République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Mohammed El Bachir El Ibrahimi B.B.A
Faculté de Mathématiques et d'informatique
Département Informatique

MEMOIRE

Présente en vue de l'obtention du diplôme

Master en informatique

Spécialité : Ingénierie de l'informatique décisionnelle

THEME

La recherche et la décontamination de séquences contaminées dans un assemblage De Novo par un modèle basé sur la classification supervisée

Présenté par :

BENNIA Anes Chems Eddine

Dirigé par :

MAACHE Salah

Promotion : 2020/2021

DEDICATION

À mes parents

Pour leur soutien inconditionnel dans tout ce que j'ai entrepris et que j'ai pu réussir grâce à eux, qui ont sacrifié des années de leurs vies. Vos prières et vos bénédictions m'ont été d'un grand secours pour mener à bien mes études

Puisse Dieu, le tout puissant, vous préserve et vous accorde santé, longue vie et bonheur

À mes frères Yacine, Khalil, sohaib et Hammem

Les mots ne suffisent pas pour exprimer l'attachement, l'amour et l'affectation que je porte pour vous

BENNIA Anes Chems Eddine

REMERCIEMENT

Je rends grâce à dieu Tout-Puissant qui m'a permis d'être ici aujourd'hui.
Toutes les personnes ayant permis de mener à bien ce travail sont assurés de ma gratitude.
Mes sincères remerciements s'adressent tout d'abord à mon promoteur, monsieur, ***MAACHE Salah****, qui a su m'en encadrer avec efficacité, je vous exprime mes profonde reconnaissance pour votre soutien et votre encouragement.*
Je ne saurais terminer mes remerciements sans une pensée à ma famille qui a toujours cru en moi, ainsi que mes amis et à tous ceux qui ont participé de près ou de loin.

Résumé :
Le problème traité dans ce mémoire s'inscrit dans le cadre de la décontamination des séquences d'ADN, l'objectif est de concevoir un modèle de décontamination basée sur la classification supervisée. L'approche proposée consiste en une classification (cible ou contaminant) après avoir extrait certains attributs à savoir le groupe d'attributs IMM et K-gram. Après l'étude de performances, les machines à vecteurs supports (SVM) ont été plus performantes que le KNN et les arbres de décision. L'évaluation du modèle est faite à partir des expérimentations sur les séquences issues d'un sequençage à haut débit en utilisant le simulateur MetaSim .

Mots clés :Décontamination des séquences d'ADN, classification supervisée, SVM, KNN, arbres de décision, K-gram, IMM,sequençage à haut débit, MetaSim.

Abstract :
The problem addressed in this thesis falls within the framework of the decontamination of DNA sequences, the objective is to design a decontamination model based on the supervised classification . The proposed approach consists of a classification (target or contaminant) after extracting some attributes namely the IMM and K-gram attribute group. After the study performance, support vector machines (SVM) have been more efficient than the KNN and decision trees. The evaluation of the model is made from experiments on high-throughput sequencing's sequences, using the MetaSim simulator.

Key-words : Decontamination of DNA sequences, supervised classification, SVM, KNN, decision trees, K-gram, IMM,high-throughput sequencing, MetaSim..

ملخص:

يندرج العمل المنجز في هذه الأطروحة ضمن ميدان إزالة تلوث سلاسل الحمض النووي، الهدف هو تصميم نموذج لإزالة التلوث يعتمد على تقنيات التعلم الخاضع للإشراف ، النهج المقترح في تصنيف سلاسل الحمض النووي (هدف أو ملوث) بعد استخراج مجموعة من السمات IMM وK-gram ، يعتمد على استعمال أجهزة المتجهات الاعتمادية SVM و KNN و أشجار القرار، بعد دراسة كفاءة النماذج، تبين أن SVM أكثر كفاءة و ذلك اعتمادا على سلاسل الحمض النووي الناتجة من أداة محاكات تسلسل عالي السرعة MetSim.

كلمات مفتاحية : إزالة تلوث سلاسل الحمض النووي ، التعلم الخاضع للإشراف، أجهزة المتجهات الاعتمادية، أشجار القرار، تسلسل عالي السرعة .

TABLE DES MATIÈRES

Table des figures

Liste des tableaux

LISTE DES ABRÉVIATIONS

ADN Acide DésoxyriboNucléique
ARN AcideRiboNucléique
BLAST Basic Local Alignment Search Tool
SVM Support Vector Machineknn
KNN K plus proches voisins(K-nearest neighbors)
NGS Next Generation Sequencing
PCR Polymerase Chain Reaction
HTS séquençage à haut débit
TGS third generation sequencing
OLC Overlap-Layout-Consensus
IA Intelligence artificielle ou AI en anglais pour Artificial Intelligence
IBM International Business Machines
RBF Radial Basis Functions
NCBI National Center for Biotechnology Information
IMM Multivariate Mutual Information
BWA Burrows-Wheeler Aligner
WR White list Ratio

INTRODUCTION

0.1 Introduction

La compréhension de l'organisation du vivant et de sa diversité a toujours été un sujet scientifique important. C'est un domaine central en biologie qui consiste à la classification et la taxonomie des organismes vivants.

Les organismes vivants sont classés au début en se basant sur les caractéristiques morphologiques observés à l'œil nu. La découverte de l'ADN et l'apparition des premières technologies bio-informatiques de séquençage et d'assemblage au XXe siècle ont initié à une révolution du domaine. Ces avancées technologiques permettent une classification plus précise et plus pointus des organismes vivants en se basant sur l'analyse de leurs ADN.

L'assemblage de novo, la reconstruction de zéro de l'ADN des organismes vivants est un sujet particulièrement difficile pour lequel il n'existe pas encore des solutions satisfaisantes, en effet, les données à assembler (les séquences ADN issues d'un séquençage à haut débit) comportent des similaires entre les différents organismes, des redondances de séquences et un problème majeur qui est la contamination des séquences qui est manifesté par la présence des séquences d'un organisme contaminant l'organisme intérêt (l'organisme cible qu'on veut étudier).

Pour un échantillon contaminé, où il existe plusieurs organismes vivant en plus de l'organisme cible sont présents, la recherche et la décontamination de séquences contaminées dans un assemblage de novo est primordiale, elle s'agit de l'ensemble des méthodes et des techniques informatiques qui ont pour objectif d'isoler les séquences de l'organisme cible des autres séquences issues de l'organisme contaminant.

Dans la littérature, de nombreuses méthodes et techniques ont été proposées pour résoudre le problème. On a réalisé une taxonomie des méthodes de décontamination.Cette taxonomie devise les méthodes en deux classes. Les méthodes à base d'alignement des transcrits sur des bases de données, elles utilisent les alignements sur des séquences d'organismes connus pour prédire si un transcrit provient d'un contaminant ou non (Deconseq, BWA, BLAST . . . etc.).

Les méthodes à basées sur les techniques d'apprentissage automatique(Arbre de décision, SVM, KNN et le clustering), elles utilisent des data sets (ensemble des séquences de différents organismes) pour construire un modèle de prédiction, par la suite l'utiliser pour prédire si une nouvelle séquence provient d'un contaminant ou non.

0.2 Motivation du travail

L'identification des séquences contaminants dans un assemblage de novo est un défi en raison de l'absence d'informations sur les espèces cibles. En plus de la contamination par de mauvaises manipulations en laboratoire, les types d'échantillons où l'organisme cible est impossible à isoler de sa matrice, comme les endoparasites, les endosymbiontes et les échantillons prélevés dans le sol, la contamination est inévitable.

La motivation du travail est de faire face au problème de contamination dans un assemblage De Novo en développant une approche pour rechercher et décontaminer une séquence ADN

à l'aide des méthodes de Machine Learning (apprentissage supervisé ou non supervisé), sans avoir recours aux bases de données de référence.

0.3 Processus de décontamination dans un assemblage de novo

Lors de l'étude d'un organisme qui est infecté par un autre organisme, un ensemble de techniques et de méthodes de décontamination de séquences contaminées sont obligatoires dans le but de filtrer les séquences de l'organisme cible afin de pouvoir reconstruire l'ADN de ce dernier.

La procédure de décontamination de séquences contaminées dans un assemblage de novo est effectuée en trois étapes suivantes :

1. Préparer le data set : la préparation du data set est un facteur important qui consiste à récolter les informations pertinentes pour chaque fragment d'ADN ;
2. Construire le modèle d'apprentissage : appliquer les différents modèles d'apprentissage sur le data set préparé au préalable pour prédire et filtrer les fragments d'ADN par la suite ;
3. Filtrer les fragments d'ADN : utiliser le modèle d'apprentissage construit pour séparer les fragments d'ADN en deux classes, un groupe pour l'organisme cible, et un groupe pour les organismes contaminants, ensuite éliminer ces derniers et garder l'organisme cible.

0.4 Objectif du travail

le travail fournit a pour objectif deux éléments :

1. Réaliser un panorama synthétique et organisé des travaux déjà réalisés sur le problème de la recherche et la décontamination de séquences dans un assemblage De Novo à savoir les méthodes, les approches et les algorithmes utilisés ;
2. Développer une nouvelle méthode ou améliorer une méthode déjà existée pour la recherche et la décontamination de séquences contaminées dans un assemblage De Novo.

0.5 Organisation du manuscrite

Le manuscrit comprend cinq chapitres précédés par une introduction générale, il termine par une conclusion générale et perspectives.

Dans l'introduction générale, j'ai introduit le contexte du travail, et présenté le problème d'une manière générale ainsi les motivations et les objectifs de ce travail.

— Dans le premier chapitre, j'ai introduit les concepts biologiques requises pour bien maîtriser la problématique.

— Le deuxième chapitre s'agit des méthodes Bio-Informatiques utilisées pour résoudre le problème de la recherche et de la décontamination de séquences dans un assemblage De Novo (séquençage, assemblage … etc.).

— Le troisième chapitre représente un état de l'art, dont lequel j'ai présenté les différents travaux réalisés concernant le problème.

— Le quatrième chapitre détaille le travail réalisé, les méthodes d'apprentissage utilisées.

— Le dernier chapitre est la validation expérimentale du travail et l'étude de performances.

La thèse est clôturée par une conclusion générale et les perspectifs attendus dans le futur pour améliorer ce modeste travail.

Chapitre

1 CONTEXTE BIOLOGIQUE

Sommaire

1.1 Introduction

1.2 Introduction à la biologie environnementale

1.2.1 Perspective historique

Le besoin d'étudier l'environnement naturel, ainsi les organismes qui vivent au sein de cet environnement remonte aux origines de la civilisation humaine. Réellement, l'étude de caractéristiques macroscopiques des organismes (animaux et plantes), a été considérée comme un domaine indispensable à la survie des êtres humains dans un environnement qu'ils ont occupé. D'ailleurs les premières tentatives de classification remontent aux civilisations égyptienne et grecque, en passant par le Moyen âge. Ces classifications principalement concentrent sur

la description, surtout la description d'animaux et de plantes utiles dans l'agriculture ou la médecine [1] .

L'étude de caractéristiques microscopiques des organismes vivants a eu lieu pour la première fois, en 1668 grâce à Antoine van Leeuwenhoek qui observe ces caractéristiques par le moyen d'un microscope de son invention. Il appelle ces organismes vivant observés animalcules ; le terme bactéries, utilisé aujourd'hui dérivant du grec pour « bâtonnet » a fait son apparition en 1838 [1] .

En 1859, Louis Pasteur, avec ses travaux sur les micro-organismes ont donné naissance à la microbiologie, notamment la découverte des mécanismes de réplication, les méthodes de destruction des micro-organismes [1] .

C'est à la fin du 19ém, les premières classifications de bactéries sont apparues, suit à l'invention de la coloration de Gram en 1884, qui partitionne les bactéries en deux partitions, bactéries Gram positifs et Gram négatifs, en se basant sur les propriétés de la paroi bactérienne [1] .

Dans les années 1925, Edouard Chatton propose sa classification des organismes cellulaires à savoir : les procaryotes (cellule sans noyau), et les eucaryotes (cellule avec noyau) [1] .

En 1977, Carl Woese propose sa classification des organismes vivants en fonction des caractères moléculaires. Grâce à l'analyse de l'ARN ribosomique 16S, il partitionne les procaryotes en deux groupes : les Eubacteria et les Archaebacteria [2] .

Cette classification phylogénétique couvre rapidement les eucaryotes, par l'utilisation de l'ARN ribosomique 18S, cette classification est l'approche de référence pour la classification des organismes vivants [2] .

1.2.2 L'ADN

L'acide désoxyribonucléique, ou ADN, est un acide nucléique présent pratiquement dans tous les organismes vivants. L'ADN contient toute l'information génétique, appelée génome, permettant le développement, le fonctionnement et la reproduction des êtres vivants [3] .

L'ADN est constitué d'un polymère de nucléotides, les adénines (A), thymines (T), guanines (G), et cytosines (C), liés par une liaison phosphodiester. L'ADN représenté dans la figure 1.1, est formé de deux brins antiparallèles enroulés l'un autour de l'autre pour former une double hélice, l'adénine (A) s'apparie avec la thymine (T) au moyen de deux liaisons hydrogène, et la guanine (G) avec la cytosine (C) au moyen de trois liaisons hydrogènes [3] .

Les deux brins sont dits complémentaires cela veut dire qu'il est possible de déduire un brin à partir de l'autre.

1.2.3 L'ARN

L'acide ribonucléique ou ARN est un acide nucléique présent pratiquement dans tous les organismes vivants. L'ARN est très similaire chimiquement à l'ADN, généralement synthétisé dans les cellules à partir de l'ADN, il est considéré comme une copie de l'ADN [3] .

Les quatre principales bases de l'ARN, sont l'adénine (A), l'uracile (U), la cytosine (C) et la guanine (G). Par rapport à l'ADN, la thymine de l'ADN est remplacée par l'uracile dans l'ARN.

1.2.4 Gène et Génome

Un gène est une séquence discrète et héritable de nucléotides dont l'expression affecte les caractères d'un organisme vivant. L'ensemble des gènes d'un organisme constitue son génome [3] .

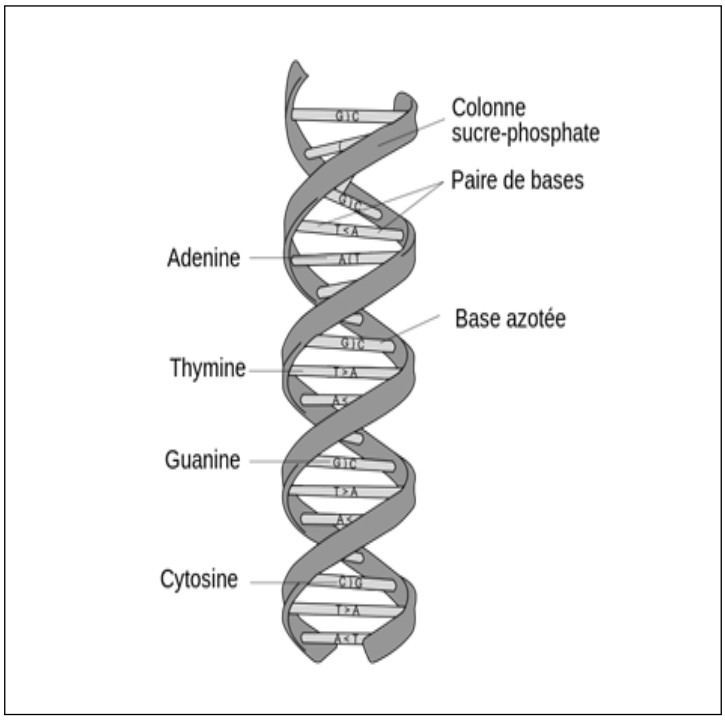

FIGURE 1.1: Structure de la double hélice d'ADN [12]

Un gène possède donc une position donnée dans le génome d'une espèce, on parle de locus génique.

La transmission des allèles des gènes des individus parents à leur descendance est à l'origine de l'héritabilité des caractères phénotypiques .

1.2.5 Classification (Taxonomie) du vivant

La classification du vivant est le processus d'assigner à chaque organisme vivant une catégorie, en se basant sur des critères scientifiques, dans le but de retracer leur histoire évolutive.

Dans un arbre phylogénétique, les organismes les plus proches dérivent des mêmes ancêtres et partagent le plus de caractères en commun. On distingue deux types de classification selon les critères choisis pour classer les organismes, une classification classique basée sur des critères physiques, une classification phylogénétique basée sur des critères biomoléculaires [5] .

1.2.5.1 Classification classique

Cette classification est initié par un naturaliste Carl von Linné au 19ém siècle, elle propose l'organisation des organismes vivants dans un arbre à sept niveaux consécutifs (rangs de taxons), l'espèce est considéré comme l'unité Base dans cette classification , puis le genre, la famille, l'ordre, la classe, l'embranchement, le règne et le domaine [6] .

Elle repose sur la présence ou l'absence de caractères multiples (biologiques, phénotypiques, physiologiques).

Actuellement, la classification traditionnelle est telle que six règnes divisent le monde vivant (bactéries, archées, protistes, champignons, végétaux, animaux).

1.2.5.2 Classification phylogénétique

Suite aux évolutions de la biologie moléculaire et de la génétique, au cours de la deuxième moitié du 20 ém siècle, le microbiologiste Carl Woese a créé une classification phylogénétique basée l'analyse phylogénétique de l'ARNr 16S [2] .

Cette Classification a donné naissance à trois grands domaines :

— **Les eucaryotes** : représentent les organismes vivants visibles à l'œil nu. Par exemple, les animaux, les champignons, les plantes, aussi les organismes pluricellulaires comme les algues rouges et brunes, ou les organismes unicellulaires. Les eucaryotes sont caractérisés par la présence dans leurs cellules d'un vrai noyau.

— **Les bactéries** : Représentent les organismes unicellulaires sans noyau. Les bactéries se trouvent dans tous les environnements terrestres. Par exemple, les bactéries des genres Bacteroides, Prevotella, Ruminococcus.

— **Les archées** : considérées au début comme des bactéries extrêmophiles, suite des travaux de Carl Woese en 1977, les archées sont classées dans un domaine à part.

1.2.6 Les enjeux

L'étude et la compréhension des organismes vivants est actuellement un enjeu majeur dans différents domaines de recherche :

— En santé, la connaissance des caractéristiques micros biologiques des organismes vivants me permet de mieux comprendre leurs impacts pour le maintien en bonne santé, la découverte de maladies ;

— En industrie pharmaceutique, analyser massivement les nouveaux gènes induira à la découverte de nouvelles protéines et molécules utilisées par la suite dans des médicaments ou des catalyseurs ;

— En écologie, la compréhension des organismes vivants, leurs évolutions, leurs interactions permettront de mieux comprendre l'impact de différents facteurs externes sur l'environnement en question, cela me donne la possibilité de mieux réagir face à ces menaces

1.3 Séquençage de l'ADN

L'analyse du matériel génétique est devenue une étape primordiale pour comprendre un organisme vivant. L'acquisition de ce matériel passe obligatoirement par une étape de séquençage, terme qui désigne le processus permettant de lire les séquences d'ADN. Pour cela, plusieurs procédés existent, qui ont progressé au fil du temps. Dans cette section on va présenter, d'une manière générale le processus de séquençage.

1.3.1 Définitions

Le séquençage de l'ADN est un processus qui consiste à déterminer l'ordre des nucléotides (A, C, G, T) d'un fragment d'ADN. On appelle lecture (read) de séquençage la séquence d'un fragment d'ADN lue par un séquenceur [7] .

Dans le cadre du séquençage à haut débit, on appelle librairie l'ensemble des fragments d'ADN issus de la préparation biomoléculaire du matériel génétique, obtenus avec différentes stratégies possibles [7] .

Un run de séquençage s'agit de l'expérience de séquençage d'une librairie donnée, en une seule fois, sur le même séquenceur [6] .

Généralement lors de processus de séquençage, des erreurs de séquençage se présentent, ils s'agissent des différences entre la séquence d'une lecture et la séquence originale du fragment d'ADN.

Les erreurs de séquençage représentées dans la figure 1.2 sont du type substitution (transformation d'un nucléotide en un autre dans la lecture de séquençage), ou du type InDel (insertion ou délétion d'un nucléotide dans la lecture).

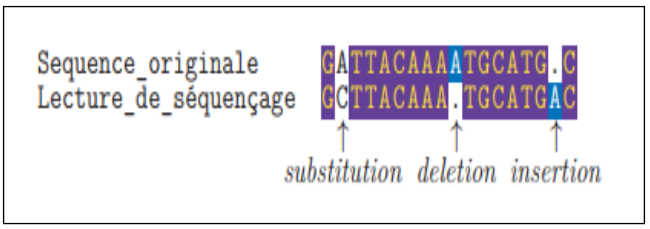

FIGURE 1.2: Les 3 types d'erreurs de séquençage

1.3.2 Perspective historique

Au début des années 1970 l'équipe de Walter Gilbert, aux États-Unis a inventée la première méthode de séquençage d'ADN, une méthode par dégradation chimique sélective. Cependant la méthode par synthèse enzymatique inventée par Frederick Sanger en 1977 au Royaume-Uni, ainsi qui porte son nom, méthode de Sanger, qui posa les bases technologiques du séquençage moderne [8] .

La méthode de Sanger, commercialisée en premier lieu par Applied Biosystems, pendant environ 40 ans, elle est demeurée la méthode la plus utilisée. La méthode de Sanger repose sur le processus naturel de réplication de l'ADN, elle repose sur la polymérisation contrôlée de simple brin de l'ADN par des enzymes ADN polymérases. Grâce à l'introduction de nucléotides modifiés (didésoxyribonucléotides), la polymérisation se termine prématurément de manière aléatoire.

La séquence complète peut être ensuite lue sur un gel ayant servi à faire migrer les produits de la polymérisation selon leur taille.

A la fin des années 1990, les premières technologies de séquençage à haut débit dites de « nouvelle génération» (NGS pour Next Generation Sequencing) sont apparues, elles en permettent de séquencer de grandes quantités d'ADN en des temps record à moindre coût. Cette méthode représente une révolution énorme des méthodes de séquençage, à titre d'exemple le projet human génome a coûté 3 milliards de dollars sur 13 ans entre 1990 et 2003 pour séquencer le génome humain en utilisant des séquenceurs de type Sanger répartis dans plusieurs laboratoires à travers le monde. Aujourd'hui, avec un séquenceur NGS Illumina HiSeq X, en trois jours, on peut séquencer trois génomes humains pour 1000 dollars chacun [9] .

Pour finir, les séquenceurs de troisième génération, les plus récents, Ce sont des séquenceurs capables de générer de très longs lectures sans avoir besoin de cloner les fragments pour amplifier le signal. C'est pour cette raison qu'on les appelle aussi "Single molecule sequencing", en revanche, ces nouvelles techniques produisent encore beaucoup d'erreurs de séquençage [10] .

1.3.3 Séquençage de Sanger

La méthode de Sanger automatisée, détaillée dans la figure 1.3, est considérée comme étant la méthode de séquençage haut-débit de première génération.

Le séquençage Sanger d'un fragment d'ADN nécessite d'abord de le cloner dans un plasmide, qui est ensuite introduit dans une cellule hôte, en général une bactérie ou une levure. En se multipliant, cette cellule hôte produit un grand nombre de copies de chaque fragment d'ADN

d'origine. Après purification, cet ADN peut être séquencé en utilisant une polymérase qui synthétise un brin complémentaire à partir d'un brin matrice du fragment d'ADN d'intérêt ; quand la polymérase incorpore un des quatre didésoxynucléotides (ddATP, ddCTP, ddGTP, ou ddTTP présents séparément dans quatre réactions individuelles), la synthèse s'arrête. Cela génère un mélange de molécules qui se terminent à chaque position où se trouve un A, un C, un G, ou un T (selon le type de didésoxynucléotide présent) [11] .

Les fragments dans ce mélange sont séparés selon leur taille par électrophorèse sur gel.

La connaissance du didésoxynucléotide qui a été incorporé dans chaque réaction permet ainsi de déduire la séquence du fragment d'ADN d'intérêt [11] .

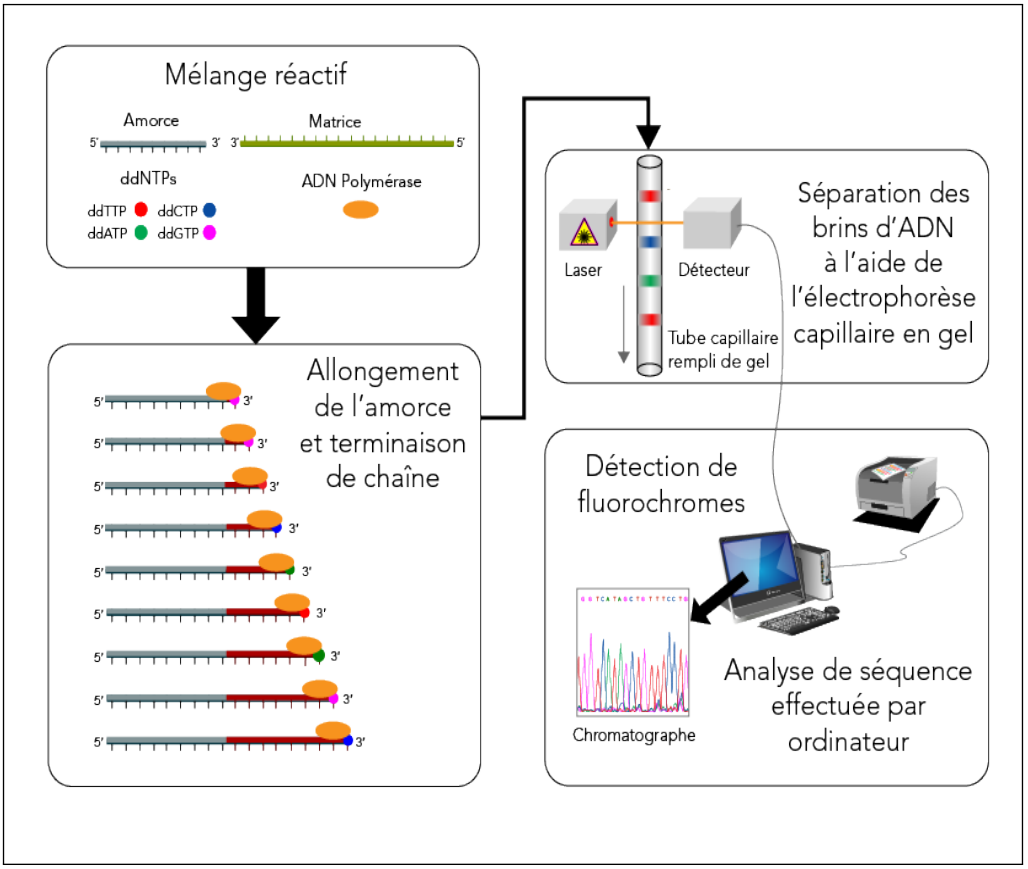

FIGURE 1.3: Technique de séquençage Sanger [12]

1.3.4 Séquençage à haut débit

Les méthodes de séquençage de nouvelle génération (next-generation sequencing, NGS) ont été développées afin de séquencer un grand nombre de génomes pour étudier la variation génétique.

Ces méthodes partagent trois améliorations majeures [11] par rapport au séquençage Sanger :

1. Au lieu d'un clonage moléculaire des fragments d'ADN suivi par l'introduction dans des cellules hôtes et l'isolement de chaque clone individuellement, une banque qui contient l'ensemble des fragments est faite directement dans un tube ; des fragments d'ADN, générés par coupures aléatoires (enzymatiques ou mécaniques) de l'ADN génomique sont reliés à des petites molécules d'ADN de séquences connues appelés « adaptateurs ».

   Les coupures aléatoires génèrent des fragments d'une grande diversité de tailles, Une sélection de taille est généralement effectuée dans le but d'éliminer les fragments plus

courts que la longueur de séquençage, et d'éliminer les fragments trop longs (d'une taille supérieure à environ 1000 nt) [13] .

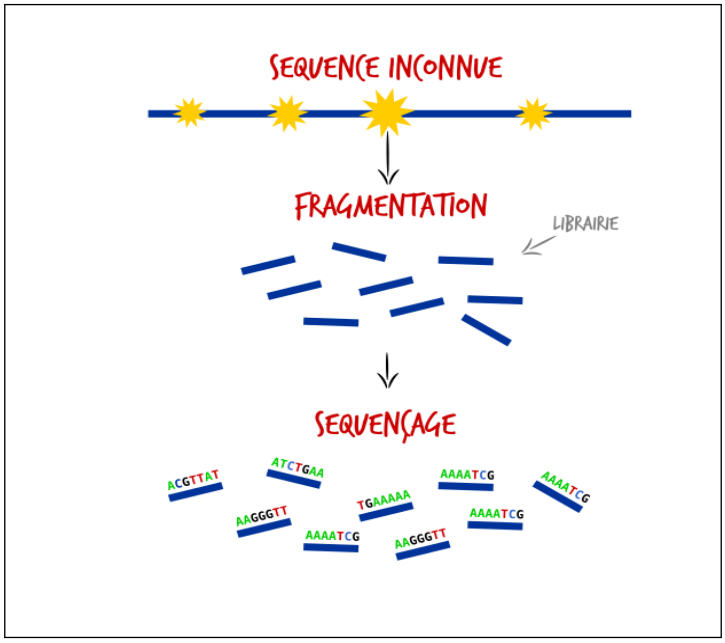

FIGURE 1.4: Processus de fragmentation de l'ADN [14]

Cette dernière étape est importante pour les techniques de NGS qui nécessitent ensuite une amplification par PCR, qui est moins efficace sur de longs fragments ;

2. Alors que les machines développées pour le projet de séquençage du génome humain n'étaient capables d'effectuer que quelques centaines de réactions de séquençage Sanger en parallèle, les séquenceurs NGS peuvent faire des millions voire des milliards de réactions de séquençage en parallèle ;
3. Les technologies NGS ne nécessitent pas de séparation des fragments par électrophorèse, la détection des nucléotides incorporés par la polymérase est faite directement après chaque cycle d'incorporation.

On distingue deux approches de séquençage à haut débit, suivant la stratégie de construction des librairies, l'approche ciblée et l'approche global [14] .

— **Séquençage ciblé** : Dans le but de séquencer une succession de quelques dizaines à quelques milliers de nucléotides d'ADN (la région d'intérêt), une étape d'amplification PCR (Polymerase Chain Reaction) précède le séquençage afin de répliquer l'ADN [14] . Cette méthode permet un séquençage même si la matière génétique est peu à cause de réplication, par contre une possibilité d'avoir des artefacts appelés biais d'amplification PCR.

— **Séquençage globale** : Dans le but de séquencer le génome entier, une fragmentation de manière aléatoire de ce génome en de très nombreux fragments de quelques dizaines à plusieurs centaines de nucléotides. Cette méthode nécessite l'existence d'une quantité suffisante de la matière génétique [14] .

1.3.4.1 Les plateformes de séquençage à haut débit (HTS)

La première technologie NGS était la méthode Roche 454 pyrosequencing by synthesis (SBS), qu'a été le premier système de séquençage de deuxième génération à succès commercial développé par 454 Life Sciences en 2005 et acquis par Roche en 2007, Les inconvénients majeurs de cette technologie sont le coût élevé des réactifs et le taux d'erreur élevé dans les répétitions d'homopolymères. Le coût estimé par million de base est de 10 $ par Roche 454 contre 0,07 $ par Illumina HiSeq 2000 [15] .

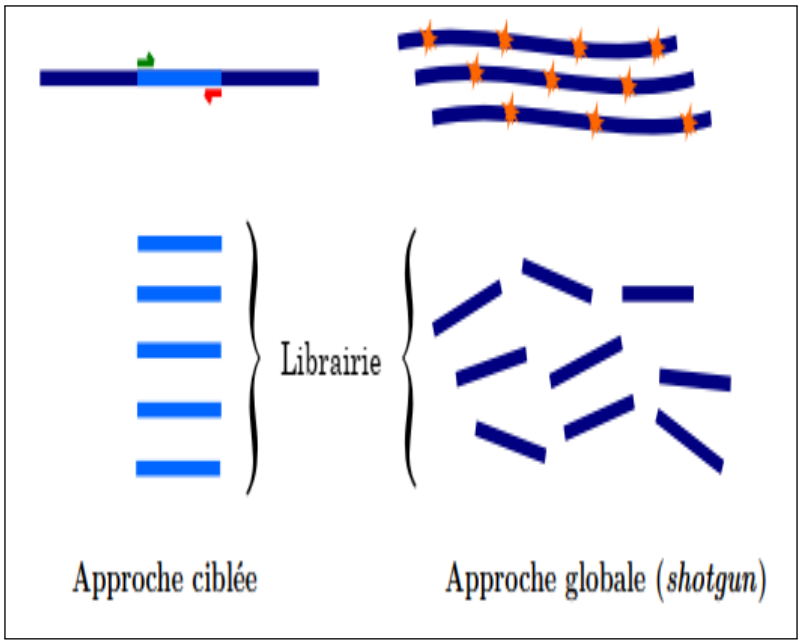

FIGURE 1.5: Principe du séquençage par approche ciblée et par approche globale (shotgun) [14]

Illumina a acheté le séquenceur Solexa en 2006 et l'a commercialisé en 2007. Aujourd'hui, c'est le système de séquençage le plus réussi avec un a revendiqué une domination de plus de 70% sur le marché, notamment avec les plateformes HiSeq et MiSeq [15] .

SOLiD est un séquenceur de nouvelle génération instrument commercialisé par Life Technologies et premier publié en 2008 par Applied Biosystems Instruments (ABI), L'avantage de cette méthode est précision avec chaque base interrogée deux fois. Les principaux inconvénients sont les courtes longueurs de lecture (50–75 pb) [15] .

plateforme	**Taille de read(bp)**	**No de reads**	**Temps d'exécution**	**Erreur(%)**	**Prix(USD)**
454 GS FLX+/Roche	700	1×10^{6}	24/48 h	1	500,000
HiSeq/Illumina	300	5×10^{9}	27/240 h	0.8	750,000
MiSeq/Illumina	600	3×10^{8}	27 h	0.8	125,000
SOLiD/Life Technologies	50	1×10^{9}	14 jours	0.01	350,000

TABLE 1.1: Les plateformes de séquençage à haut débit [15]

1.3.5 Séquençage de troisième génération (séquençage long-reads)

Même si les technologies NGS sont extrêmement puissantes, elles présentent aussi quelques faiblesses, par exemple la faible longueur des fragments séquencés. Des génomes contiennent souvent de nombreuses séquences répétées dont la longueur excède celle des fragments lus par NGS, ce qui mène à des difficultés dans l'assemblage du génome, afin d'éliminer ce problème les technologies dites de troisième génération (TGS, third generation sequencing) ou de lectures longues (long read) sont apparues. Elles se caractérisent par le séquençage, en temps réel de molécules uniques. Ces technologies permettent de générer des lectures d'une longueur de plusieurs kilo bases voire même des centaines de plusieurs kilos bases.

1.4 Conclusion

Dans ce chapitre,j'ai pu présenté les notions biologiques de base liées au problème de contamination dans un échantillon ADN, à savoir ADN, ARN, Gène et génome, dans l'objectif de comprendre le contexte biologique du problème.

Enfin, j'ai détaillé le processus de séquençage d'un ADN, son définition, l'évolution historique des méthodes de sequençage depuis son apparition, les différences entre eux, dans le but de mieux comprendre la source de mes séquences ADN.

Chapitre

2 METHODES BIO INFORMATIQUES

Sommaire

2.1 Introduction

je présente dans ce chapitre les approches et les méthodes bio-informatiques utilisées pour analyser les données présentées auparavant dans le chapitre précédent. Dans un premier lieu j'introduis la notion de comparaison entre deux séquence ADN en commençant par définir les formats des séquences ADN (FASTA et FASTQ) puis les algorithmes d'alignement existés. Dans la deuxième partie de ce chapitre, j'ai détaillé l'assemblage de l'ADN (technique d'assemblage De No et paradigmes utilisées).

2.2 Comparaison de séquences

Pour comparer les séquences ADN, il faut premièrement connaître la représentation et les formats de ces séquences, ensuit, l'alignement et les méthodes de base utilisées pour aligner deux séquences ADN.

2.2.1 Les formats de séquences (FASTA/FASTQ)

En Bio-Informatique, les données manipulées le long de mon projet (séquences ADN) sont stockées sous deux formats, FASTA et FASTQ.

2.2.1.1 Le format FASTA [16]

Le format FASTA est un format de fichier texte utilisé pour représenter et stocker les séquences nucléotidique/protéique.

Le fichier FASTA est composé au minimum de deux lignes. La première ligne commence obligatoirement par le chevron supérieur (>), suivi par un identifient de la séquence et un commentaire qui sont optionnel. La deuxième ligne et peut couvrir plusieurs lignes continue représente la succession de nucléotidiques/protéiques représentant la séquence.

> [identifiant][informations complémentaires]

GATCGTGANNNNTCATGCTTGGC

ACACGCTATGCTAGCTAGTACTA

2.2.1.2 Le format FASTQ [17]

Le format FASTQ est un format de fichier texte utilisé pour stocker à la fois les séquences nucléotidiques et les scores de qualité associés. La séquence et le score sont chacune codées avec un seul caractère ASCII.

Le fichier FASTQ utilise en principe 4 lignes par séquence. La première ligne commence par un caractère "@" (obligatoire) suivi de l'identifiant de la séquence et éventuellement d'une description (optionnels). La deuxième ligne représente la séquence nucléotidique. La troisième ligne commence par un caractère "+", parfois suivi par la répétition de l'identifiant de la séquence et de sa description.

La quatrième ligne est une succession de caractères (même nombre que le nombre des nucléotides représentant la séquence) en code ASCII, chaque caractère représente la probabilité d'erreur de séquençage associée à chaque nucléotide en Phred avec deux échelles (Phred+33 ou Phred+64).

@ [identifiant] [informations complémentaires]

AGTCGATCCGTACTAGCTA

+ [même identifiant, optionnel]

eed]‘]-Ba-MYBBB–[B

2.2.2 Problème d'alignement

L'alignement de séquences est une problématique importante de la bioinformatique, il s'agit de représenter le degré de similarité entre deux séquences. Dans un alignement, chaque position peut être un match (les nucléotides des deux séquences sont similaires pour une position donnée), une édition (les nucléotides des deux séquences sont différents pour une position donnée).

Une édition peut être un mismatch, (existence d'une substitution d'un nucléotide par un autre dans une position donnée), un indel (l'insertion ou la délétion d'un nucléotide dans une position donnée) [18] .

Dans un problème d'alignement, le pourcentage d'identité entre deux séquences est quotient du nombre de matchs et la longueur de l'alignement. On dit que deux séquences sont identiques si ce rapport est à 100%.

Dans un problème d'alignement, on cherche à maximiser son score d'homologie ou de minimiser le score de distance entre deux séquences.

2.2.2.1 Types d'alignements

Le problème d'alignement en bio-informatique peut être classé en trois types [18] :

— **Alignement global** : consiste à chercher un alignement sur toute leur longueur deux séquences de tailles comparables, toute en pénalisant les éditions en amont ou en aval des séquences.

— **Alignement local** : il s'agit de trouver le meilleur alignement en manière de score entre deux sous séquences des séquences originales. Généralement utilisé pour aligner une petite séquence sur une séquence bien plus grande.

— **Alignement semi-global (glocal)** : consiste à trouver l'alignement qui couvre le début et la fin de l'une ou l'autre des séquences. Utilisé dans la comparaison entre le préfixe de la première séquence avec le suffixe de la deuxième, et inversement, il est utilisé aussi dans le cadre de l'assemblage avec graphe de chevauchement.

2.2.3 Algorithmes d'alignement

Pour résoudre le problème d'alignements, plusieurs algorithmes ont été proposés, on peut les classer en trois types en se basant sur la taille des données à aligner [18] :

2.2.3.1 Alignement Gène contre gène

Utilisé dans des applications simples, ont pour objectif aligner deux séquences courtes (quelques milliers de nucléotides), il s'appuie sur des méthodes exactes par programmation dynamique.

— algorithme de Needleman-Wunsch [19] pour le problème d'alignement global, avec une complexité quadratique ;

— algorithme de Smith-Waterman [20] pour le problème d'alignement local, avec une complexité quadratique.

2.2.3.2 Alignement Gène contre génome

Utilisé pour résoudre le problème d'alignement ensemble réduit (quelques milliers) de courtes séquences contre un génome complet, dans ce cas les méthodes exactes sont déconseillées par rapport à leurs lenteurs, les méthodes heuristiques sont nécessaires pour ce type de problème.

— Les techniques à base de graines, d'abord une recherche de courtes sous séquences (k-mers) similaires dans les deux séquences à aligner, en suite, l'alignement sera étendu par programmation dynamique ;

— La technique d'indexation des génomes, m'a permit d'accélérer la recherche de graines dans les deux séquences.

Basic Local Alignment Search Tool BLAST [21] exploite une hybridation de deux approches, index et grain, il est considéré comme étant le logiciel d'alignement le plus populaire, il aligne un nombre raisonnable de courtes séquences sur un génome complet avec une vitesse important.

2.2.3.3 Alignement de lectures de séquençage contre génome

Apparue avec la révolution de séquençage à haut débit, il s'agit d'aligner d'importante quantité de lectures (plusieurs milliards), de petite taille (quelques centaines de nucléotides)

sur un génome complet de référence, plusieurs méthodes existantes pour résoudre ce type de problème telles que Bowtie [22] et BWA [23] , récemment, des approches telles que SortMeRNA [24] sont apparues.

2.3 Assemblage de l'ADN

L'opération de séquençage à haut débit, notion évoquée dans le chapitre précèdent, consiste à passer d'une séquence initiale d'ADN vers un ensemble de lectures (reads) de séquençage. L'assemblage est l'opération inverse, c'est-à-dire ; reconstruire l'intégralité de la séquence initiale à partir de ces courtes lectures. Le problème d'assemblage est un problème central en bio-informatique, au même poids que l'alignement de séquences, On distingue deux types d'assemblage :

— Assemblage en s'appuyant sur des génomes de référence, c'est une technique Le mappage, qui aligne les fragments courts sur une séquence de référence au format FASTA ;

— L'assemblage de novo, c'est la technique qui m'intéresse dans mon projet, elle consiste à une reconstruction de zéro (de novo) sans avoir l'information sur les génomes de référence par plusieurs méthodes d'assemblage.

2.3.1 Définitions

Le problème de l'assemblage peut être comparé à celui de la reconstruction du texte d'un livre à partir de plusieurs copies de celui-ci, préalablement déchiquetées en petits morceaux.

Un assemblage de novo consiste à chevaucher des petites lectures issues d'une séquence originale suite à une opération de séquençage à haut débit, dans le but de reconstruire cette séquence de départ [35] .

En bio-informatique l'assemble est utilisé pour reconstruire les génomes, les gènes ou l'ADN complet d'un organisme vivant.

L'assemblage de novo confronte plusieurs difficultés qui le rendent un problème très complexe, parmi ces difficultés [18] , on cite :

— La présence d'erreurs de séquençage, selon la technologie utilisé, l'ensemble de lectures présente les types d'erreurs cités auparavant (substitution et InDel) avec un pourcentage d'erreurs qu'il faut prendre en considération ;

— la présence de répétitions dans les génomes séquencés, il faut utiliser une technique de séquençage produisant les lectures les plus longs possibles ;

— La grande quantité de données générées par l'opération de séquençage ;

— Le problème de contamination illustré par la présence de différentes organismes vivants dans le même échantillon, donc il faut développer des méthodes informatiques pour séparer entres ces organismes.

On peut classer les assembleurs en trois classes selon le paradigme d'assemblage utilisé [26] :

— Les assembleurs gloutons ;

— Les assembleurs OLC (Overlap-Layout-Consensus) ;

— Les assembleurs basés sur un graphe de De Bruijn.

2.3.2 Le paradigme glouton

Historiquement la première stratégie d'assemblage, elle consiste à chercher les chevauchements entre les lectures deux à deux, par la suite construire une super séquence en regroupant les lectures chevauchées dans l'ordre décroissant de leur score de chevauchement. Ce paradigme mène à des optimums locaux sans prendre en considération les relations globales entre les

lectures. A cause de régions répétées existantes dans un génome, le paradigme glouton n'est pas adapté pour le problème d'assemblage de génome.

Les assembleurs phrap, CAP3 [27] , ainsi que les assembleurs les plus récents tels que VCAKE [28] reposent sur ce paradigme d'assemblage.

Dans la pratique, les assembleurs modernes sont des assembleurs à base du paradigme OLC ou De Bruijn.

2.3.3 Le paradigme OLC

Le paradigme OLC (Overlap-Layout-Consensus) consiste à chercher les chevauchements entre les lectures (overlap) deux à deux pour construire une super séquence mais les chevauchements sont traités de manière globale contrairement au paradigme glouton.

Dans ce paradigme un graphe de chevauchements est construit, à partir de ce dernier, on identifie les contigs (sous-graphes), et la séquence la plus envisageable pour chacun des contigs est retenue (Consensus).

2.3.3.1 Graphe de chevauchement

Est un graphe orienté, les nœuds repensent les lectures du séquençage, tandis que l'arc représente un chevauchement entre le suffixe du son nœud de départ et le préfixe du son nœud d'arrivée.

L'approche naïve pour établir un graphe de chevauchement est de calculer l'alignement semi-global entre chaque paire de lectures de l'ensemble global des lectures. Une arête est ajoutée entre les deux lectures si le chevauchement est suffisamment long et similaire sinon pas d'arête. Cette approche a une complexité quadratique par rapport au nombre de lectures, dans le cas d'une grande quantité de lectures (technologies de séquençage à haut débit, séquençage d'échantillons méta génomiques) l'approche devient peu pratique.

Une autre approche plus intelligente est utilisée, d'abord pour accélérer la comparaison des paires de lectures on appuie sur les méthodes heuristiques. Ainsi pour chercher les chevauchements entre paires de lecture sans erreur, on utilise les structures d'index qui permet le calcule rapide d'ensemble des chevauchements entre les lectures. Plusieurs structures sont utilisées, telles que la table des suffixes/préfixes [29] , la table des suffixes compressés [30] ou le FM-index [31] .

Exploiter le graphe de chevauchement dans un problème d'assemblage consiste à identifier la plus courte super-séquence, il s'agit de manière schématique, à trouver le chemin minimal qui visite tous les nœuds du graphe une et une seule fois. C'est le problème connu de chemin hamiltonien, qui est un problème NP-complet, sa résolution est NP-difficile, donc on fait appel aux heuristiques pour exploiter le graphe de chevauchement dans des temps raisonnables.

2.3.3.2 Les méthodes OLC

Le paradigme OLC est populaire grâce aux travaux de Gene Myers, plus précisément l'assembleur Celera [32] , le plus utilisé jusqu'à l'émergence des technologies de séquençage haut débit. Ces méthodes ont montré leurs incapacités d'assembler les grandes quantités de données issues des technologies de séquençage haut débit, dans ce cas le stockage de graphes de chevauchement nécessite un espace mémoire important.

Pour remédier ce problème, un nouveau paradigme d'assemblage est apparu et il fait appel à des graphes de De Bruijn.

Plus récemment, on remarque la réapparition des méthodes basées sur le paradigme OLC, notamment grâce à la naissance de l'assembleur SGA [33] qui exploite les performances de la structure FM-index, cet assembleur est capable d'assembler de grands génomes eucaryotes toute en utilisant une mémoire raisonnable (50 GB RAM).

2.3.4 Le paradigme basé graphe de De Bruijn

Un paradigme d'assemblage proposé par Pevzner et al. en 2001 [34] , il consiste à construire un graphe de De Bruijn, il a pour but de réduire les ressources nécessaires à l'assemblage pour les grandes quantités de données générées par les méthodes de séquençage à haut débit.

2.3.4.1 Graphe de Brujin

Est le graphe particulier de chevauchement de tous les k-mers de taille fixée de l'ensemble de toutes les lectures. Est un graphe orienté composé de nœuds sous formes des mots de taille k, reliées par des arrêtes qui représentent les chevauchements entre les mots (k-mers) de taille k - 1 entre les k-mers [34] .

Un graphe de De Bruijn représente l'information compressée de l'ensemble de lectures. En pratique, On peut stocker un graphe de Brujin dans un espace réduit comparant aux graphes de chevauchement, à titre d'exemple, on stocke seulement les k-mers dans une table de hachage, tandis que les chevauchements seront déduits sans les stocker. Ainsi qu'on puisse aussi stocker le nombre d'apparitions de chaque mot de taille k, ce qui assigne un poids pour chaque arête du graphe de De Bruijn. La construction du graphe de De Bruijn peut être réalisée en temps linéaire, contrairement à la construction d'un graphe de chevauchement de lectures [34] .

Pour exploiter un graphe de Brujin, théoriquement, dans le cas couverture de séquençage homogène et de lectures sans erreurs de séquençage, l'exploitation du graphe est facile, Chercher la super séquence commune la plus courte revient à résoudre le problème de chemin Eulérien, soit le chemin qui visite toutes les arêtes du graphe une seule fois [34] .

Pour ce type de problèmes, des algorithmes pour trouver un chemin eulérien dans un graphe existes et efficaces et déterministes, contrairement aux problèmes liés aux graphes de chevauchement. Toutefois, En pratique, cela veut dire une présence d'erreurs de séquençage, de répétitions complexifie, l'exploitation de ce graphe devient non triviale. Des heuristiques standards sont utilisées en post assemblage pour simplifier et nettoyer et un graphe de De Bruijn, on cite :

— L'éclatement des bulles, en éliminant les arêtes de faible poids ;
— L'élimination des tips ;
— la suppression de chemins chimériques.

2.3.4.2 Le scaffolding (échafaudage)

Est une étape post assemblage supplémentaire, utilisée par la plupart des assembleurs modernes, elle consiste à exploiter les informations supplémentaires autres que celles portées par les lectures générées par un séquenceur dans le but d'ordonner les contigs entre eux.

Le scaffolding repose principalement sur l'information d'appariement des lectures de séquençage pour les librairies appariées ou mate-pairs pour réaligner les lectures de séquençage sur les contigs, en premier lieu une organisation et une orientation des contigs entre eux en se basant sur les lectures appariées et mate-pairs, ensuite une estimation de la distance entre ces diffents contigs en exploitant l'infirmation de la taille des fragments sur lesquels sont séquencées. Finalement, une génération d'un scaffold en concaténant les contigs ordonnés, et en rajoutant des nucléotides inconnus (notés « N ») entre les contigs [18] .

2.4 Conclusion

Dans ce chapitre, d'abord, j'ai présenté les notions bio-informatiques de base utilisées dans la décontamination des séquences d'ADN, à savoir la notion de la comparaison entre deux séquences d'ADN, toute en détaillant les formats de séquences, le problème et les algorithmes d'alignement.

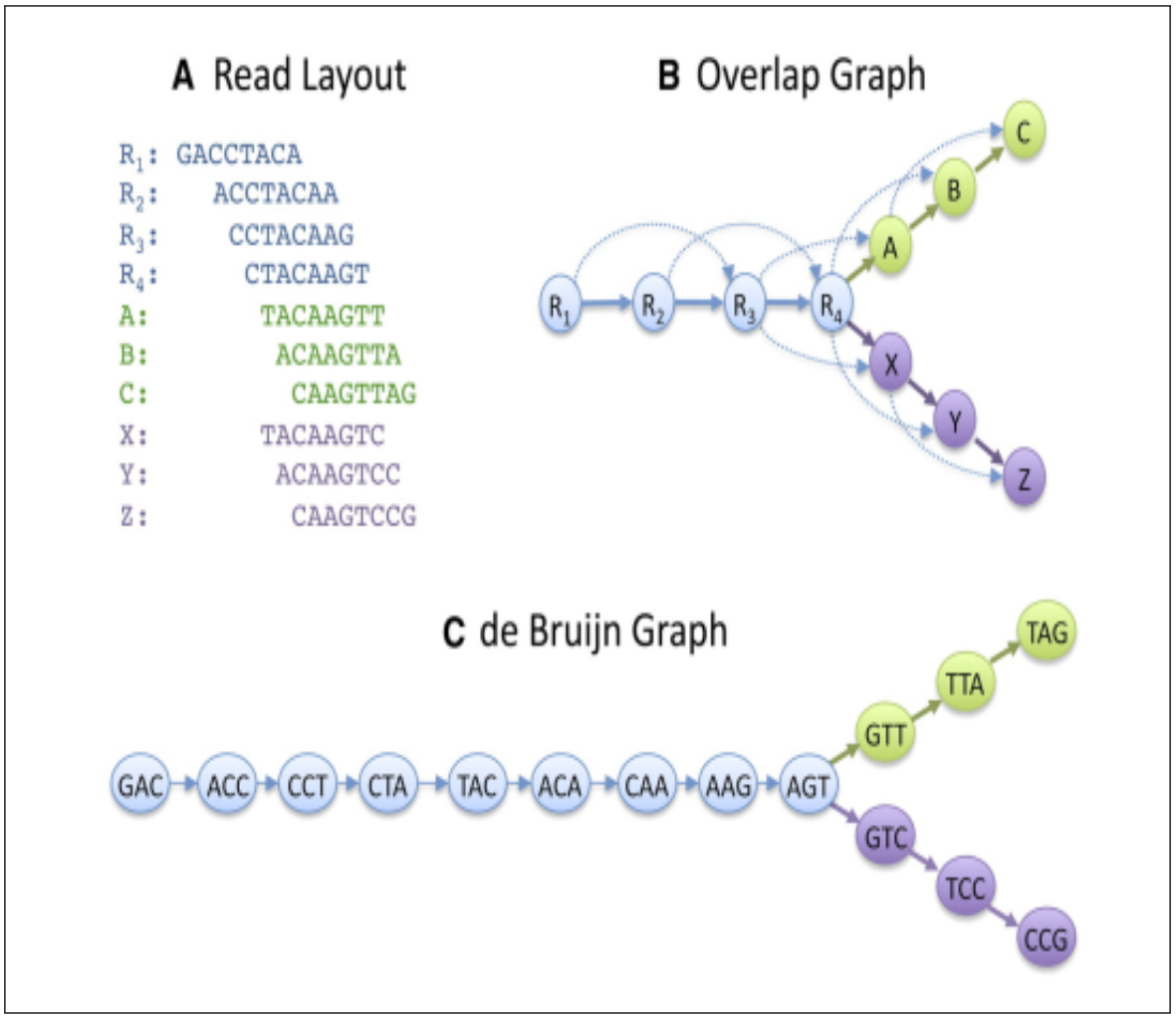

FIGURE 2.1: Différences entre un graphe de chevauchement et un graphe de De Bruijn [35]

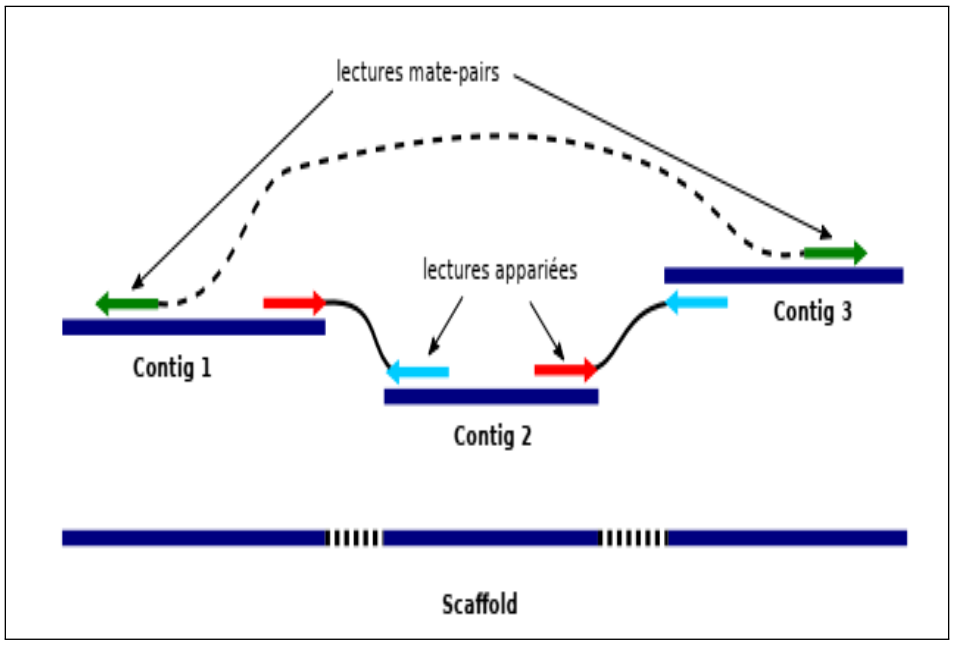

FIGURE 2.2: Principe du scaffolding [18]

Ensuite, j'ai présenté la notion d'assemblage des séquences d'ADN, toute en détaillent les paradigmes d'assemblage .

Chapitre

3 ÉTAT DE L'ART

Sommaire

3.1 Introduction

Ce chapitre est composé de deux sections, premièrement, une introduction aux concepts informatiques utilisés dans la partie pratique (le chapitre suivant) à savoir les techniques informatiques telles que l'intelligence artificielle, machine learning et apprentissage supervisés.

Deuxièmement, une présentation des méthodes existantes dans la littérature et les travaux effectués pour résoudre le problème de contamination des séquences d'ADN. Dans ce chapitre, les avantages et les incontinents de chaque méthode, ainsi qu'une conception d'une taxonomie de ces méthodes en s'appuyant sur leurs concepts de base pour décontaminer un échantillon donné.

3.2 Notions Informatiques

3.2.1 Intelligence artificielle

L'intelligence artificielle (IA, ou AI en anglais pour Artificial Intelligence) consiste à mettre en œuvre un certain nombre de techniques visant à permettre aux machines d'imiter une forme

d'intelligence réelle. L'IA se retrouve implémentée dans un nombre grandissant de domaines d'application [36] .

La notion a vu le jour dans les années 1950 grâce au mathématicien Alan Turing. Dans son livre Computing Machinery and Intelligence, ce dernier soulève la question d'apporter aux machines une forme d'intelligence. Il décrit alors un test aujourd'hui connu sous le nom « Test de Turing » dans lequel un sujet interagit à l'aveugle avec un autre humain, puis avec une machine programmée pour formuler des réponses sensées. Si le sujet n'est pas capable de faire la différence, alors la machine a réussi le test et, selon l'auteur, peut véritablement être considérée comme « intelligente » [36] .

De Google à Microsoft en passant par Apple, IBM ou Facebook, toutes les grandes entreprises dans le monde de l'informatique planchent aujourd'hui sur les problématiques de l'intelligence artificielle en tentant de l'appliquer à quelques domaines précis. Chacun a ainsi mis en place des réseaux de neurones artificiels constitués de serveurs et permettant de traiter de lourds calculs au sein de gigantesques bases de données [36] .

Parmi les technologies d'intelligence artificielle on cite le Deep learning et le Machine learning..

3.2.2 Machine Learning

Dans de nombreuses disciplines scientifiques, l'objectif premier est de modéliser la relation entre un ensemble de grandeurs observables (entrées) et un autre ensemble de variables qui leurs sont liées (sorties). Une fois qu'un tel modèle mathématique est déterminé, il est possible de prédire la valeur des variables souhaitées en mesurant les entrées [37] .

Malheureusement, de nombreux phénomènes du monde réel sont trop complexes à modéliser directement sous la forme d'une relation entrée-sortie [37] .

L'apprentissage automatique (Machine learning) fournit des techniques qui peuvent automatiquement construire un modèle informatique de ces relations complexes en traitant les données disponibles et en maximisant les critères de performance [37] .

Le processus automatique de modélisation est appelé l'entraînement et les données utilisées sont des données d'apprentissage.

Les techniques d'apprentissage automatique peuvent être classées en deux grandes catégories, apprentissage supervisé et apprentissage non supervisé [37] .

3.2.3 Classification supervisée

Dans le contexte supervisé on dispose déjà d'exemples dont la classe est connue et étiquetée. Les données sont donc associées à des labels des classes notés = q1, q2, ..., qn [38] .

L'objectif est alors d'apprendre à l'aide d'un modèle d'apprentissage des règles qui permettent de prédire la classe des nouvelles observations ce qui revient à déterminer une fonction Cl qui à partir des descripteurs (D) de l'objet associe une classe qi et de pouvoir aussi affecter toute nouvelle observation à une classe parmi les classes disponibles. Ceci revient à la fin à trouver une fonction qu'on note Ys qui associe à chaque élément de X un élément de Q. On construit alors un modèle en vue de classer les nouvelles données. Parmi les méthodes supervisées on cite : les k-plus proches voisins, les arbres de décision, les réseaux de neurones, les machines à support de vecteurs (SVM) et les classificateurs de Bayes [38] .

Quel que soit le type de la classification, on est confronté à différents problèmes. Dans le cas supervisé, un problème important peut être le manque de données pour réaliser l'apprentissage ou la disponibilité de données inadéquates par exemple incertaines et imprécises ce qui empêche la construction d'un modèle correct. Pour la classification non-supervisée, la délimitation des frontières entre les classes n'est pas toujours franche et reconnaissable. Indépendamment du type de classification, les données multi dimensionnelles, ou encore la dépendance des méthodes

de classification aux paramètres initiaux comme le nombre de classes peuvent poser problèmes [38] .

3.2.3.1 Le voisin le plus proche (K-NN)

Un algorithme k-plus proche voisin, souvent abrégé k-NN, est une méthode standard d'apprentissage supervisé qui consiste à classer les observations non étiquetées par en les affectant à la classe des étiquetés les plus similaires [39] .

La methode KNN repose sur deux principes de base [39] .

L'une est la méthode pour calculer la distance entre deux observations. Par défaut, le La fonction KNN utilise la distance euclidienne.

Un autre concept est le paramètre k qui décide combien de voisins seront choisis pour l'algorithme kNN.

La clé pour choisir une valeur k appropriée est de trouver un équilibre entre le sur-apprentissage et le sous-apprentissage [39] .

3.2.3.2 Arbres de décision

Un arbre de décision (Quilan, 1986), (Quinlan, 1983) est la représentation graphique d'une procédure de classification. Il permet de modéliser simplement, graphiquement et rapidement un phénomène mesuré plus ou moins complexe. Pour certains domaines d'application, il est essentiel de produire des procédures de classification compréhensibles par l'utilisateur [40] .

Un arbre de décision est constitué d'un ensemble de règles permettant de segmenter (diviser, partitionner) un ensemble de données en groupes homogènes. Chaque règle associe une conjonction de tests sur les variables descriptives. Le premier sommet est appelé la « racine » de l'arbre, les variables suivantes qui correspondent aux nœuds non terminaux (tests sur les attributs) sont des variables de segmentation, chaque branche (arête, arc) correspond à une modalité de la variable (réponse à un test) considérée à ce niveau de l'arbre [40] .

Lorsque les tests sont binaires, l'une des branches correspond à une réponse positive au test et l'autre branche à une réponse négative, les feuilles représentent les classes [40] .

3.2.3.3 Machine à vecteurs de support (SVM)

SVM (Support Vector Machines) ou machines à vecteurs supports est un classificateur discriminatif formellement défini par un hyperplan séparateur. Cette méthode découle directement des travaux de Vapnik [41] sur la théorie de l'apprentissage statistique.

Principe de fonctionnement du SVM :

Cette technique fait appel à un jeu de données dit « d'apprentissage » dont la variable classe (étiquette, label) des instances est donné par un expert, et ce pour générer un modèle permettant de prédire l'étiquette de classe d'un autre ensemble dit « de test » en fonction des attributs (variables observées).

Le principe de base des SVMs consiste de trouver un hyperplan séparateur, qui peut être utilisé pour la classification, la régression ou d'autres tâches. Intuitivement, une bonne séparation est obtenue par l'hyperplan qui maximise la distance entre les classes (la marge), dont l'appellation de classificateur à vaste marge. La marge est la distance entre la frontière de séparation et les échantillons les plus proches, ces derniers constituent un vecteur appelé le vecteur de support [42] .

Utilisation des noyaux : Le fait d'admettre la mal-classification de certains exemples, ne peut pas toujours donner une bonne généralisation pour un hyperplan même si ce dernier est optimisé.

Plutôt qu'une droite, la représentation idéale de la fonction de décision serait une représentation qui colle le mieux aux données d'entraînement.

La détermination d'une telle fonction non linéaire est très difficile voire impossible. Pour cela les données sont amenées dans un espace où cette fonction devient linéaire (figure 2.5), cette astuce permet de garder les mêmes modèles de problèmes d'optimisation vus dans les sections précédentes, utilisant les SVMs basées essentiellement sur le principe de séparation linéaire. Cette transformation d'espace est réalisée souvent à l'aide d'une fonction appelé "Mapping function" [42] .

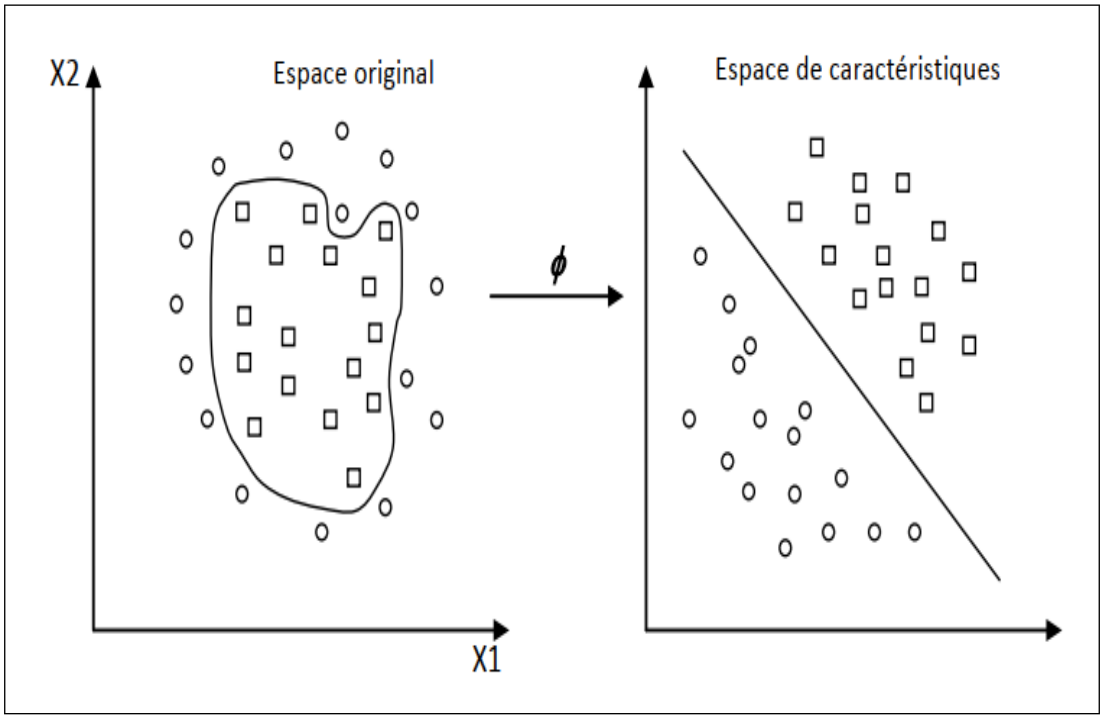

FIGURE 3.1: Transformation d'espace [42]

Dans le calcul de l'optimum dans l'espace de caractéristiques, on introduit le calcul d'une fonction appelée « noyau » (kernel). Cette fonction permet de surmonter le problème de détermination de la transformation [42] .

l existe des noyaux qui sont largement utilisés, dont l'exemple des :

— **Noyau linéaire** [42] : Si les données sont linéairement séparables, on n'a pas besoin de changer.

— **Noyaux RBF (Radial Basis Functions)** [42] : à titre d'exemple le noyau Gaussien. d'espace

3.3 Travaux Connexes

Dans cette partie, je vais présenter les méthodes pour la recherche et la décontamination de séquences ADN contaminées dans un assemblage De Novo selon la philosophie de résolution et les outils utilisés dans chaque méthode. Deux catégories, les méthodes de décontamination **dépendantes** des bases de données et les méthodes **indépendantes** des bases de données.

En premier lieu et pour résoudre le problème de décontamination de séquences d'ADN, les méthodes dépendantes des bases de données ont été largement utilisées, ces dernières reposent sur le principe d'alignement sur des séquences d'organismes connus pour prédire si un transcrit provient d'un contaminant ou non. Ces méthodes ont montré leurs efficacités lorsqu'il s'agit

d'un organisme cible connu et des organismes contaminants réduits et connus en littérature, ainsi que ces méthodes dépendent de la qualité de la base de données, une base de données qui couvre un maximum d'organismes favorise et améliore le résultat de décontamination, contrairement à une base de données réduite. D'autre part ces méthodes sont inappropriées dans le cas d'un nouvel organisme inexistant dans la base de données de références, ainsi dans le cas d'un organisme qui ne possède pas un génome de référence. Dans ces situations certains transcrits n'auront aucun alignement et d'autres auront plusieurs alignements, ces transcrits ambiguës sont difficiles à classer.

Une autre famille des méthodes de décontamination est apparue, elle s'agit des méthodes indépendantes des bases de données, elles reposent essentiellement sur le principe d'apprentissage supervisé ou non supervisé, l'introduction de ce principe a marqué une énorme évolution de la décontamination de séquences d'ADN et m'a permit l'élimination des problèmes liés à la première catégorie des méthodes à savoir les organismes vivants inconnus. Par contre Peu de recherches ont été faites pour développer les méthodes de cette famille.

On a réalisé une taxonomie (voir la figure 3.2)pour classer les différentes méthodes et techniques qui ont pour but la décontamination de séquences ADN en démontrant le principe de chaque méthode, les points de similarité entre eux, ainsi que les différences majeures entre les méthodes existantes, la figure suivante illustre d'une manière globale la taxonomie liée à ce problème.

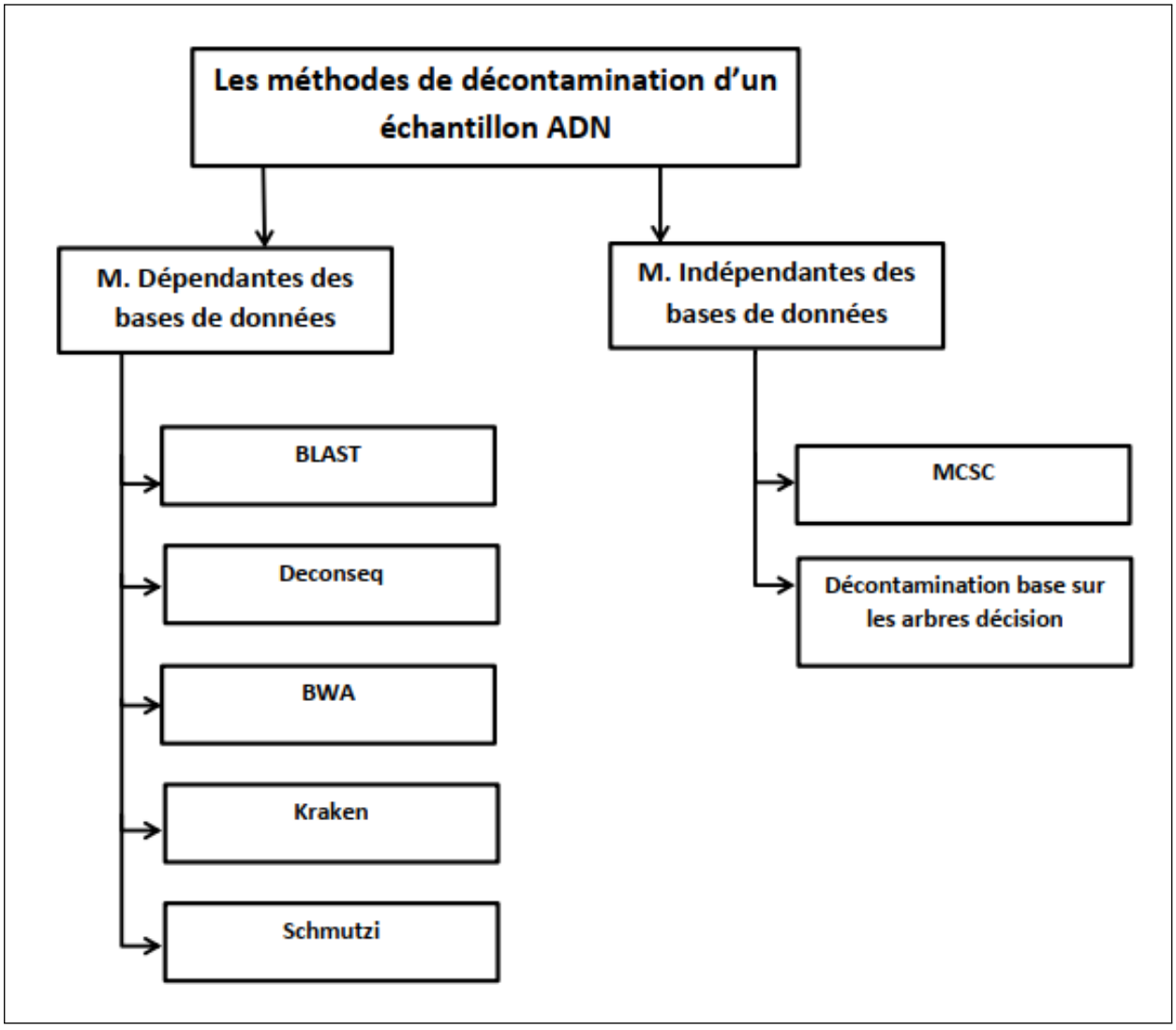

FIGURE 3.2: Taxonomie des méthodes de la décontamination de séquences adn contaminantes

3.3.1 Les méthodes de décontamination dépendantes des bases de données

Il existe plusieurs outils qui se sont développés pour résoudre le problème de contamination des séquences d'ADN en se basant sur des bases de données.

Dans cette section on va présenter le principe quelques méthodes existantes telles que BLAST, BWA, DeconSeq, Kraken et Schmutzi . Ainsi le principe de chacune de ces méthodes.

— **BLAST** [43] (basic Local Alignment Search Tool), est une méthode qui recherche dans une base de données génétique, des segments qui sont localement homologues à une séquence d'étude.

BLAST utilise une matrice de similarité pour calculer des scores d'alignement. Il fournit un score pour chaque alignement trouvé et utilise ce score pour donner une évaluation statistique de la pertinence de cet alignement.

— **Deconseq** [44] est méthode caractérisée par l'utilisation de deux bases de données, une base de données white, qui regroupe les séquences génétique proches à l'organisme d'étude (organisme cible), et une deuxième base de données dite black, qui comporte les séquences génétiques potentiellement proches aux organismes contaminants.

Le principe de cette méthode est d'aligner la séquence d'ADN qu'on veut classer, sur ces deux bases de données en se basant sur BWA (Li and Durbin 2009, Li and Durbin 2010), par la suite et en fonction du résultat d'alignement obtenu, on assigne chaque séquence à sa base de données correspondante.

— **BWA** [45] est une méthode rapide qui aligne des lectures relativement courtes, de l'ordre de 50, 100, ou 1,000 bp sur une séquence de référence (ex :génome complet)

— **Kraken** [46] (Kraken : ultrafast metagenomic sequence classification using exact alignments) est une méthode ultrarapide et très précise pour attribuer des étiquettes taxonomiques à des séquences d'ADN méta génomiques, Kraken classe 100 lectures de paires de bases à un taux de plus de 4,1 millions de lectures par minute.

— **Schmutzi** [47] est un Outil bioinformatique développé pour estimer les contaminations dans les anciens ensembles de données d'ADN humain et reconstruire le génome mitochondrial endogène (même avec une contamination >50 %).

3.3.2 Les méthodes de décontamination indépendantes des bases de données

— **MCSC decontamination method** [48] : est une méthode de décontamination en utilise l'apprentissage non supervisé plus précisément le clustering. Ce modèle statistique permet à l'algorithme de construire une représentation efficace de chaque cluster en utilisant un arbre de suffixes probabiliste et calculer les similitudes entre les clusters de séquences au lieu des séquences- séquences, l'algorithme MCSC fonctionne comme suit :

1. Construire un seul cluster qui contient tous les séquences, ce cluster est divisé en deux clusters préliminaires par une analyse floue des correspondances multiples (F-MCA) sur le vecteur représentation des séquences ;
2. Un centre statistique est calculé pour chaque cluster et la similarité du Chi-square de chaque séquence est calculée par rapport à chaque cluster ;
3. Réaffectation de chaque séquence à l'autre cluster si elle est plus similaire à ce cluster ;
4. Le modèle est construit pour chaque cluster et certaines réaffectations sont effectuées pour l'amélioration finale ;
5. Après le regroupement, pour chaque cluster, un alignement BLAST de toutes ses séquences contre les listes blanche et noire, utilisé pour calculer un ratio de liste blanche (WR) pour chaque cluster ;
6. Le Clusters avec un rapport WR inférieur à 0,8 est étiqueté comme contaminant, donc il contient des séquences contaminants.

— **Méthode de décontamination basée sur les arbres de décision** [49] : est une méthode d'apprentissage supervisée a pour objectif de séparer les séquences d'un échantillon en deux classes, classe cible te classe contaminant. L'algorithme fonctionne comme suit :

1. Construire le data set d'apprentissage c à d, calculer pour chaque séquence 8 attributs pour construire un vecteur représentant la séquence ((1) longueur, (2) Contenu du GC, (3) couverture moyenne du séquençage de l'ADN, (4) couverture moyenne du séquençage de l'ARN, (5) pourcentage de l'échafaudage couvert par l'alignement de l'ADN, (6) pourcentage de l'échafaudage couvert par l'alignement de l'ARN, (7) Contenu GC des lectures d'ADN alignées, and (8) Contenu GC des lectures d'ARN alignées) ;
2. Construire l'arbre de décision en se basant sur le data set ;
3. Prédire les nouvelles séquences.

3.4 Conclusion

Dans ce chapitre, je vais présenté les notions informatiques liées au probléme, à savoir, l'intelligence artificielle, machine learning et la classification supervisée.

En deuxième lieu, je vais détaillé les travaux et les méthodes connexes au problème, en exposant ma taxonomie de ces travaux, qui classe ces méthodes, en méthodes indépendantes des bases de données et une autre catégorie dépendante des bases de données.

Chapitre

4 CONTRIBUTION

Sommaire

4.1 Introduction

Le séquençage à haut débit a permis théoriquement d'obtenir des séquences génomiques à hautes qualités, mais en pratique, les extraits d'ADN sont souvent contaminés par des séquences d'autres organismes vivants. Actuellement, il existe peu de méthodes existantes pour décontaminer rigoureusement ces séquences. La plupart sont des méthodes de filtrage basées sur la similarité des nucléotides avec des génomes de référence, par ailleurs elles risquent d'éliminer les séquences d'organisme cible.

Dans ce chapitre, on va présenter ma contribution pour résoudre le problème de contamination qui s'appuie sur l'apprentissage supervisé, toute en détaillant les étapes de la conception et de la réalisation de cette méthode, les outils informatiques utilisées pour la réalisation de ma méthode.

4.2 Conception du modèle de décontamination de séquence d'ADN

Pour résoudre le problème de contamination au sein d'un échantillon séquences d'ADN contaminé, en exploitant les avantages des méthodes indépendantes de base de données évoquées dans le chapitre précédent, on a opté pour les techniques d'apprentissage supervisé.

On a pu réaliser trois techniques d'apprentissage supervisé à savoir l'arbre de décision, les KNN et les SVM.

Pour cela, le processus de conception est devisé en étapes suivantes :

— Collection et préparation des données ;

— Détermination d'attributs représentants mes données ;

— Choix et entraînement du modèle d'apprentissage ;

— Évaluation des performances du modèle.

4.2.1 Collection et préparation des données

La tâche de préparation des données comprend la collecte, leurs nettoyages et étiquetages, on s'intéresse aux séquences d'ADN issues de plusieurs organismes vivants (cibles et contaminants).

La classe cible contient les séquences de génome vibrio cholerae (organisme tareget), tandis que les contaminants comportent les génomes suivant ; Vibrio alginolyticus, Vibrio cincinnatiensis, Vibrio fluvialis, Vibrio furnissii, Vibrio harveyi, Vibrio metoecus, Vibrio mimicus, Vibrio parahaemolyticus et autres.

À la fin de cette étape, on aura un ensemble de séquences d'ADN étiquetées.

4.2.2 Détermination d'attributs représentants mes données

Cette étape consiste à transformer les données brutes (séquences d'ADN) en caractéristiques (valeurs numériques) représentants plus précisément le problème sous-jacent au modèle prédictif, en appliquant une connaissance du domaine pour extraire des représentations analytiques à partir des données brutes et de les préparer pour le Machine Learning. Pour cela on a utilisé les attributs suivants :

1. **k-Gram** : l'attribut est représenté par une paire de valeurs (v,c).
   — **v** : est décrit comme un mélange de plusieurs unités nucléotidiques (A, G, C, T), avec k=1, et k=2 ; la taille du mélange.
   — **c** : représente le nombre d'occurrences de chaque v dans une séquence ADN donnée divisé par la taille de la séquence en question.
   — k-gram est utilisé pour récupérer les caractéristiques des séquences et la liste G d'une combinaison de nucléotides peut être représentée comme suit :

$$\begin{aligned} G &= G_1 \cup G_2 \\ &= \{N_i\} \cup \{N_i N_j\} \\ &= \{A, C, G, T, AA, AC....., TT\} \end{aligned} \tag{4.1}$$

   L'ensemble G1 comprend 4 attributs 1-Gram et l'ensemble G2 comprend 16 attributs 2-Gram.

2. **MMI (Multivariate Mutual Information)** :est une procédure avancée pour collecter les caractéristiques des séquences nucléotidiques.
   — j'ai décris d'abord un ensemble de composition nucléotidique à 2 tuples T2 et un ensemble de composition nucléotidique à 3 tuples T3.

$$\begin{aligned} T_2 &= \{AA, AC, AG, AT, CC, CG, CT, GG, GT, TT\} \\ T_3 &= \{AAA, CCC, CCG, AAC, CCT, ...TTT\} \end{aligned} \tag{4.2}$$

— Pour les éléments de T2, je décris la mutuelle 2 tuples informations comme suit :

$$I(N_1N_2) = f(N_1, N_2) \ln \frac{f(N_1, N_2)}{f(N_1)f(N_2)} \tag{4.3}$$

— Pour les éléments de T3, je décris la mutuelle 3 tuples informations comme suit :

$$\begin{aligned} I(N_1N_2N_3) = f(N_1, N_2) \ln \frac{f(N_1, N_2)}{f(N_1)f(N_2)} \\ + \frac{f(N_1, N_3)}{f(N_3)} \ln \frac{f(N_1, N_3)}{f(N_3)} \\ - \frac{f(N_1, N_2, N_3)}{f(N_2)f(N_3)} \ln \frac{f(N_1, N_2, N_3)}{f(N_2)f(N_3)} \end{aligned} \tag{4.4}$$

Pour une section spécifique :
$f(N_i)$ est l'occurrence du nucléotide N_i dans cette section.
$f(N_i, N_j)$ et $f(N_i, N_j, N_k)$ sont les occurrences de 2 tuples et 3 tuples.

4.2.3 Choix et entraînement du modèle d'apprentissage

Cette phase nécessite le bon choix du modèle d'apprentissage, dans mon cas on a utilisé trois modèles (KNN, SVM et arbre de décision), on a entraîné les modèles, par la suite, on a réglé leurs hyper paramètres. Pour cela, le plan d'action est suivi en effectuant les étapes suivantes :

— Choisir le bon algorithme en fonction de l'objectif de l'apprentissage et de ses besoins en données ;
— Configurer et régler les hyper paramètres pour optimiser les performances, et choisir une méthode d'itération pour arriver aux meilleurs hyper paramètres ;
— Identifier les attributs qui fourniront les meilleurs résultats.

4.2.4 Évaluation des performances du modèle

Cette étape comprend l'évaluation des métriques du modèle, la matrice de confusion et les métriques de performance (Accuracy, Recall, Precision

Au cours du processus d'évaluation du modèle, on a procédé comme suit :

— Déterminer les valeurs de la matrice de confusion dans le cadre des problèmes de classification ;
— Identifier des méthodes de validation croisée ;
— Affiner les hyper paramètres pour optimiser la performance.

4.3 Réalisation de la contribution

4.3.1 Environnement de travail

Pour réaliser ma conception, on a fait appel aux outils informatiques nécessaires y compris simulateur Metasim et au langage de programmation adéquat qui est python.

4.3.1.1 Python

Python est un langage de programmation multiparadigme et multiplateformes. Il supporte la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes (Garbage Collector) et d'un système de gestion d'exceptions [50] .

La version utilisée pour effectuer mes simulations est python 3.

Python, en tant que langage seul, n'est pas spécialement plus adapté que les autres langages pour faire de l'apprentissage supervisé, mais Il dispose aussi de quelques librairies spécialisées en ce domaine qu'on a exploité.

4.3.1.2 Scikit Learn

Scikits Learn est une librairie d'apprentissage automatique couvrant l'ensemble de la discipline :

- Les types d'apprentissage : supervisé et non supervisé ;
- Les algorithmes :
  - Régression linéaire ;
  - Arbre de décision ;
  - SVM (machines à vecteur de support) ;
  - Classification naïve bayésienne ;
  - KNN (Plus proches voisins) ;
  - Réseaux de neurones.

Scikits Learn dispose d'une excellente documentation fournissant de nombreux exemples, d'une grande communauté et elle est très bien intégrée avec d'autres Librairies.

4.3.1.3 MetaSim

MetaSim est un simulateur conçu pour simuler le séquençage d'une seule séquence ADN ou d'un ensemble de séquences [51] .

MetaSim prend en entrée un ensemble de séquences génomiques connues et un profil d'abondance. Ce profil détermine quelles séquences génomiques sont sélectionnées pour la simulation et l'abondance relative de chaque séquence génomique dans l'ensemble de données.

Pour la construction d'un ensemble de données de lecture réaliste, MetaSim comprend un simulateur de séquençage de lecture polyvalent. L'utilisateur peut choisir parmi différents modèles d'erreur (adaptables) des technologies de séquençage actuelles (par exemple, Sanger, Roche's 454 et Illumina).

MetaSim fournit en sortie un fichier fasta qui comporte l'ensemble de lectures, résultat de séquençage.

MetaSim est écrit en Java et peut être exécuté avec un utilisateur en mode graphique ou ligne de commande, il est librement disponible pour Linux/Unix, MacOS X et Windows.

MetaSim est utilisé en exécutant les étapes suivantes :

1. Sélection des séquences du génome source à partir d'une base de données ;
2. Configuration du profil d'abondance des espèces en définissant le nombre relatif de copies des séquences du génome ;
3. Séquençage de prélèvements de fragments selon l'espèce profils d'abondance ;
4. Application de modèles d'erreurs spécifiques à la technologie aux fragments pour créer des lectures de séquençage.

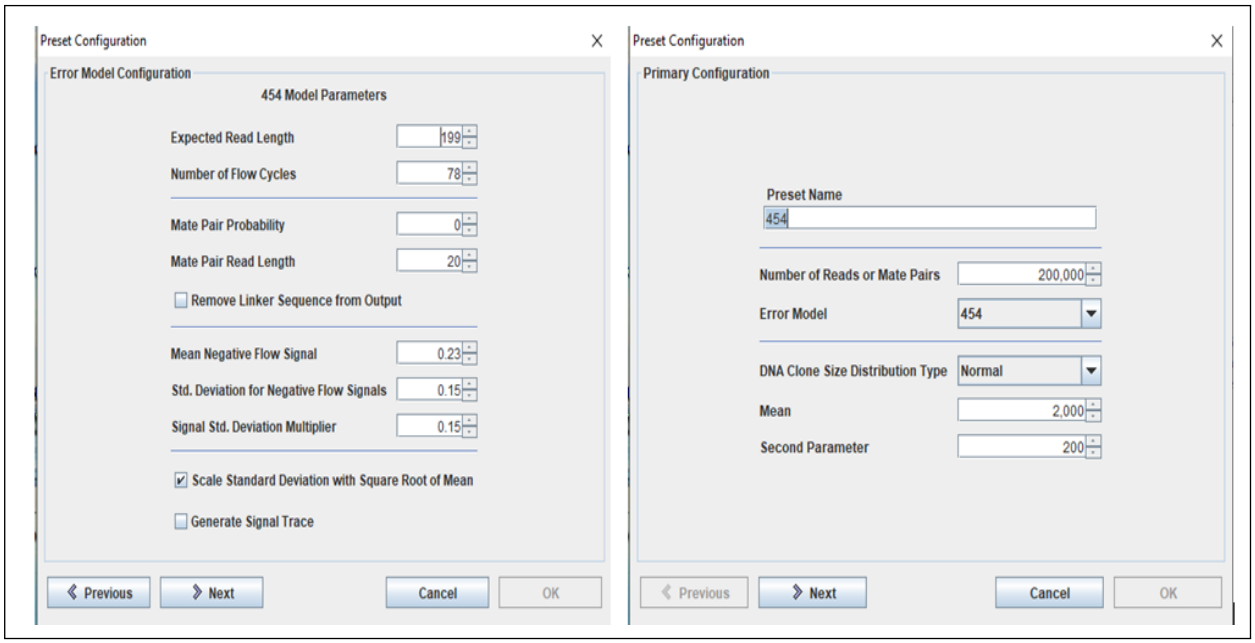

FIGURE 4.1: Interface graphique du MetaSim

4.3.2 Implémentation

Après avoir finalisé la conception de mon modèle, on va vous présenter dans cette partie le détaille de mes implémentations, en commençant par le traitement de données suivi par l'implémentation des trois modelés, arbre de décision, KNN et SVM.

4.3.2.1 Traitement de données

L'échantillon contaminé qu'on veut décontaminer est composé des génomes appartenant à l'espèce vibrio, pour cela on a travaillé sur des génomes vibrio de référence qu'on a téléchargé du site officielle de NCBI (National Center for Biotechnology Information), dans le but de concevoir mes data sets d'apprentissage, le tableau suivant résume les informations de ces génomes.

Les génomes sont introduits au simulateur MetaSim pour simuler l'opération du séquençage, MetaSim est configuré dans le séquençage 454 avec des tailles de lecteurs 200, 300, 400, 500 pb. Ensuit les fichiers résultants sont traités sous python pour extraire les 50 caractéristiques analytiques (K-Gram et MMI) en associant chaque lecteurs à son classe (cible=1 ; contaminant=0), la figure 4.2 démontre le processus de traitement de données.

4.3.2.2 Entraînement des Modèles

Trois modèles sont conçus pour la phase d'apprentissage, ils ont comme entrée un fichier csv des caractéristiques analytiques 70% de ces données sont des données d'apprentissage et 30 % sont des données de test, après avoir entraîné les modèles, on peut prédire les nouvelles séquences.la figure 4.3 présente le modèle d'apprentissage supervisé.

Référence de génome	Taille en pb	Taille en MO	Classe
Vibrio cholerae	4,030,944	4.031	cible
Vibrio alginolyticus	5,177,838	5.178	contaminant
Vibrio cincinnatiensis	3,800,816	3.801	contaminant
Vibrio fluvialis	4,827,733	4.828	contaminant
Vibrio furnissii	4,993,326	4.993	contaminant
Vibrio harveyi	5,881,490	5.881	contaminant
Vibrio metoecus	3,988,124	3.988	contaminant
Vibrio mimicus	4,313,453	4.313	contaminant
Vibrio parahaemolyticus	5,165,770	5.166	contaminant

TABLE 4.1: Configuration du MetaSim

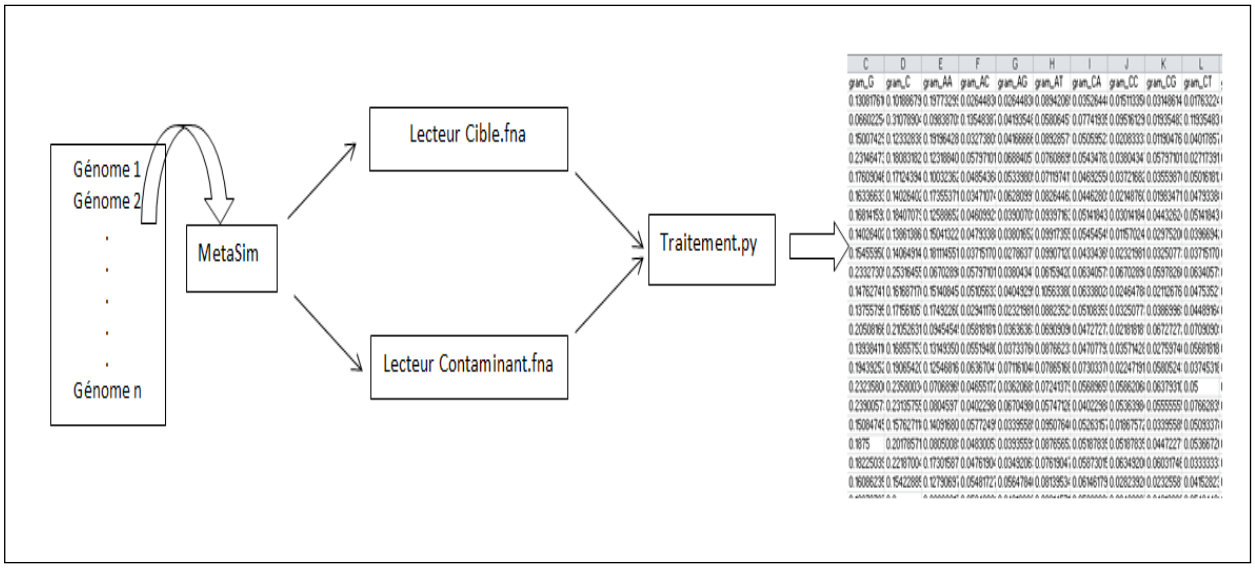

FIGURE 4.2: Processus de traitement de données

4.4 Conclusion

A la fin de ce chapitre, j'ai pu présenter ma contribution, toute en détaillant ses étapes principales, ses bases théoriques ainsi les outils informatiques utilisés pour développer les trois modèles KNN, SVM et arbre de décision, à savoir l'outil de simulation MetaSim et le langage de programmation python.

Dans le but de tester le fonctionnement de ma contribution, dans le chapitre suivant, j'ai effectuer les tests de performance nécessaires pour la validation, j'ai élaborer une comparaison quantitative entre les trois modèles conçus et un des méthodes présentées au niveau du chapitre état de l'art.

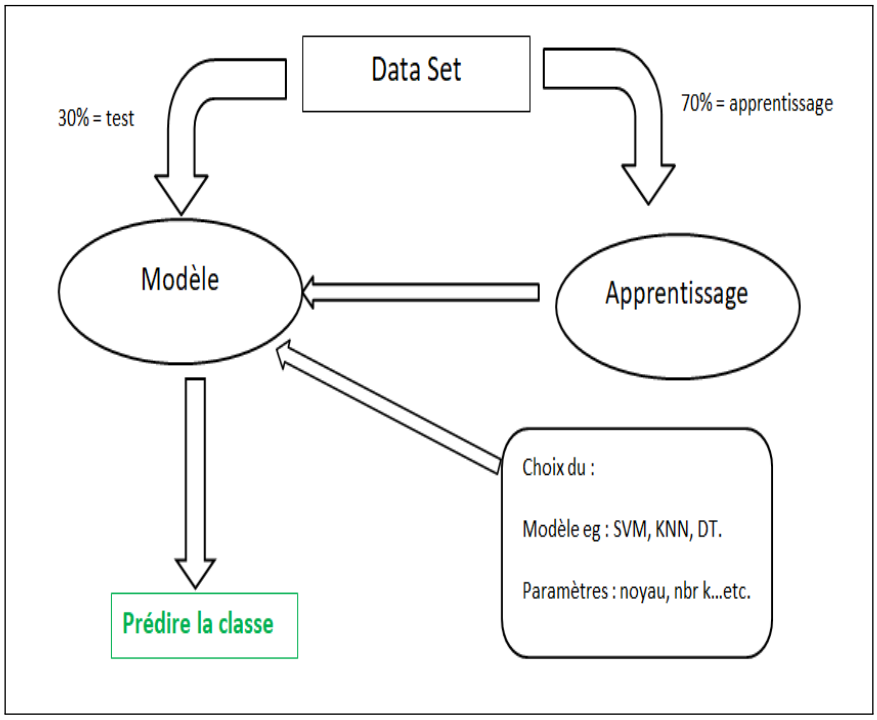

FIGURE 4.3: Modèle d'apprentissage

Chapitre

5 VALIDATION EXPÉRIMENTALE

Sommaire

5.1 Introduction

Dans l'objectif de valider mon approche pour décontaminer les échantillons d'un ADN contaminé, je présente dans ce chapitre une étude de performance quantitative (mesurer la précision, la sensibilité et la spécificité du modèle d'apprentissage).je mene dans ce chapitre un processus de validation de mon approche détaillée dans le chapitre précèdent, toute en présentant les différents tests sur les trois méthodes réalisées d'apprentissage supervisé : arbre de décision, KNN et SVM ainsi les data sets utilisés pour entraîner les modelés.

Une comparaison quantitative sera présentée, entre d'un part les trois modèles et l'arbre de décision présenté dans le chapitre état de l'art d'autre part, l'objectif derrière cette comparaison est de déterminer et de valider la meilleure configuration et la meilleure méthode pour décontaminer un échantillon contaminé. Une deuxième comparaison sera présentée entre les trois modèles de ma contribution en utilisant mon data set dans l'objectif d'étudier les performances de mes modèles sur mes données.

5.2 Étude comparative entre ma contribution et l'état de l'art

Dans cette partie, je présente une comparaison quantitative en s'appuyant sur les mesures de performance entre ma contribution et la méthode de décontamination (arbre de décision) [49] , pour cela j'ai reconstruit le même environnement de test comparant à la méthode présentée dans le chapitre état de l'art [49] , à savoir le même data set d'apprentissage et les mêmes tests.

5.2.1 Description du data set

Le data set est compose de deux classes, classe cible et contaminant, la classe cible est représenté par le génomes Caenorhabditis remane , Caenorhabditis elegans et Caenorhabditis briggsae , d'autre part les génomes Escherichia co , Chryseobacterium sp , Stenotrophomonas maltophilia , Stenotrophomonas rhizophil . . . etc, appartient aux contaminants.

— Le nombre des séquences ADN de la classe "cible" est de : 200000 séquences, avec un pourcentage de : 57,14% ;

— Le nombre des séquences ADN de la classe "contaminant" est de : 150000 séquences avec un pourcentage de : 42,86%.

La taille de chaque séquence est variée de 200 à 600 paires de bases.

Le nombre d'attributs de ce data set est de 50 attributs plus l'attribut classe qui représente la classe de chaque séquence.

5.2.2 Expérimentations et résultats

Plusieurs protocoles de test sont effectués pour étudier les performances des trois méthodes réalisées dans ce travail et la méthode décrite dans la partie état de l'art, ces comparaisons ont pour objectif la validation de ma contribution.

Pour chaque test, une matrice de confusion est affichée et trois mesures de performance sont calculées, dans le but de tester la performance d'un modèle donné :

— Précision (Accuracy) : indique le pourcentage de vraies prédictions.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{5.1}$$

— Sensibilité (Recall) : indique le pourcentage de vrais positifs.

$$Recall = \frac{TP + TN}{TP + FN} \tag{5.2}$$

— Spécificité (specificity) : indique le pourcentage de vrais négatifs.

$$specificity = \frac{TP}{TP + FP} \tag{5.3}$$

Vrais Positifs (TP)	**Faux Positifs (FP)**
Faux Négatifs (FN)	**Vrais Négatifs (TN)**

TABLE 5.1: Matrice de confusion

Modèle	**rbf (IMM + K-gram)**	**rbf (IMM)**	**rbf (K-gram)**	**linéaire (IMM + K-gram)**	**linéaire (IMM)**	**linéaire (K-gram)**
Précision %	99.34	97.69	95.31	**99.84**	85.60	91.42
Sensibilité %	99.11	98.29	97.30	**99.91**	90.87	94.67
Spécificité %	99.73	97.69	94.64	**99.81**	84.95	90.65

TABLE 5.2: Variation de la performance en fonction de noyau et d'attributs utilisés

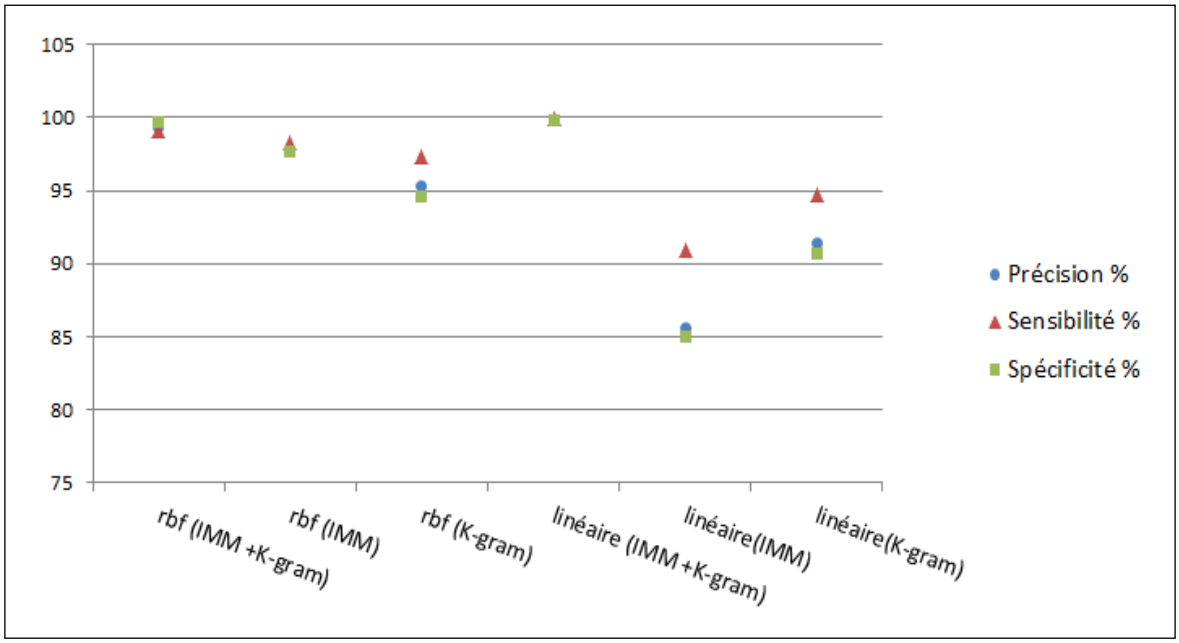

FIGURE 5.1: Variation de la performance en fonction de noyau et d'attributs utilisés

5.2.2.1 Expérimentation 01

Dans cette expérience, j'effectue les tests de performances sur le modèle d'apprentissage SVM. Pour cela, j'étudis l'impact de noyau utilisé d'un SVM (rbf ou linéaire), ainsi que le choix d'attributs (IMM ou K-gram ou les deux).

Le tableau 5.2 présente les variations des mesures de performance du modèle en fonction du noyau utilisé et les attributs K-gram, IMM.

La figure 5.1 montre que , d'un coté l'utilisation d'un seul groupe d'attribut (IMM ou K-gram) dégrade d'une manière remarquable les performance du modèle d'apprentissage. Contrairement à l'utilisation de deux groupes IMM et K-gram en même temps.

D'autre coté les résultats obtenues pour un noyau linéaire sont similaires lors de l'utilisation du noyau rbf.

5.2.2.2 Expérimentation 02

Dans cette expérience, j'éffectue les tests de performances sur le modèle d'apprentissage KNN. Pour cela, j'étudis l'impact de nombre de voisions (k) sur la performance du modèle, ainsi que le choix d'attributs (IMM ou K-gram ou les deux).

Le tableau 5.3 présente les variations de la précision du modèle en fonction du nombre k de voisins et les attributs K-gram, IMM.

Valeur de k	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Précision (K-gram et IMM) %	95.60	96.59	96.63	96.60	**96.69**	96.58	96.67	96.57
Précision (IMM) %	96.14	96.31	96.30	96.37	**96.44**	96.35	96.40	96.35
Précision (K-gram) %	93.72	94.94	95.15	95.12	95.31	95.19	**95.33**	95.19

TABLE 5.3: Variation de la précision en fonction de nombre de voisins k et d'attributs utilisés

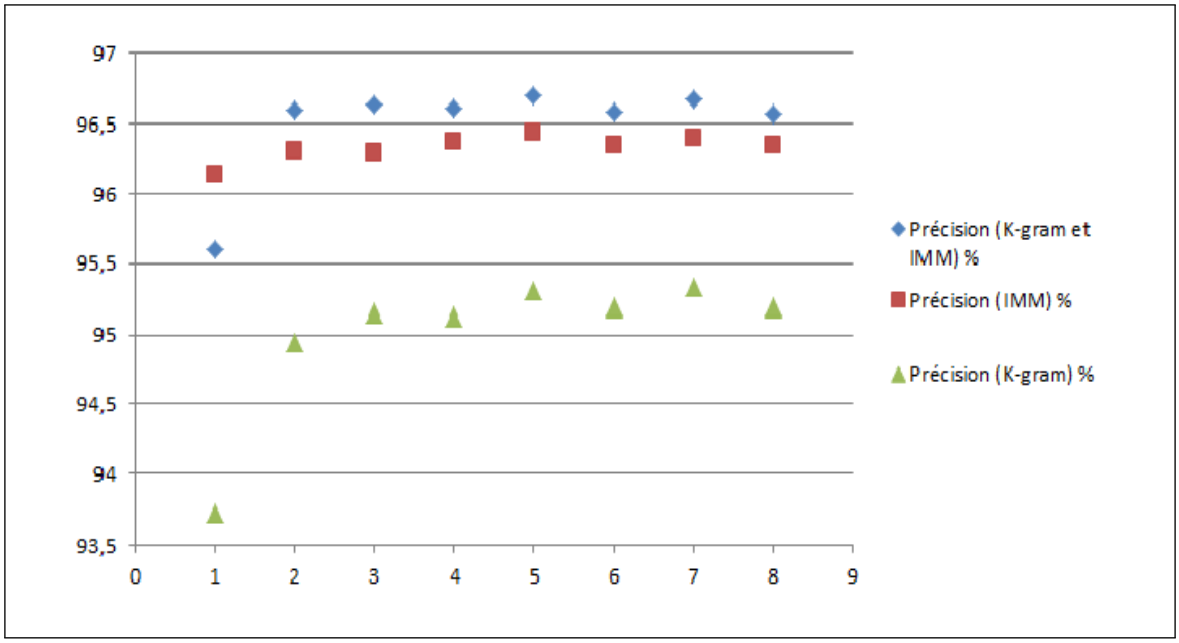

FIGURE 5.2: Variations de la précision du modèle en fonction du nombre k de voisins

La figure 5.2 montre que, d'un part l'augmentation de nombre de voisins considérés dans mon apprentissage, de k égale à 1 jusqu'à k égale à 5 augmente les performances du modèle, tandis que ces performances diminue pour k de 6 à 8.

D'autre part les résultats sont plus importants en utilisant le couple d'attribut IMM et K-gram.

5.2.2.3 Expérimentation 03

Dans cette expérience, j'effecte les tests de performances sur le modèle d'apprentissage arbre de décision. Pour cela, j'étudis l'impact du choix d'attributs (IMM ou K-gram ou les deux) sur la performance du modèle.

Le tableau 5.4 présente les variations des mesures de performance du modèle en fonction d'attributs K-gram, IMM.

Attributs	IMM et K-gram	IMM	K-gram
Précision %	**96.36**	93.92	94.86
Sensibilité %	**97.31**	95.48	96.29
Spécificité %	**96.35**	93.97	94.79

TABLE 5.4: Variation de la performance en fonction d'attributs utilisés

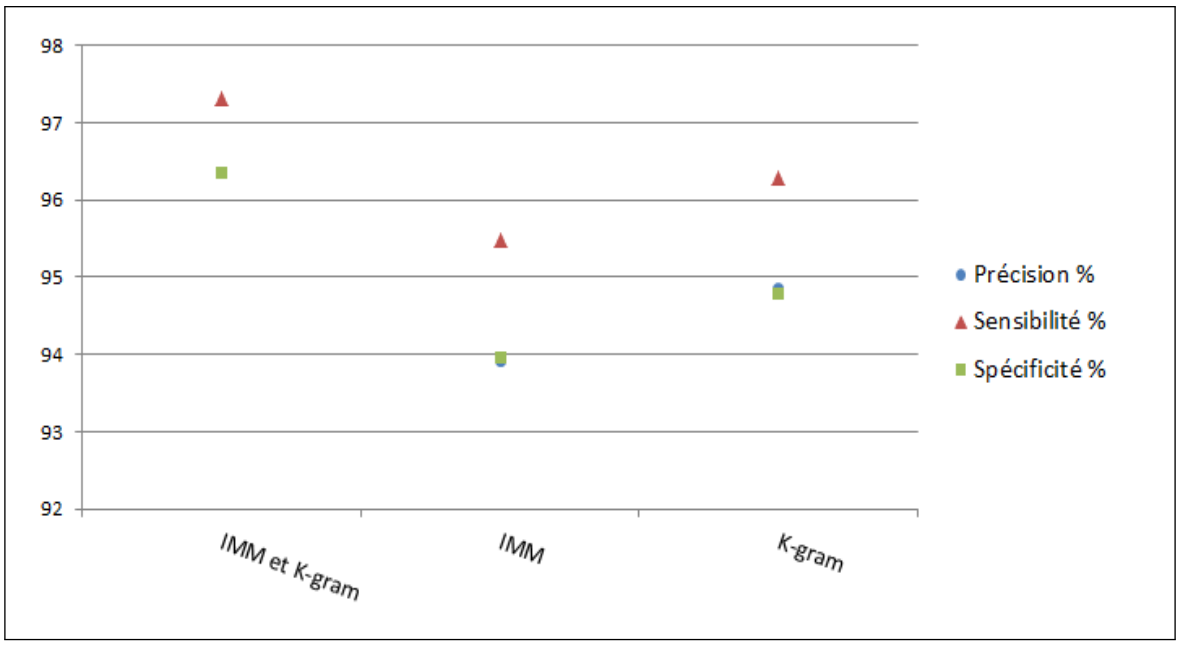

FIGURE 5.3: Variations des mesures de performance du modèle en fonction d'attributs

La figure 5.3 montre que, en utilisant le couple d'attributs IMM et K-gram augment les performances de mon modèle (arbre de décision), contrairement dans le cas de l'utilisation d'un seule groupe d'attribut.

5.2.2.4 Discussion des Résultats

En examinant les résultats des trois précédentes expériences :

— Les résultats de la méthode, présentée dans le chapitre état de l'art [49] , qui exploite les arbres de décision, sont exprimés ci-dessous :

  — La précision de ce modèle est supérieure à 99%.
  — La sensibilité et la spécificité sont aussi supérieure à 99%.

— tableau 5.5 présente une comparaison quantitative entre les performances maximales des trois modèles conçus.

— Dans le cas de mon modèle KNN, je remarque que la précision est entre 95% et 97%, elle est la plus importante lors de l'utilisation de deux attributs IMM et K-gram en même temps. Elle est moins importante si le groupe d'attributs IMM est utilisé. Elle a la valeur minimale en utilisant le groupe d'attributs K-gram. mes remarques concernant la précision du modèle, sont valables pour les autres métriques de performance à savoir la spécificité et sensibilité.

— Dans le cas de mon modèle arbre de décision, je remarque que la précision est entre 94% et 98%, elle est la plus importante lors de l'utilisation de deux attributs IMM et K-gram

Modèle	**SVM**	**KNN**	**Arbre de décision**
Précision Max (%)	99.84	96.69	96.36
Sensibilité Max (%)	99.91	96.44	97.31
Spécificité Max (%)	99.81	95.33	96.35

TABLE 5.5: Tableau comparatif des performances maximales des trois modèles

en même temps. Elle est moins importante si le groupe d'attributs K-gram est utilisé. Elle a la valeur minimale en utilisant le groupe d'attributs IMM. mes remarques concernant la précision du modèle, sont valables pour les autres métriques de performance à savoir la spécificité et sensibilité.

— mon modèle SVM est le modèle le plus performant de mes trois modèles, je remarque que la précision est autour de 99% en utilisant le noyau rbf comme en utilisant le noyau linéaire avec l'utilisation du couple d'attributs IMM et K-gram. Les mêmes remarques sont constatées pour la valeur de la spécificité ainsi la valeur de la sensibilité. Les performances du modèle diminuent à l'utilisant d'un des deux groupes IMM et K-gram séparément.

— D'après les constatations, le modèle SVM est le modèle le plus performant des trois modèles, elle est comparable au modèle de référence cité dans la partie état de l'art.

— Donc, je peux valider le modèle SVM, en se basant sur les résultats expérimentaux. Dans la partie suivante du travail, je vais le tester sur mon data set.

5.3 Étude de performance du modèle SVM sur mon data set

Dans cette partie je vais utiliser le modèle SVM, validé auparavant sur mon data set dans l'objectif d'étudier l'efficacité et le comportement de ce modèle face à mes données.

5.3.1 Data set

Le data set est composé de deux classes, classe cible et contaminant, la classe cible est représenté par le génome vibrio colorea et les autres vibrio appartient aux contaminants.

— Le nombre des séquences d'ADN de la classe "cible" est de : 152024, avec un pourcentage de : 59.81% ;

— Le nombre des séquences d'ADN de la classe "contaminant" est de : 134365, avec un pourcentage de : 40.19% .

Le nombre d'attributs de ce data set est de 50 attributs plus l'attribut classe qui représente la classe de chaque séquence.

5.3.2 Expérimentations et résultats

Dans cette expérience, j'ai effectué les tests de performances sur le modèle validé auparavant à savoir SVM pour mes données, dans l'objectif d'étudier le comportement de mon modèle face à mes données.

Le tableau 5.6 présente les variations des mesures de performance du modèle en fonction du noyau utilisé et les attributs K-gram, IMM.

Modèle	**rbf (IMM + K-gram)**	**rbf (IMM)**	**rbf (K-gram)**	**linéaire (IMM + K-gram)**	**linéaire (IMM)**	**linéaire (K-gram)**
Précision %	93.36	79.98	65.16	**93.55**	77.46	61.67
Sensibilité %	87.80	77.92	73.33	**87.86**	80.52	70.67
Spécificité %	99.65	83.28	65.30	**100**	77.96	62.24

TABLE 5.6: Variation de la performance en fonction de noyau et d'attributs utilisés

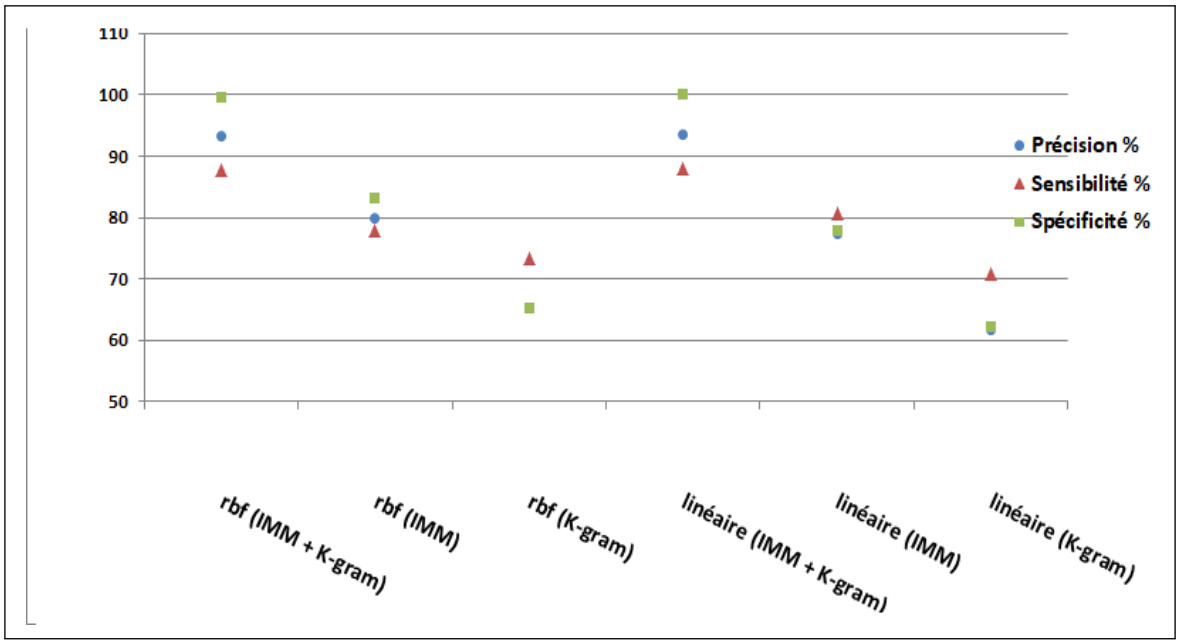

FIGURE 5.4: Variation de la performance en fonction de noyaux et d'attributs utilisés

La figure 5.4 montre que , d'un coté l'utilisation d'un seul groupe d'attribut (IMM ou K-gram) dégrade d'une manière remarquable les performance du modèle d'apprentissage. Contrairement à l'utilisation de deux groupes IMM et K-gram en même temps. D'autre coté les résultats obtenues se rapprochent en utilisant le noyau rbf ou linéaire .

5.3.2.1 Discussion des Résultats

En examinant les résultats de cette expérience :

— Le modèle SVM répond bien au problème de contamination des séquences ADN, pour mes données d'apprentissage. j'ai remarqué que le modèle réussi à réaliser ses valeurs maximales, pour un noyau linéaire et pour le couple d'attributs K-gram et IMM.Pour ce cas, la précision est à 93.55%, la sensibilité est à 87.86%, la spécificité est à 100%.

— Pour les autres configurations, le modèle réalise des mesures de performances moins importantes, à l'exception de la configuration, noyau rbf et couple d'attributs K-gram et IMM qui s'approche au modèle le plus performant.

— j'ai remarqué que les résultats obtenus pour mes données sont moins importants que lors de l'utilisation de data set de test. Cette diminution de performances est due à la composition de mon data set :

— La classe contaminant et la classe cible appartient au même espèce qui est vibrio, contrairement aux séquences du premier data set, qui sont des génomes de différentes espèces, donc une bonne séparation contrairement à mes données.
— Les classes de même espèce sont plus difficiles à séparer, ça n'empêche pas à juger que les résultats réalisés sont satisfaisants.

5.4 Application du décontamination d'un échantillon ADN

j'ai réalisé cette application python dans l'objectif d'automatiser les opérations de prétraitement de données, d'apprentissage supervisé et de prédiction de nouvelles séquences, et donner à l'utilisateur la possibilité d'effectuer la construction du modèle très simplement en utilisant une interface graphique. La figure 5.5 présent l'application réalisée : L'application est devisée en

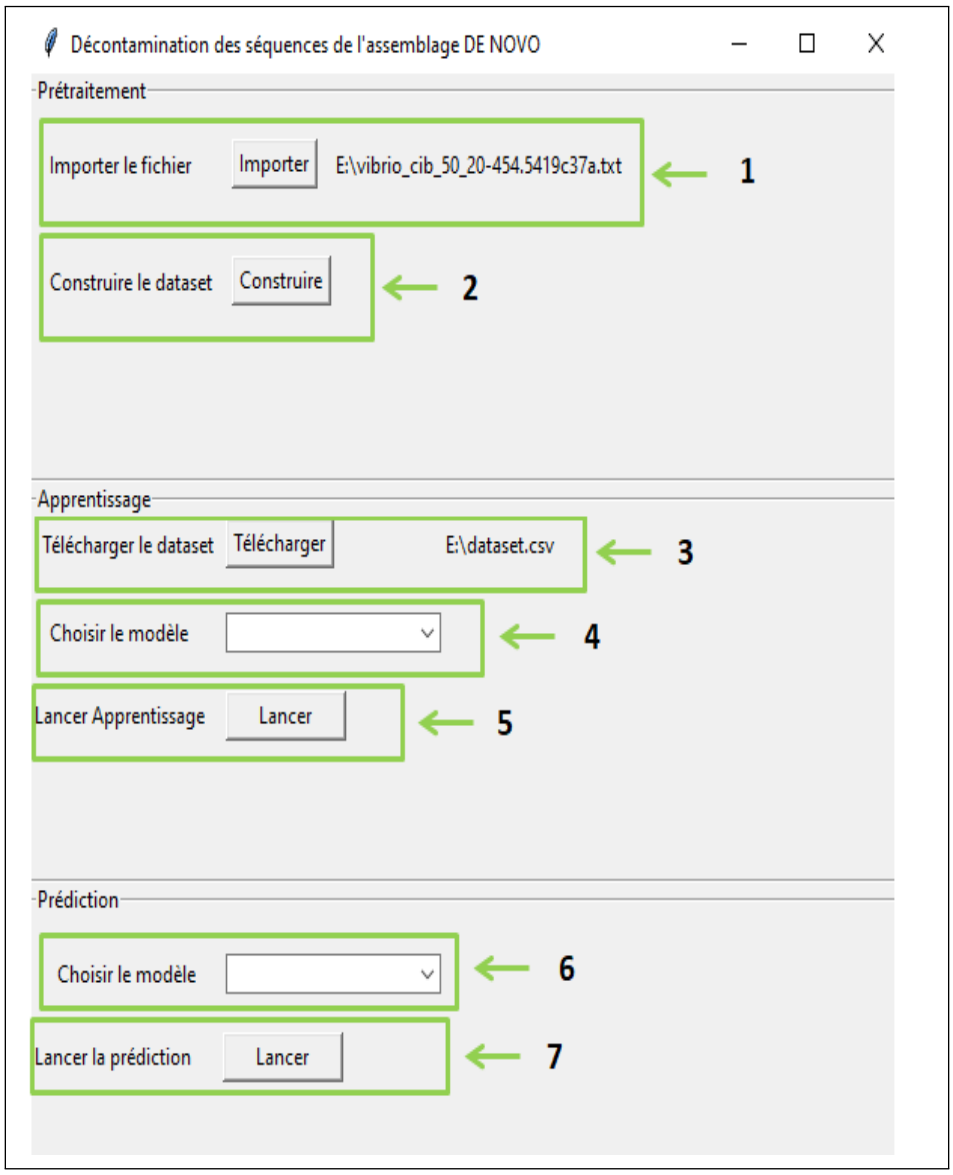

FIGURE 5.5: Interface graphique de l'application de décontamination

trois étapes :

— **Prétraitement des données** : cette étape a pour objectif, la construction du data set, en passant d'un fichier texte qui représente les séquences d'ADN à un fichier csv qui

représente l'enregistrement qui comporte les 50 attributs (K-gram et IMM) et la classe de chacune de ses séquences ADN, cette opération comporte :

1. Télécharger le fichier texte qui comporte les séquences ADN (élément 1) ;
2. Construire le data set, faire appel aux fonctions qui calculent les attributs K-gram et IMM de chaque séquence, ensuite, construire un fichier csv qui englobe toutes ces mesures d'attributs (élément 2).

— **Apprentissage** : cette étape a pour objectif, l'entraînement du modèle (SVM, KNN ou arbre de décision), en suivant les étapes suivantes :

1. Télécharger le data set d'apprentissage (fichier csv) (élément 3) ;
2. Choisir le modèle d'apprentissage entre SVM, KNN et arbre de décision (élément 4) ;
3. Lancer l'opération d'apprentissage (élément 5).

— **Prédiction** : cette étape a pour objectif, la prédiction et la classification de nouvelles séquences (contaminant ou cible), elle s'effectue en étapes suivantes :

1. Télécharger le fichier texte comportant les nouvelles séquences à prédire (élément 5) ;
2. Lancer l'opération de la prédiction (élément 6).

5.5 Conclusion

Dans ce chapitre, j'ai validé ma contribution, en mesurant les performances de chaque modèle d'apprentissage parmi les trois modèles réalisés. j'ai effectué une étude comparative de ma contribution, à la fois avec la méthode décrite dans l'état de l'art, et une comparaison entre ces trois modèles (SVM, KNN et arbre de décision).

A la fin de ce chapitre, j'ai validé la méthode d'apprentissage supervisée SVM qui utilise les 50 attributs (IMM et K-gram) à la fois.

j'ai réalisé une application python, dans l'objectif d'automatiser le processus d'apprentissage, et de simplifier son utilisation par un simple utilisateur.

.

Conclusion Générale et Perspectives

j'ai montré au cours de mon étude le bien-fondé de l'utilisation des techniques d'apprentissage supervisé, en particulier les SVM dans le cadre de la décontamination des séquences d'ADN, l'étude concerne les séquences d'ADN contaminées issues d'un séquençage à haut débit, dans le but de séparer les séquences d'ADN en deux classes, une classe cible représentée par le génome vibrio cholerae et une classe des contaminants qui regroupe l'ensemble des autres génomes vibrio.

Afin d'élaborer une décontamination fondée je passe essentiellement par trois phases : l'acquisition des séquences d'ADN, le traitement des séquences qui inclut l'extraction des caractéristiques (IMM et K-gram) et enfin la prise de décision où la classification des séquences d'ADN après l'extraction des attributs. Les travaux ont été menés à partir d'expérimentations réalisées sur des séquences d'ADN contaminées issues d'un simulateur de séquençage à haut débit (MétaSim).

Pour aborder l'étude, j'ai en premier lieu présenté dans le premier et le deuxième chapitre, des généralités sur le contexte biologique et bioinformatique liées à ce problème dans l'objectif d'introduire et de mieux comprendre les travaux effectués.

Une étude bibliographique a été menée dans le troisième chapitre sur les techniques et les méthodes utilisées pour résoudre le problème de contamination des séquences d'ADN, une taxonomie de ces méthodes a été réalisée pour les classer dans deux catégories, des méthodes indépendantes des bases de données et des méthodes dépendantes.

Dans le quatrième chapitre, une présentation détaillée a été élaborée de la méthode de décontamination basée sur les techniques d'apprentissage supervisé, trois modèles de décontamination ont été réalisées pour résoudre le problème.

Dans le dernier chapitre une étude de performances a été effectuée afin de mesurer l'efficacité des modèles réalisés et de comparer d'un côté, entre eux et d'autre coté avec une méthode d'apprentissage supervisé existante dans la littérature. Les résultats obtenus des expériences nous permettent de conclure que le modèle SVM qui exploite les attributs IMM et K-gram est le plus performant des trois modèles. Une application python a été réalisée pour mettre en exploitation d'une manière simplifie le processus de la décontamination des séquences d'ADN.

Dans le but de poursuivre les travaux dans cette direction, il serait également nécessaire d'augmenter les bases de données d'apprentissage et d'utiliser des données réelles.

BIBLIOGRAPHIE

[1] **E**rnst Mayr. The Growth of Biological Thought : Diversity, Evolution, and Inheritance. en. Google-Books-ID : pHThtE2R0UQC. Harvard University Press, 1982.

[2] **C** R Woese et G E Fox, Phylogenetic Structure of the Prokaryotic Domain : The Primary Kingdoms. In : Proceedings of the National Academy of Sciences of the United States of America 74.11 (nov. 1977), p. 5088–5090. issn : 0027-8424.

[3] **P**hilipe LUCHETTA, L'essentiel de biologie moléculaire.

[4] **h**ttps ://fr.wikipedia.org/wiki/Acide-dC3%A9soxyribonucl%C3%A9ique

[5] **W**illi Hennig, Grundzüge einer Theorie der phylogenetischen Systematik, Deutscher Zentralverlag, Berlin, 1950.

[6] **N**icholas Hudson, From "Nation to "Race" : The Origin of Racial Classification in Eighteenth-Century Thought pub in : 1996.

[7] **A**TTIGNON V et al, SÉQUENÇAGE DE NOUVELLE GÉNÉRATION D'UN PANEL DE GÈNES POUR L'ANALYSE EN GÉNÉTIQUE SOMATIQUE /Validation de la méthode, Institut National du Cancer France pub : 2016.

[8] **F**. Sanger, S. Nicklen et A. R. Coulson, DNA Sequencing with Chain-Terminating Inhibitors, pub : Proceedings of the National Academy of Sciences (1977).

[9] **J**eantine E. Lunshof, PhD ; Jason Bobe, MS ; John Aach et al, Personal genomes in progress : from the Human Genome Project to the Personal Genome Project (2010).

[10] **H**agan Bayley, Sequencing single molecules of DNA (2006).

[11] **A**llan M. Maxam et Walter Gilbert, « A new method for sequencing DNA », Proc. Natl. Acad. Sci. USA, vol. 74, 1977, p. 560-564.

[12] **h**ttps ://parlonssciences.ca/ ressources-pedagogiques/ documents-dinformation/ sequencage-de-sanger.

[13] **E**rwin van Dijk, Claude Thermes, La révolution de la génomique : les nouvelles méthodes de séquençage et leurs applications, pub in 2021.

[14] **S**acha Schutz, Le séquençage de nouvelle génération.

[15] **J**erzy K. Kulski, Next-Generation Sequencing — An Overview of the History, Tools, and "Omic" Applications (2016).

[16] **h**ttps ://www.ncbi.nlm.nih.gov/sra /docs/submitformats/ fasta-with-qual-file-pairs, Accessed : 20-08-2021.

[17] **h**ttps ://www.ncbi.nlm.nih.gov/sra/ docs/submitformats/ fastq-files, Accessed : 20-08-2021.

[18] **P**ierre Pericard, Algorithmes pour la reconstruction de séquences de marqueurs conservés dans des données de métagénomique, pub in :2018.

[19] **S**. B. Needleman et C. D. Wunsch, A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins, In : Journal of Molecular Biology (1970), p. 443–453.

[20] **T**. F. Smith et M. S. Waterman, Identification of Common Molecular Subsequences . IN : Journal of Molecular Biology (1981), p. 195–197.

[21] **S**tephen F. Altschul et al, Basic Local Alignment Search Tool, In : Journal of Molecular Biology 215.3 (oct. 1990), p. 403–410.

[22] **B**en Langmead et al, Ultrafast and Memory-Efficient Alignment of Short DNA Sequences to the Human Genome ». In : Genome Biology 10 (mar. 2009).

[23] **H**eng Li et Richard Durbin, Fast and Accurate Short Read Alignment with Burrows–Wheeler Transform, In : Bioinformatics 25.14 (juil. 2009), p. 1754–176.

[24] **E**vguenia Kopylova et a,. SortMeRNA 2 : ribosomal RNA classification for taxonomic assignation, In : Workshop on Recent Computational Advances in Metagenomics, ECCB 2014. 2014.

[25] **N**icolas Maillet, Comparaison de novo de données de séquençage issues de très grands échantillons métagénomiques : application sur le projet Tara Oceans, pub :2013.

[26] **N**iranjan Nagarajan et Mihai Pop, Sequence Assembly Demystified, In : Nature Reviews Genetics 14.3 (jan. 2013), p. 157–167.

[27] **X**iaoqiu Huang et Anup Madan, CAP3 : A DNA Sequence Assembly Program, In : Genome Research 9.9 (1999), p. 868–877.

[28] **W**illiam R. Jeck et al, Extending Assembly of Short DNA Sequences to Handle Error, In : Bioinformatics 23.21 (2007), p. 2942–2944.

[29] **U**di Manber et Gene Myers, Suffix Arrays : A New Method for On-Line String Searches, In : Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms. SODA '90. Philadelphia, PA, USA : Society for Industrial and Applied Mathematics, 1990, p. 319–327.

[30] **R**. Grossi et J. Vitter, Compressed Suffix Arrays and Suffix Trees with Applications to Text Indexing and String Matching, In : SIAM Journal on Computing 35.2 (jan. 2005), p. 378–407.

[31] **J**ared T. Simpson et Richard Durbin, Efficient Construction of an Assembly String Graph Using the FM-Index, In : Bioinformatics 26.12 (juin 2010), p. i367–i373. issn : 1367-4803.

[32] **E**ugene W. Myers et al, A Whole-Genome Assembly of Drosophila, In : Science 287.5461 (mar. 2000), p. 2196–2204

[33] **J**ared T. Simpson et Richard Durbin, Efficient de Novo Assembly of Large Genomes Using Compressed Data Structures, In : Genome Research 22.3 (jan. 2012), p. 549–556.

[34] **P**avel A. Pevzner, Haixu Tang et Michael S. Waterman, An Eulerian Path Approach to DNA Fragment Assembly, In : Proceedings of the National Academy of Sciences of the United States of America 98.17 (août 2001), p. 9748–9753.

[35] **M**ichael C. Schatz, Arthur L. Delcher et Steven L. Salzberg. « Assembly of Large Genomes Using Second-Generation Sequencing ». en. In : Genome Research 20.9 (jan. 2010), p. 1165–1173.

[36] **J**ean Claude Heudin, directeur du laboratoire de recherche de l'IIM , Intelligence artificielle.

[37] **B**aştanlar, Y., Özuysal, M. (2014). Introduction to machine learning. miRNomics : MicroRNA Biology and Computational Analysis, 105-128.

[38] **F**atma Karem, Mounir Dhibi, Arnaud Martin. Combinaison de classification supervisée et non-supervisée par la théorie des fonctions de croyance.

[39] **Z**hongheng Zhang, Introduction to machine learning : k-nearest neighbors, pub : 2016.

[40] **S**ENOUSSI Hafida, Sélection de Données pour l'Apprentissage des Réseaux de Neurones, Arbres de Décision et les k-Plus Proches Voisins : Application en Diagnostic de Pannes, pub : 2015.

[41] **V**apnik V.N., « The Nature of Statistical Learning Theory », Springer-Verlag, New York,1995.

[42] **D**jeffal A, « Utilisation des méthodes support vector machine (SVM) dans l'analyse des bases de données », thèse de doctorat, Université Mohamed Khider, Biskra, Algérie, 2012.

[43] **A**ltschul SF, Gish W, Miller W, Myers EW, Lipman DJ, Basic local alignment search tool, pub : 1990 ;215 :403–410.

[44] **S**chmieder R, Edwards R, Fast identification and removal of sequence contamination from genomic and metagenomic datasets pub : mars 2011.

[45] **L**i H, Durbin R. Fast and accurate long-read alignment with burrows-wheeler transform. Bioinformatics,pub : 2010 ;26 :589–595.

[46] **L**u J, Salzberg SL. Removing contaminants from databases of draft genomes, pub : 2018.

[47] **R**enaud G, Slon V, Duggan AT, Kelso J. Schmutzi : Estimation of contamination and endogenous mitochondrial consensus calling for ancient DNA, pub : 2015 ;16 :224.

[48] **L**afond-Lapalme J, Duceppe MO, Wang S, Moffett P, Mimee B. A new method for decontamination of de novo transcriptomes using a hierarchical clustering algorithm. Bioinformatics, pub : 2017 May.

[49] **F**ierst, J.L., Murdock, D.A. Decontaminating eukaryotic genome assemblies with machine learning, pub : BMC Bioinformatics 18, 533 (2017).

[50] **p**ython. https ://www.python.org. Accessed : 01-08-2021.

[51] **D**aniel C. Richter ,Felix Ott, Alexander F. Auch, Ramona Schmid, Daniel H. Huson, A Sequencing Simulator for Genomics and Metagenomics, pub : October ,2008.