A.-M. et De Oliveira, T. (2009). A standardized framework for accurate, high-throughput genotyping of recombinant and non-recombinant viral sequences. *Nucleic acids research*, *37*(suppl_2), W634–W642.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W. et Lipman, D. J. (1997). Gapped blast and psi-blast : a new generation of protein database search programs. *Nucleic acids research*, *25*(17), 3389–3402.

Arnau, V., Gallach, M. et Marín, I. (2008). Fast comparison of dna sequences by oligonucleotide profiling. *BMC Research Notes*, *1*(1), 5.

Bailey, T. L., Boden, M., Buske, F. A., Frith, M., Grant, C. E., Clementi, L., Ren, J., Li, W. W. et Noble, W. S. (2009). Meme suite : tools for motif discovery and searching. *Nucleic acids research*, *37*(suppl_2), W202–W208.

Bailey, T. L., Bodén, M., Whitington, T. et Machanick, P. (2010). The value of position-specific priors in motif discovery using meme. *BMC bioinformatics*, *11*(1), 179.

Baize, S., Pannetier, D., Oestereich, L., Rieger, T., Koivogui, L., Magassouba, N., Soropogui, B., Sow, M. S., Keïta, S., De Clerck, H. *et al.* (2014). Emergence of

zaire ebola virus disease in guinea. *New England Journal of Medicine*, *371*(15), 1418–1425.

Baltimore, D. (1971). Expression of animal virus genomes. *Bacteriological reviews*, *35*(3), 235.

Bao, Y., Chetvernin, V. et Tatusova, T. (2014). Improvements to pairwise sequence comparison (pasc) : a genome-based web tool for virus classification. *Archives of virology*, *159*(12), 3293–3304.

Blaisdell, B. E. (1986). A measure of the similarity of sets of sequences not requiring sequence alignment. *Proceedings of the National Academy of Sciences*, *83*(14), 5155–5159.

Blaisdell, B. E. (1989). Effectiveness of measures requiring and not requiring prior sequence alignment for estimating the dissimilarity of natural sequences. *Journal of molecular evolution*, *29*(6), 526–537.

Blanco, R., Larrañaga, P., Inza, I. et Sierra, B. (2004). Gene selection for cancer classification using wrapper approaches. *International Journal of Pattern Recognition and Artificial Intelligence*, *18*(08), 1373–1390.

Bø, T. H. et Jonassen, I. (2002). New feature subset selection procedures for classification of expression profiles. *Genome biology*, *3*(4), research0017–1.

Bonham-Carter, O., Steele, J. et Bastola, D. (2013). Alignment-free genetic sequence comparisons : a review of recent approaches by word analysis. *Briefings in bioinformatics*, *15*(6), 890–905.

Boser, B. E., Guyon, I. M. et Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. Dans *Proceedings of the fifth annual workshop on Computational learning theory*, 144–152. ACM.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Breiman, L. (2017). *Classification and regression trees*. Routledge.

Chan, R. H., Chan, T. H., Yeung, H. M. et Wang, R. W. (2012). Composition vector method based on maximum entropy principle for sequence comparison. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, *9*(1), 79–87.

Chikhi, R. et Medvedev, P. (2013). Informed and automated k-mer size selection for genome assembly. *Bioinformatics*, *30*(1), 31–37.

Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B. *et al.* (2009). Biopython : freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, *25*(11), 1422–1423.

Cover, T. et Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), 21–27.

De Oliveira, T., Deforche, K., Cassol, S., Salminen, M., Paraskevis, D., Seebregts, C., Snoeck, J., Van Rensburg, E. J., Wensing, A. M., Van De Vijver, D. A. *et al.* (2005). An automated genotyping system for analysis of hiv-1 and other microbial sequences. *Bioinformatics*, *21*(19), 3797–3800.

Deng, M., Yu, C., Liang, Q., He, R. L. et Yau, S. S.-T. (2011). A novel method of characterizing genetic sequences : genome space with biological distance and applications. *PloS one*, *6*(3), e17293.

Díaz-Uriarte, R. et De Andres, S. A. (2006). Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, *7*(1), 3.

Duffy, S., Shackelton, L. A. et Holmes, E. C. (2008). Rates of evolutionary change in viruses : patterns and determinants. *Nature Reviews Genetics*, *9*(4), 267.

Edgar, R. C. (2004). Muscle : multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, *32*(5), 1792–1797.

Edgar, R. C. (2010). Search and clustering orders of magnitude faster than blast. *Bioinformatics*, *26*(19), 2460–2461.

Edgar, R. C. et Batzoglou, S. (2006). Multiple sequence alignment. *Current opinion in structural biology*, *16*(3), 368–373.

Fiscon, G., Weitschek, E., Cella, E., Presti, A. L., Giovanetti, M., Babakir-Mina, M., Ciotti, M., Ciccozzi, M., Pierangeli, A., Bertolazzi, P. *et al.* (2016). Missel : a method to identify a large number of small species-specific genomic subsequences and its application to viruses classification. *BioData mining*, *9*(1), 38.

Friedman, N., Geiger, D. et Goldszmidt, M. (1997). Bayesian network classifiers. *Machine learning*, *29*(2-3), 131–163.

Goodwin, S., McPherson, J. D. et McCombie, W. R. (2016). Coming of age : ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, *17*(6), 333.

Gouy, M., Guindon, S. et Gascuel, O. (2009). Seaview version 4 : a multiplatform graphical user interface for sequence alignment and phylogenetic tree building. *Molecular biology and evolution*, *27*(2), 221–224.

Guyon, I., Weston, J., Barnhill, S. et Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, *46*(1-3), 389–422.

Hall, M. A. (1999). Correlation-based feature selection for machine learning.

Halstead, S. B. (1988). Pathogenesis of dengue : challenges to molecular biology. *Science*, *239*(4839), 476–481.

Hatcher, E. L., Zhdanov, S. A., Bao, Y., Blinkova, O., Nawrocki, E. P., Ostapchuck, Y., Schäffer, A. A. et Brister, J. R. (2016). Virus variation resource–improved response to emergent viral outbreaks. *Nucleic acids research*, *45*(D1), D482–D490.

Hesper, B. et Hogeweg, P. (1970). Bioinformatica : een werkconcept. *Kameleon*, *1*(6), 28–29.

Holland, J. (1975). Adaptation in natural and artificial systems : an introductory analysis with application to biology. *Control and artificial intelligence.*

Hsu, C.-W., Chang, C.-C., Lin, C.-J. *et al.* (2003). A practical guide to support vector classification.

Hunt, E. B., Marin, J. et Stone, P. J. (1966). Experiments in induction.

Jafari, P. et Azuaje, F. (2006). An assessment of recently published gene expression data analyses : reporting experimental design and statistical factors. *BMC Medical Informatics and Decision Making*, *6*(1), 27.

Jolliffe, I. (2011). Principal component analysis. In *International encyclopedia of statistical science* 1094–1096. Springer.

Jong, K., Marchiori, E., Sebag, M. et Van Der Vaart, A. (2004). Feature selection in proteomic pattern data with support vector machines. Dans *Computational Intelligence in Bioinformatics and Computational Biology, 2004. CIBCB'04. Proceedings of the 2004 IEEE Symposium on*, 41–48. IEEE.

Kolekar, P., Kale, M. et Kulkarni-Kale, U. (2012). Alignment-free distance measure based on return time distribution for sequence analysis : applications to

clustering, molecular phylogeny and subtyping. *Molecular phylogenetics and evolution*, *65*(2), 510–522.

Koller, D. et Sahami, M. (1996). *Toward optimal feature selection.* Rapport technique, Stanford InfoLab.

Kotsiantis, S. B., Zaharakis, I. et Pintelas, P. (2007). Supervised machine learning : A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, *160*, 3–24.

Kuiken, C., Thurmond, J., Dimitrijevic, M. et Yoon, H. (2011). The lanl hemorrhagic fever virus database, a new platform for analyzing biothreat viruses. *Nucleic acids research*, *40*(D1), D587–D592.

Kuiken, C., Yusim, K., Boykin, L. et Richardson, R. (2004). The los alamos hepatitis c sequence database. *Bioinformatics*, *21*(3), 379–384.

Larranaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J. A., Armañanzas, R., Santafé, G., Pérez, A. *et al.* (2006). Machine learning in bioinformatics. *Briefings in bioinformatics*, *7*(1), 86–112.

Lauber, C. et Gorbalenya, A. E. (2012). Partitioning the genetic diversity of a virus family : approach and evaluation through a case study of picornaviruses. *Journal of virology*, JVI–07173.

Lebatteux, D., Remita, A. M. et Diallo, A. B. (2019). Toward an alignment-free method for feature extraction and accurate classification of viral sequences. *Journal of Computational Biology.*

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J. et Liu, H. (2017). Feature selection : A data perspective. *ACM Computing Surveys (CSUR)*, *50*(6), 94.

Li, M., Badger, J. H., Chen, X., Kwong, S., Kearney, P. et Zhang, H. (2001). An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics*, *17*(2), 149–154.

Liu, H. et Setiono, R. (1995). Chi2 : Feature selection and discretization of numeric attributes. Dans *Tools with artificial intelligence, 1995. proceedings., seventh international conference on*, 388–391. IEEE.

Liu, X., Wan, L., Li, J., Reinert, G., Waterman, M. S. et Sun, F. (2011). New powerful statistics for alignment-free sequence comparison under a pattern transfer model. *Journal of theoretical biology*, *284*(1), 106–116.

Liu, Z., Meng, J. et Sun, X. (2008). A novel feature-based method for whole genome phylogenetic analysis without alignment : application to hev genotyping and subtyping. *Biochemical and biophysical research communications*, *368*(2), 223–230.

Lu, G., Zhang, S. et Fang, X. (2008). An improved string composition method for sequence comparison. *BMC bioinformatics*, *9*(6), S15.

Luscombe, N. M., Greenbaum, D. et Gerstein, M. (2001). What is bioinformatics ? a proposed definition and overview of the field. *Methods of information in medicine*, *40*(04), 346–358.

Mahmoudabadi, G. et Phillips, R. (2018). A comprehensive and quantitative exploration of thousands of viral genomes. *eLife*, *7*, e31955.

Matsen, F. A., Kodner, R. B. et Armbrust, E. V. (2010). pplacer : linear time maximum-likelihood and bayesian phylogenetic placement of sequences onto a fixed reference tree. *BMC bioinformatics*, *11*(1), 538.

Mehmood, M. A., Sehar, U. et Ahmad, N. (2014). Use of bioinformatics tools in different spheres of life sciences. *Journal of Data Mining in Genomics & Proteomics*, *5*(2), 1.

Müller, A. C., Guido, S. *et al.* (2016). *Introduction to machine learning with Python : a guide for data scientists.* " O'Reilly Media, Inc.".

Nalbantoglu, O. U., Way, S. F., Hinrichs, S. H. et Sayood, K. (2011). Raiphy : phylogenetic classification of metagenomics samples using iterative refinement of relative abundance index profiles. *BMC bioinformatics*, *12*(1), 41.

Otu, H. H. et Sayood, K. (2003). A new sequence distance measure for phylogenetic tree construction. *Bioinformatics*, *19*(16), 2122–2130.

Ounit, R., Wanamaker, S., Close, T. J. et Lonardi, S. (2015). Clark : fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. *BMC genomics*, *16*(1), 236.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. *et al.* (2011). Scikit-learn : Machine learning in python. *Journal of machine learning research*, *12*(Oct), 2825–2830.

Pickett, B. E., Sadat, E. L., Zhang, Y., Noronha, J. M., Squires, R. B., Hunt, V., Liu, M., Kumar, S., Zaremba, S., Gu, Z. *et al.* (2011). Vipr : an open bioinformatics database and analysis resource for virology research. *Nucleic acids research*, *40*(D1), D593–D598.

Pierangeli, A., Ciccozzi, M., Chiavelli, S., Concato, C., Giovanetti, M., Cella, E., Spano, L., Scagnolari, C., Moretti, C., Papoff, P. *et al.* (2013). Molecular epidemiology and genetic diversity of human rhinovirus affecting hospitalized children in rome. *Medical microbiology and immunology*, *202*(4), 303–311.

Pond, S. L. K., Posada, D., Stawiski, E., Chappey, C., Poon, A. F., Hughes, G., Fearnhill, E., Gravenor, M. B., Brown, A. J. L. et Frost, S. D. (2009). An evolutionary model-based algorithm for accurate phylogenetic breakpoint mapping and subtype prediction in hiv-1. *PLoS computational biology*, *5*(11), e1000581.

Quinlan, J. R. (2014). *C4. 5 : programs for machine learning*. Elsevier.

Reinert, G., Chew, D., Sun, F. et Waterman, M. S. (2009). Alignment-free sequence comparison (i) : statistics and power. *Journal of Computational Biology*, *16*(12), 1615–1634.

Remita, M. A., Halioui, A., Diouara, A. A. M., Daigle, B., Kiani, G. et Diallo, A. B. (2017). A machine learning approach for viral genome classification. *BMC bioinformatics*, *18*(1), 208.

Ren, J., Ahlgren, N. A., Lu, Y. Y., Fuhrman, J. A. et Sun, F. (2017). Virfinder : a novel k-mer based tool for identifying viral sequences from assembled metagenomic data. *Microbiome*, *5*(1), 69.

Saeys, Y., Inza, I. et Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *bioinformatics*, *23*(19), 2507–2517.

Sandve, G. K. et Drabløs, F. (2006). A survey of motif discovery methods in an integrated framework. *Biology direct*, *1*(1), 11.

sangkawibha, n., rojanasuphot, s., ahandrik, s., viriyapongse, s., jatanasen, s., salitul, v., phanthumachinda, b. et halstead, s. b. (1984). Risk factors in dengue shock syndrome : a prospective epidemiologic study in rayong, thailand : I. the 1980 outbreak. *American journal of epidemiology*, *120*(5), 653–669.

Silva, J. C. F., Carvalho, T. F., Basso, M. F., Deguchi, M., Pereira, W. A., Sobrinho, R. R., Vidigal, P. M., Brustolini, O. J., Silva, F. F., Dal-Bianco, M.

et al. (2017a). Geminivirus data warehouse : a database enriched with machine learning approaches. *BMC bioinformatics*, *18*(1), 240.

Silva, J. C. F., Carvalho, T. F., Fontes, E. P. et Cerqueira, F. R. (2017b). Fangorn forest (f2) : a machine learning approach to classify genes and genera in the family geminiviridae. *BMC bioinformatics*, *18*(1), 431.

Simmonds, P., Bukh, J., Combet, C., Deléage, G., Enomoto, N., Feinstone, S., Halfon, P., Inchauspé, G., Kuiken, C., Maertens, G. *et al.* (2005). Consensus proposals for a unified system of nomenclature of hepatitis c virus genotypes. *Hepatology*, *42*(4), 962–973.

Sims, G. E., Jun, S.-R., Wu, G. A. et Kim, S.-H. (2009). Alignment-free genome comparison with feature frequency profiles (ffp) and optimal resolutions. *Proceedings of the National Academy of Sciences*, pnas–0813249106.

Skalak, D. B. (1994). Prototype and feature selection by sampling and random mutation hill climbing algorithms. In *Machine Learning Proceedings 1994* 293–301. Elsevier.

Soares, I., Goios, A. et Amorim, A. (2012). Sequence comparison alignment-free approach based on suffix tree and l-words frequency. *The Scientific World Journal*, *2012*.

Struck, D., Lawyer, G., Ternes, A.-M., Schmit, J.-C. et Bercoff, D. P. (2014). Comet : adaptive context-based modeling for ultrafast hiv-1 subtype identification. *Nucleic acids research*, *42*(18), e144–e144.

Taylor, B. S., Sobieszczyk, M. E., McCutchan, F. E. et Hammer, S. M. (2008). The challenge of hiv-1 subtype diversity. *New England Journal of Medicine*, *358*(15), 1590–1602.

Thompson, J. D., Higgins, D. G. et Gibson, T. J. (1994). Clustal w : improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, *22*(22), 4673–4680.

Ulitsky, I., Burstein, D., Tuller, T. et Chor, B. (2006). The average common substring approach to phylogenomic reconstruction. *Journal of Computational Biology*, *13*(2), 336–350.

Van Belkum, A., Struelens, M., de Visser, A., Verbrugh, H. et Tibayrenc, M. (2001). Role of genomic typing in taxonomy, evolutionary genetics, and microbial epidemiology. *Clinical microbiology reviews*, *14*(3), 547–560.

van den Berg, R. A., Hoefsloot, H. C., Westerhuis, J. A., Smilde, A. K. et van der Werf, M. J. (2006). Centering, scaling, and transformations : improving the biological information content of metabolomics data. *BMC genomics*, *7*(1), 142.

Vapnik, V. N. et Chervonenkis, A. Y. (2015). On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity* 11–30. Springer.

Vaughn, D. W., Green, S., Kalayanarooj, S., Innis, B. L., Nimmannitya, S., Suntayakorn, S., Endy, T. P., Raengsakulrach, B., Rothman, A. L., Ennis, F. A. *et al.* (2000). Dengue viremia titer, antibody response pattern, and virus serotype correlate with disease severity. *The Journal of infectious diseases*, *181*(1), 2–9.

Verikas, A., Vaiciukynas, E., Gelzinis, A., Parker, J. et Olsson, M. C. (2016). Electromyographic patterns during golf swing : Activation sequence profiling and prediction of shot effectiveness. *Sensors*, *16*(4), 592.

Vinga, S. (2014). Alignment-free methods in computational biology.

Vinga, S. et Almeida, J. (2003). Alignment-free sequence comparison—a review. *Bioinformatics*, *19*(4), 513–523.

Wang, J.-D. (2011). A comparison study of virus classification by genome sequences. Dans *Bioinformatics and Bioengineering (BIBE), 2011 IEEE 11th International Conference on*, 270–273. IEEE.

Wen, J., Chan, R. H., Yau, S.-C., He, R. L. et Yau, S. S. (2014). K-mer natural vector and its application to the phylogenetic analysis of genetic sequences. *Gene*, *546*(1), 25–34.

Williams, R. C. (1989). Restriction fragment length polymorphism (rflp). *American Journal of Physical Anthropology*, *32*(S10), 159–184.

Wong, K. M., Suchard, M. A. et Huelsenbeck, J. P. (2008). Alignment uncertainty and genomic analysis. *Science*, *319*(5862), 473–476.

Wood, D. E. et Salzberg, S. L. (2014). Kraken : ultrafast metagenomic sequence classification using exact alignments. *Genome biology*, *15*(3), R46.

Wu, T.-J., Huang, Y.-H. et Li, L.-A. (2005). Optimal word sizes for dissimilarity measures and estimation of the degree of dissimilarity between dna sequences. *Bioinformatics*, *21*(22), 4125–4132.

Xing, Z., Pei, J. et Keogh, E. (2010). A brief survey on sequence classification. *ACM Sigkdd Explorations Newsletter*, *12*(1), 40–48.

Yu, C., Hernandez, T., Zheng, H., Yau, S.-C., Huang, H.-H., He, R. L., Yang, J. et Yau, S. S.-T. (2013). Real time classification of viruses in 12 dimensions. *PloS one*, *8*(5), e64328.

Yu, C., Liang, Q., Yin, C., He, R. L. et Yau, S. S.-T. (2010a). A novel construction of genome space with biological geometry. *DNA research*, *17*(3), 155–168.

Yu, Z.-G., Chu, K. H., Li, C. P., Anh, V., Zhou, L.-Q. et Wang, R. W. (2010b). Whole-proteome phylogeny of large dsdna viruses and parvoviruses through a composition vector method related to dynamical language model. *BMC evolutionary biology*, *10*(1), 192.

Zhang, Q., Jun, S.-R., Leuze, M., Ussery, D. et Nookaew, I. (2017). Viral phylogenomics using an alignment-free method : A three-step approach to determine optimal length of k-mer. *Scientific reports*, *7*, 40712.

Zielezinski, A., Vinga, S., Almeida, J. et Karlowski, W. M. (2017). Alignment-free sequence comparison : benefits, applications, and tools. *Genome biology*, *18*(1), 186.