UCLouvain

École polytechnique de Louvain

Utilisation d'outils de classification supervisée et de pattern matching pour la discrimination chromosomique et plasmidique de séquences d'ADN bactérien

Auteur: **Maxime Bonjean**
Promoteurs: **Axel Legay, Jérôme Ambroise**
Lecteurs: **Jean-Luc Gala, Bernadette Govaerts**
Année académique 2020–2021
Master [120] : ingénieur civil biomédical

Remerciements

Tout d'abord, je tiens à remercier Jérôme Ambroise, co-promoteur de ce mémoire, pour sa disponibilité et son accompagnement pédagogique. J'ai eu la chance d'apprendre énormément auprès de lui.

Je souhaite également remercier mon promoteur, Axel Legay ainsi que mes lecteurs, Bernadette Govaerts et Jean-Luc Gala pour l'intérêt manifesté à l'encontre du sujet de ce mémoire ainsi que pour leurs remarques constructives.

Je remercie l'entreprise Syngulon pour l'acceuil chaleureux lors de mon stage et pour le partenariat durant ce mémoire.

Ensuite, je souhaite remercier chaleureusement ma compagne, Sophie, mes parents ainsi que mes beaux-parents qui m'ont toujours soutenu et accompagné durant mon cursus universitaire.

Enfin, je tiens à remercier Alexandre, Grégoire et Augustin pour avoir égayé ces années d'études.

Abstract

Ce mémoire porte sur l'utilisation d'outils de classification supervisée. Initialement, nous souhaitions les utiliser afin de découvrir de nouvelles bactériocines dans les génomes du Vibrio Cholerae, nous avons dû changer l'axe du mémoire suite à la trop grande complexité de ce domaine. Nous nous sommes dès lors intéressés à l'utilisation d'outils de classification supervisée afin de prédire si une séquence provient du chromosome ou du plasmide pour les espèces bactériennes suivantes : Escherichia Coli et Vibrio Cholerae. En effet, le séquençage actuel (NGS) ne permet pas de déterminer l'origine chromosomique ou plasmidique des contigs qu'il génère. Or, si un gène est découvert dans un de ces contigs, il est important de savoir si ce gène fait partie d'un chromosome ou d'un plasmide. Grâce à cette information, nous pourrons être capables de connaitre le mode de transmission de ce gène qui diffère en fonction de sa localisation. Nous avons dès lors recouru à une méthode basée sur des k-mers afin de caractériser des séquences de chromosomes et de plasmides. Ensuite, nous avons utilisé des outils de classification supervisée afin de faire de la prédiction sur l'origine des séquences provenant du séquençage. Les outils de classification supervisée utilisés sont le SVM et la régression logistique. Afin d'optimiser les classificateurs, nous avons testé différents paramètres pour obtenir les paramètres les plus performants. Enfin, les résultats obtenus ont été comparés à la littérature. L'utilisation de k-mers avec les outils de classification SVM et régression logistique est performante dans la discrimination chromosomique ou plasmidique de contigs provenant de V. Cholerae. Cependant, les résultats obtenus dans ce mémoire concernant Escherichia Coli sont moins concluants que ceux obtenus par mlplasmid.

Abréviations

ACP - Analyse en Composante Principale

ADN - Acide DésoxyriboNucléique

dNTP - désoxyribonucléotides-triphosphates

ddNTP - didéoxyribonucléotides triphosphates

HMM - Hidden Markov Model

OFR - Opening Reading Frame

OTU - Unité Taxonomique Opérationnelle

PB - Paire de Bases

RBF - Radial basis function

SMO - sequential minimal optimization

Table des matières

Chapitre 1

Introduction

« La résistance aux antibiotiques constitue aujourd'hui l'une des plus graves menaces pesant sur la santé mondiale, la sécurité alimentaire et le développement », constate l'Organisation Mondiale de la Santé (OMS, 2020)(1). Les antibiotiques sont de moins en moins performants contre les bactéries car celles-ci deviennent de plus en plus résistantes. Dans les prochaines décennies, la principale cause de mortalité sera l'infection bactérienne. La découverte de nouvelles bactériocines serait potentiellement une alternative aux antibiotiques (2).

Ce mémoire prend place dans la continuité du stage en entreprise réalisé durant mon cursus universitaire. Ce stage en entreprise a été effectué au sein de l'entreprise Syngulon, spécialisée dans la microbiologie et plus particulièrement dans les bactériocines. Une chercheuse au sein de Syngulon, Anandi Martin PhD, réalise une étude sur les bactériocines dans le V. Cholerae. Il nous a alors été proposé de réaliser ce mémoire sur ce sujet passionnant.

1.1 Objectifs du mémoire

L'objectif initial de ce mémoire était la découverte de nouvelles bactériocines et plus particulièrement dans les génomes de Vibrio Cholerae (bactérie à l'origine du choléra).

Pour cela, nous souhaitions utiliser une méthode basée sur des k-mers et des outils de classification supervisée (SVM et régression logistique). En effet, l'utilisation des k-mers est exploitée dans de nombreux domaines de la bio-informatique. Cependant, cet objectif initial s'est avéré trop ambitieux dans le cadre d'un mémoire. Nous nous expliquerons davantage dans la suite de ce travail.

Nous avons dès lors décidé d'orienter ce travail vers la discrimination plasmidique et chromosomique de séquences de Vibrio Cholerae et d'Escherichia Coli tout en maintenant l'utilisation de la méthode basée sur des k-mers et des outils de classification supervisée. En effet, la connaissance de la localisation d'une séquence est importante car cela permet de déterminer si un gène de la séquence fait partie d'un chromosome ou d'un plasmide. Comme nous le verrons dans la suite de ce mémoire, le séquençage actuel donne des séquences (contigs) dont l'origine plasmidique ou chromosomique est inconnue. Or, cette origine plasmidique ou chromosomique permet de connaitre le mode de transmission de ce gène qui diffère en fonction de sa localisation.

1.2 Biologie moléculaire

Nous allons à présent aborder les différents points théoriques nécessaires pour la bonne compréhension de ce travail au niveau biologique. Tout d'abord, nous commencerons par définir une bactérie ainsi que ses deux types. Nous expliquerons plus en profondeur les bactéries E. Coli et V. Cholerae, analysées dans ce mémoire. Ensuite, nous aborderons des notions concernant l'ADN, dont les différences entre un chromosome et un plasmide ainsi que quelques notions sur les deux types de transferts possibles. Nous continuerons par une explication du séquençage de l'ADN et plus particulièrement par la méthode de Sanger. Enfin, nous clôturerons ce chapitre par diverses explications théoriques sur les bactériocines.

1.2.1 Bactérie

Définition

« Micro-organisme formé d'une seule cellule, sans noyau, à structure très simple, considéré comme ni animal ni végétal. » (Le Robert, 2021). Il existe des bactéries dans tous les milieux, même dans les plus extrêmes. Plus de 10 000 espèces ont été recensées à ce jour. Les scientifiques estiment qu'il en existe plus de dix millions (3).

Types de bactéries

Il existe deux types de bactéries : les bactéries à Gram positif (+) et les bactéries à Gram négatif (-). Les bactéries à Gram positif possèdent une membrane tandis que les bactéries à Gram négatif en possèdent deux. On peut les distinguer grâce à une manipulation expérimentale (4).

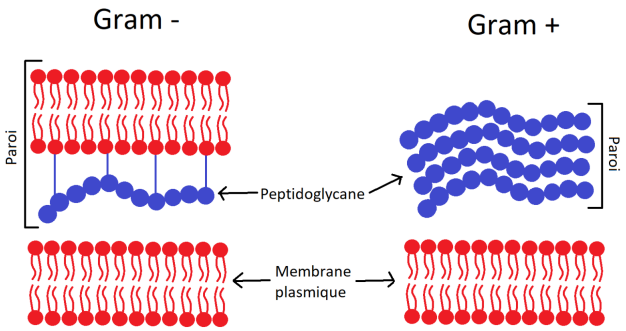


Figure 1.1: La différence entre Gram + et Gram - (5)

Escherichia Coli

E. Coli est une bactérie à Gram -. Il s'agit de la bactérie qui compose principalement la flore intestinale des vertébrés même si elle peut se situer également dans d'autres systèmes (ex : système urinaire). Certaines souches peuvent être pathogènes et entrainent des maladies. La bactérie a une forme de bâtonnet. Cette bactérie est la plus étudiée à l'heure actuelle (6).

Vibrio Cholerae

Le V. Cholerae est une bactérie à Gram -. Elle est à l'origine de la maladie du Cholera. Elle vit principalement dans des environnements aquatiques. Les humains tombent malades en s'abreuvant d'eau contaminée (7).

1.2.2 ADN (acide désoxyribonucléique)

L'ADN est une macromolécule biologique qui contient l'information génétique permettant à un être vivant de fonctionner. L'ADN contient 4 bases nucléotidiques : A (Adénine), T (Thymine), C (Cytosine) et G (Guanine). Les 4 bases forment des "mots" qui constituent les gènes. Il existe deux grandes familles de cellules (eucaryotes et procaryotes). La différence principale entre les eucaryotes et les procaryotes est respectivement la présence ou l'absence d'une membrane autour du noyau. De plus, les procaryotes sont des organismes unicellulaires. Les eucaryotes, quant à eux, peuvent être des organismes unicellulaires ou pluricellulaires (8).

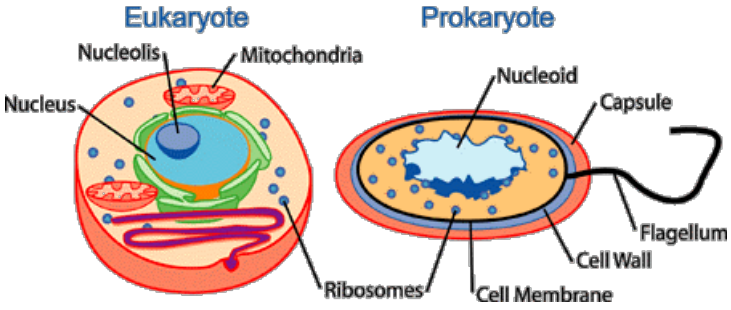


Figure 1.2: Les différences entre les cellules eucaryotes et procaryotes (9)

Procaryotes

L'information génétique dans les cellules procaryotes est présente sous forme de chromosomes circulaires et de plasmides. Le nombre de chromosomes présent est généralement de un. Cependant, il existe certaines espèces qui font exception comme le V. Cholerae qui contient 2 chromosomes. En plus de contenir un chromosome, les procaryotes possèdent

également des plasmides. Les plasmides sont également des molécules d'ADN. Les gènes responsables des avantages sélectifs (par exemple : gènes résistants aux antibiotiques) se trouvent principalement dans les plasmides. Ils sont plus petits que les chromosomes et sont capables de traverser la membrane externe des cellules. La taille moyenne d'un génome de procaryote se situe entre 600Kpb et 9.5Mpb (8).

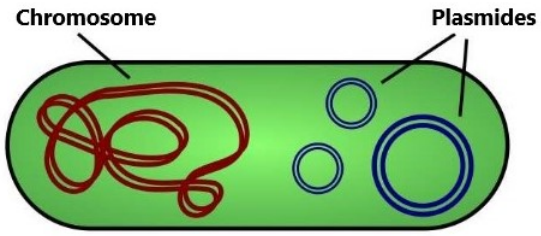


Figure 1.3: Les formes d'ADN dans les cellules procaryotes (10)

Eucaryotes

L'information génétique dans les cellules eucaryotes est exclusivement présente sous forme de chromosomes. Les cellules eucaryotes possèdent généralement plusieurs chromosomes. En effet, l'être humain, par exemple, est composé de cellules eucaryotes qui possèdent chacune 23 paires de chromosomes. Le génome complet d'un être humain comporte 3.2 milliards Pb (8).

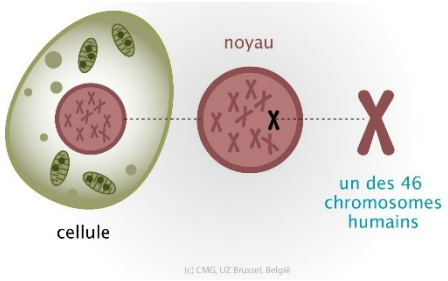


Figure 1.4: La forme d'ADN dans les cellules eucaryotes (11)

Transfert vertical et horizontal de l'information

Le transfert des gènes peut s'effectuer de deux manières différentes, verticalement ou horizontalement.

Le transfert vertical est le transfert d'informations de manière héréditaire. Il s'agit par exemple d'une bactérie qui reçoit des gènes de sa bactérie-mère. Ce type de transfert se fait généralement via les chromosomes (12).

Le transfert horizontal est le transfert d'informations d'un organisme à un autre sans en être le descendant. Ce transfert se fait par promiscuité. Les plasmides sont en grande partie responsables de ce type de transfert. En effet, ils ont la caractéristique de pouvoir sortir d'un organisme afin d'entrer dans un nouvel hôte (12).

Il est dès lors important de connaître la localisation d'un gène afin de comprendre son mode de transmission.

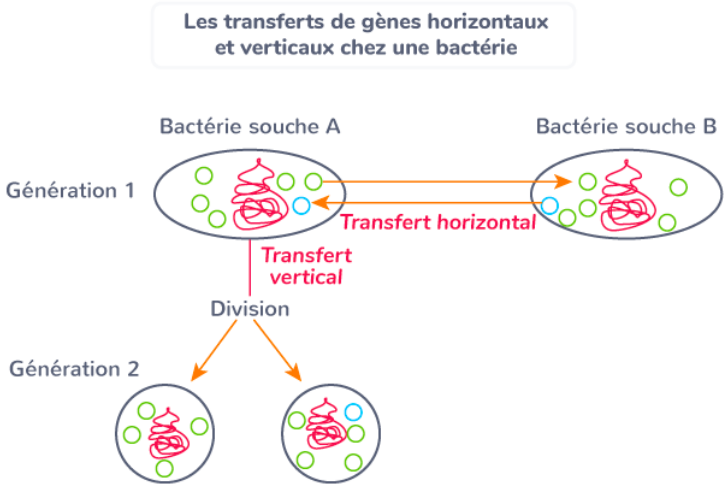


Figure 1.5: La différence entre le transfert vertical et horizontal (13)

1.2.3 Séquençage d'ADN

L'objectif du séquençage de l'ADN est de déterminer la succession des nucléotides lors de l'analyse d'une séquence d'ADN.

Il a fallu attendre la fin des années 1970 pour que deux méthodes de séquençage soient développées : la méthode de Sanger et la méthode de Maxam-Gilbert. Actuellement, la méthode de Maxam-Gilbert a été abandonnée au profit de celle de Sanger, qui est davantage performante (14).

La possibilité de séquencer des génomes d'êtres vivants y compris des génomes humains, via ces technologies, a permis de grandes avancées dans le monde biologique et médical. Il s'agit par exemple de la compréhension des maladies génétiques et des pistes quant à leur guérison potentielle ou encore de classifications phylogénétiques. Face à de tels enjeux de santé publique, ces méthodes ont sans cesse été améliorées, comme le montre la figure 1.6.

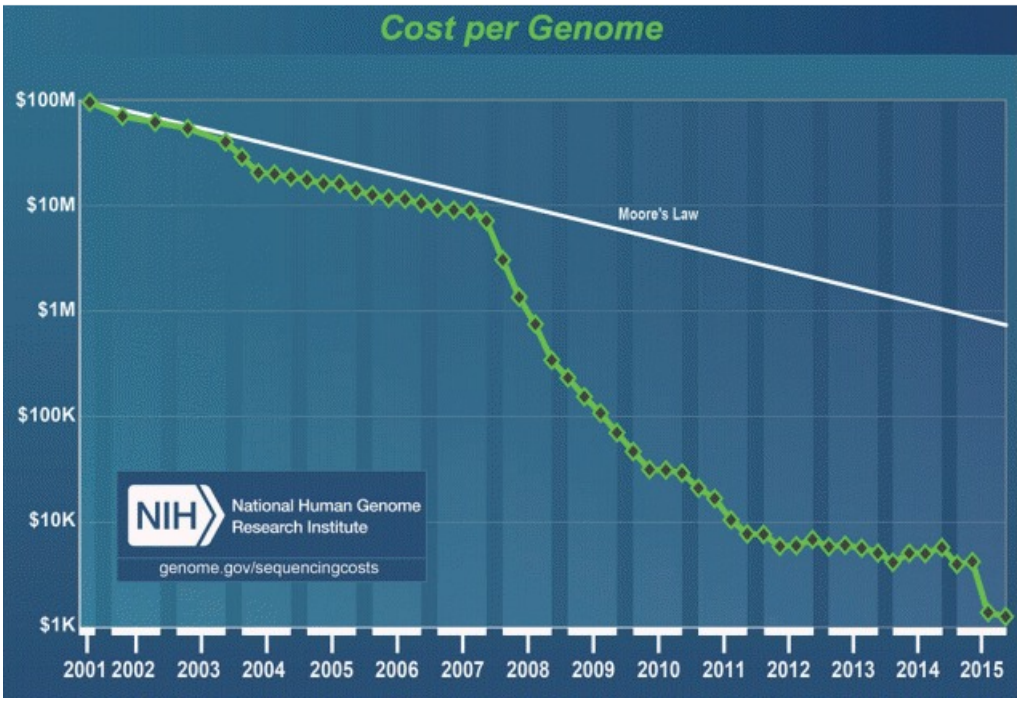


Figure 1.6: L'évolution du prix d'un génome humain (15)

Le coût de séquençage pour un génome humain n'a fait que décroitre au fil des années

pour atteindre en 2015 les 1000 dollars américains (15).

La méthode de Sanger

Tout d'abord, l'ADN va être multiplié grâce à la technique PCR (Polymerization Chain Reaction). Il sera ensuite dénaturé afin d'obtenir uniquement de simples brins d'ADN. Puis, une enzyme ADN polymérase est ajoutée afin de répliquer les brins d'ADN simples.

Dans son milieu naturel, cette enzyme ajoute des désoxyribonucléotides-triphosphates ou dNTP (ACGT) complémentaires. Lors de ces manipulations scientifiques, des didéoxyribonucléotides triphosphates ou ddNTP (ACGT modifiés) sont insérés dans le mélange. L'enzyme va ajouter de manière aléatoire des ddNTP. Contrairement aux dNTP, les ddNTP ne possèdent pas de groupe 3-hydroxyle qui permet la continuité de l'allongement de la chaine par l'ADN polymérase. L'allongement de la chaîne s'arrête dès lors après l'ajout d'un ddNTP. La venue aléatoire de ddNTP explique les tailles variables des séquences d'ADN. De plus, ces ddNTP sont munis de marqueurs fluorescents. Chaque nucléotide est représenté par une couleur spécifique. Ces marqueurs fluorescents permettent l'identification de la succession des nucléotides.
Cette méthode, développée par Sanger en 1977, ne permet le séquençage que de 800 nucléotides (14; 16).

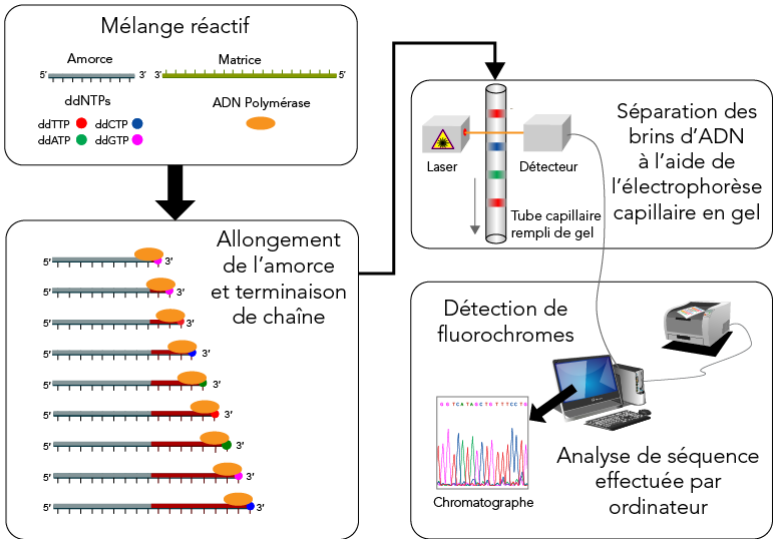


FIGURE 1.7: Les étapes de la méthode de Sanger (17)

Next Generation Sequencing (NGS)

Il existe deux générations de séquençage à haut débit : la deuxième et la troisième génération. Ces deux méthodes sont appelées NGS et utilisent la méthode de Sanger. Notons que la première génération, celle de Sanger, n'est pas à haut débit. La deuxième et la troisième génération ont des caractéristiques différentes. En effet, la deuxième génération va séquencer des reads (= petites portions d'ADN) de longueur variant de 150 à 300 bp dont le taux d'erreur est faible. La troisième génération va quant à elle séquencer des reads de longueur variant de 5000 à 20000 bp. Elle aura un taux d'erreurs plus important. Cette génération sera dès lors moins privilégiée que la deuxième (16).

Cependant, le séquençage NGS de troisième génération est de plus en plus utilisé pour reconstruire les génomes des bactéries. En effet, dans les génomes de bactéries, certaines régions génomiques assez longues (par exemple 5000 nucléotides pour le complexe 5s, 16s, 23s) peuvent être répétées plusieurs fois dans le génome. Si le séquençage de deuxième génération est utilisé, les séquences produites sont courtes (moins de 300 nucléotides) et ne permettent pas d'assembler précisément ces régions génomiques répétées et leur entourage. Cela produit des génomes assemblés (ou 'draft génomes') qui sont très fragmentés.

Or, la qualité de la reconstruction d'un 'draft génome' est généralement mesurée par le nombre de fragments (ou contigs) qui sont obtenus après l'étape d'assemblage. Dès lors, le séquençage NGS de troisième génération présente un grand avantage car il produit de très longues séquences. Le taux d'erreurs important est compensé par une grande profondeur de lecture. Chaque nucléotide est séquencé plus de 20 fois.

Grâce à la bio-informatique, les reads vont alors être assemblés afin de créer de plus grandes séquences, appelées des contigs. Cependant, il reste très compliqué de recréer le génome en entier à partir de ces contigs.

Dans le cadre de génomes bactériens, il est difficile de savoir si les contigs appartiennent à des chromosomes ou des plasmides.

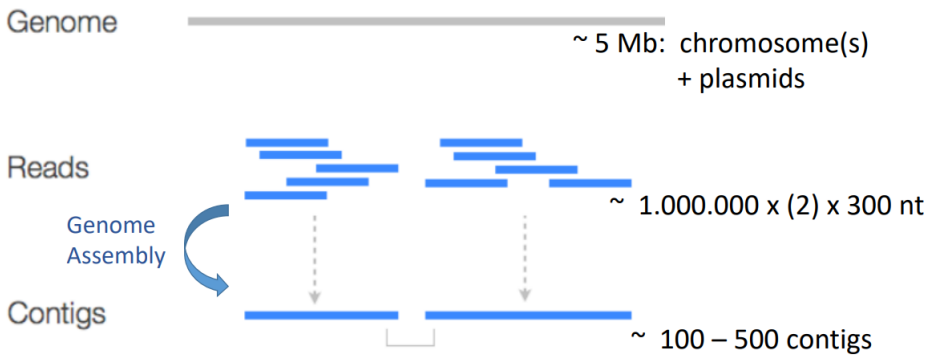


Figure 1.8: Les étapes du séquençage d'un génome par le NGS

1.2.4 Bactériocines

« Les bactériocines sont des protéines, ou complexes de protéines, avec une activité bactéricide contre des espèces proches de la souche productrice. Les bactériocines représentent une large classe de substances antagonistes qui varient considérablement du point de vue de leur poids moléculaire, de leurs propriétés biochimiques, de leur spectre d'action et de leur mode d'action »(18) (Klaenhammer, 1988). Les bactériocines sont synthétisées par un ribosome. Elles ont des propriétés antimicrobiennes qui sont utilisées dans diffé-

rentes applications comme la conservation alimentaire. Une application qui est en cours de développement est l'utilisation des bactériocines comme antibiotique. L'avantage principal d'une bactériocine est l'étroitesse de son spectre de destruction. La bactériocine ne pourrait dès lors tuer que la bactérie responsable de l'infection. Cela serait particulièrement avantageux contrairement aux antibiotiques dont le spectre de destruction est plus large. En effet, ils tuent une grande majorité de bactéries sans distinction, en ce compris les bactéries dont notre corps a besoin (2; 19).

À ce jour, 5 classes de bactériocines ont été recensées (voir fig : 1.9).

Class	Type
IA	Lantibiotics
IB	Lantibiotics
IIa	Non-lantibiotics
IIb	Two-peptide non-lantibiotics
IIc	
II microcin	
III	Large heat labile
IV	
V	Circular

Figure 1.9: Les 5 classes de bactériocines (20)

Les bactériocines sont classées en fonction : de leur nature, des modifications post-traductionnelles, de l'activité antibactérienne spécifique, de la formation d'oligomères, de la taille des protéines, de la présence de fragments de sucre, de la présence d'acides aminés chargés positivement et de leur mode d'action (20).

Localisation des gènes de bactériocines

Les bactériocines se trouvent dans les chromosomes et dans les plasmides. Le gène codant pour la bactériocine se trouve généralement proche du gène codant pour l'immunité contre celle-ci. En effet, si la bactérie ne possède pas l'immunité contre la bactériocine qu'elle produit, la bactérie s'autodétruirait dès la production de la bactériocine. Le gène codant pour la bactériocine se trouve également proche d'un gène codant pour la lyse. On appelle cela un complexe de gènes (19; 20).

Figure 1.10: Complexe de gène codant pour la Colicin E3 (19)

1.3 Outils bio-informatiques

Nous allons aborder les différents points théoriques nécessaires pour la bonne compréhension de ce travail au niveau de la bio-informatique. Tout d'abord, nous ferons une courte introduction sur ce domaine ainsi que sur le langage informatique utilisé dans le cadre de ce mémoire. Ensuite, nous expliquerons les k-mers ainsi que leur utilisation dans la discrimination plasmidique et chromosomique rencontrée dans la littérature. Puis, nous continuerons par une explication du programme Bagel4. Nous aborderons également la notion d'OTU et l'arbre phylogénétique. Enfin, nous clôturerons ce point par une courte description de la classification de séquences.

1.3.1 Introduction

La bio-informatique est un domaine multidisciplinaire qui a été défini dans les années 1970 par deux biologistes, Ben Hesper et Paulien Hogeweg. La bio-informatique regroupe des personnes venant de disciplines différentes : biologistes, informaticiens, physiciens,

médecins, mathématiciens... (21)

Ce domaine permet d'analyser et d'interpréter des informations biologiques en appliquant des concepts de statistique et d'informatique. À l'heure actuelle, ce domaine se concentre principalement au niveau moléculaire sur l'étude des séquences d'ADN et le repliement des protéines. La classification des séquences est une partie importante de la bio-informatique car cela permet de regrouper les différentes séquences d'ADN (22).

La bio-informatique utilise différents langages informatiques comme R, Python, Perl, ...

1.3.2 Langage R

Il existe différents langages qui auraient pu être utilisés dans le cadre de ce mémoire.

Cependant, le langage de programmation utilisé dans le mémoire est le langage R car celui-ci offre des packages plus complets pour la classification de séquences. Il a été développé à des fins statistiques. R a également été préféré aux autres car il existe une grande communauté active derrière lui, ce qui est un atout important. De plus, cette communauté a créé le projet « Bioconductor ». Celui-ci met à disposition énormément d'outils dans l'analyse et la compréhension génomique (23). Bioconductor fournit des centaines de packages. La plupart des packages utilisés dans ce mémoire seront issus de ce projet.

Un package est un ensemble de fonctions et de jeux de données. Il contient du code R et de la documentation sur les fonctions. Les packages sont créés par la communauté et peuvent être partagés avec d'autres personnes. Cela permet d'élargir les possibilités dans R. Ils permettent également de structurer le travail de chacun (24).

1.3.3 K-mers

À présent, nous allons expliquer l'utilisation des k-mers en bio-informatique.

Un k-mer est une sous-chaîne de caractères de longueur k. Dans le cadre de ce mémoire, les k-mers sont les sous-séquences d'ADN de longueur k.

Pour un k défini, le nombre total de combinaisons de nucléotides différentes est donné par la formule 4^k (4 pour les 4 nucléotides ACGT).

Par exemple, k=2 va donner 16 k-mers différents :

Figure 1.11: Tous les k-mers pour un k=2

Dans ce mémoire, des analyses vont être réalisées afin de compter l'occurrence de chaque k-mer dans une séquence donnée. Ces analyses vont donner un vecteur de longueur 4^k avec le nombre d'occurrences de chaque k-mer. Chaque k-mer sera dès lors une variable.

La difficulté majeure de cette technique est de trouver le k optimal. La taille des k-mers est cruciale car, si le k est trop petit, il y aura une perte d'informations de la séquence. En effet, il sera compliqué pour les petits k-mers de découvrir les zones de répétition dans l'ADN. De plus, avec un petit k, la possibilité de trouver une séquence telle que AGTCGTAGATGCTG est plus faible que celle de trouver une séquence plus petite telle que AGTC. Par contre, prendre un k trop grand amènera énormément de variables, ce qui risque de rendre plus complexe la classification (25; 26).

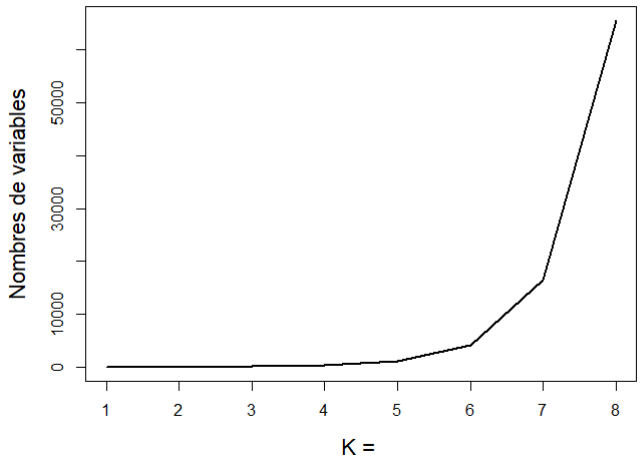


FIGURE 1.12: Le nombre de variables généré en fonction d'un k

Cette méthode est utilisée dans différentes branches de la bio-informatique : classification de séquences, assemblage de génomes, alignement de séquences, regroupement de séquences, correction des erreurs de lecture de séquençage, estimation de la taille du génome, identification des répétitions, ...

Cette méthode est également utilisée dans la distinction entre des fragments d'ADN venant d'un plasmide ou d'un chromosome.

Littérature : plasmides et chromosomes

Dans la littérature, il existe 2 articles de référence dans l'utilisation des k-mers pour la discrimination chromosomique vs plasmidique.

Tout d'abord, Zhou & Xu (2010) ont élaboré un programme informatique appelé « Cbar ». Après avoir effectué différentes analyses afin de déterminer le k optimal, il s'est avéré que le k=5 était le plus performant. Leur méthode consiste à travailler avec des génomes entiers d'espèces différentes. Cbar arrive à des performances allant jusqu'à 91.8% de précision en utilisant un outil de classification SMO (sequential minimal optimization). Cependant, comme expliqué dans 1.2.3 « Séquençage », le séquençage actuel fournit des contigs dont

l'origine chromosomique ou plasmidique est inconnue. Or, Cbar travaille sur des génomes entiers. Il n'est dès lors pas capable de résoudre correctement le problème de discrimination entre des contigs venant de chromosomes ou de plasmides.(27)

C'est pourquoi, Krawczyk, Lipinski & Dziembowski (2018) ont développé un autre programme informatique appelé PlasFlow. Ce programme utilise un neural network approach comme outil de classification. Les auteurs ont effectué différentes analyses afin de déterminer le k optimal. Il s'est avéré que les k de 5 à 7 étaient les plus performants. Les auteurs ont téléchargé des génomes entiers d'espèces différentes. Afin de pallier le problème de la discrimination entre des contigs venant de chromosomes ou de plasmides, ils ont fragmenté aléatoirement ces génomes en des séquences de 10000 nucléotides. Ils ont obtenu un pourcentage de réussite de 96%. Cependant, bien qu'il s'agisse d'un programme informatique plus performant que Cbar, PlasFlow ne peut travailler qu'avec des plasmides de plus de 10000 nucléotides. Il passe donc à côté de tous ceux dont le nombre de nucléotides est inférieur (28).

Un troisième programme informatique, issu de ces deux programmes de référence, a été créé : mlplasmid. Celui-ci utilise un k=5. Ce programme utilise le SVM comme outil de classification. Ce programme se distingue des deux premiers sur deux points. D'une part, il effectue le même traitement des séquences des génomes entiers que PlasFlow mais ne traite que des séquences que de 1000 nucléotides. D'autre part, mlplasmid effectue ses analyses espèce par espèce et non plus toutes les espèces ensemble comme le font les deux premiers programmes. Les espèces étudiées par mlplasmid sont Enterococcus faecium, Klebsiella pneumoniae et Escherichia coli (29).

Dans le cadre de ce mémoire, nous allons réaliser nos analyses sur le Vibrio Cholerae et sur Escherichia Coli en prenant exemple sur mlplasmid en utilisant le SVM et la régression logistique comme outils de classification.

1.3.4 Bagel4

À l'heure actuelle, peu de techniques existent pour la découverte de nouvelles bactériocines car le nombre de bactériocines connu n'est pas assez grand. Or, il est plus aisé de faire de plus amples découvertes dans un domaine quand celui-ci est déjà exploité. De plus, les bactériocines sont toutes très différentes les unes des autres. Les méthodes actuelles utilisées en bio-informatique sont soit HMM (Hidden Markov Model), soit Blast. Certains chercheurs ont développé un serveur web qui regroupe les méthodes les plus utilisées dans la découverte des bactériocines. Il s'agit de Bagel4.

Bagel est un serveur web créé en 2006. Il en est à ce jour à sa quatrième version. « Bagel4 permet aux utilisateurs d'identifier et de visualiser des groupes de gènes dans l'ADN procaryote impliqués dans la production de peptides synthétisés par voie ribosomale et modifiés par voie post-traductionnelle (RiPPs) et de bactériocines (non modifiées) » (20). Outre dans la recherche de traitements alternatifs aux antibiotiques, les bactériocines jouent également un rôle important dans la conservation des aliments, l'écologie microbienne et le biocontrôle des plantes (20).

Bagel4 présente deux avantages principaux. Tout d'abord, ce serveur web contient une grande base de données de bactériocines. Ensuite, comme expliqué dans le sous-point 1.2.4 « Bactériocines », les gènes de bactériocines se trouvent dans un complexe de gènes. Ce complexe de gènes comprend également des protéines nécessaires à la transformation, la modification, le transport, la régulation et/ou l'immunité de ces bactériocines. Contrairement aux autres méthodes qui ne tiennent compte que des séquences de bactériocines, Bagel4, en plus de tenir compte des séquences de bactériocines, tient également compte de toutes les séquences de protéines de ce complexe de gènes (20).

Concrètement, l'utilisation de Bagel4 est très simple. Il suffit d'uploader un fichier fasta comprenant une séquence d'ADN sur le serveur web. Ensuite, Bagel4 va analyser cette séquence à la recherche de régions potentielles comprenant une bactériocine. Une fois

l'analyse réalisée, Bagel4 réalise un tableau reprenant les différentes régions potentielles (exemple fig : 1.13) (30).

AOI	start	end	Class	Fasta header
CP0076531.1.AOI_01	1516910	1536910	Microcin	CP007653.1
CP0076551.2.AOI_01	19013	45613	Sactipeptides	CP007655.1

Figure 1.13: Tableau sorti par Bagel4 sur le génome CP007653.1

Légende :

AOI : Area of interest

Start : Numéro du nucléotide de départ de la région potentielle

End : Numéro du nucléotide de fin de la région potentielle

Class : La classe ou sous-classe de bactériocine potentielle

Fasta header : Le nom du fichier fasta

Bagel4 propose d'analyser ces régions plus profondément. La figure 1.14 représente une région potentielle comprenant une bactériocine et ses ORF avoisinants.

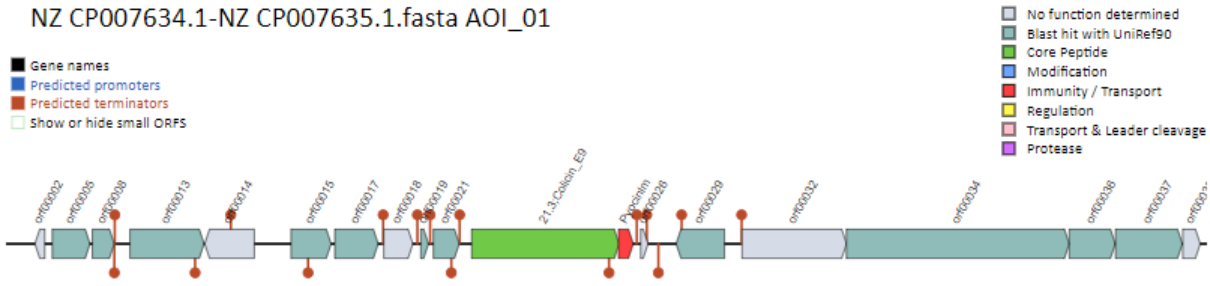


Figure 1.14: Région potentielle de colicin E9 sorti par Bagel4

Enfin, il est possible d'extraire facilement ces régions via un langage informatique.

1.3.5 OTU

Une OTU ou Unité Taxonomique Opérationnelle est un ensemble de séquences d'ADN qui ont une suite de nucléotides similaires. Il s'agit de groupes de séquences dont le taux d'identité est supérieur à un certain seuil préalablement défini. Dans la littérature, ce seuil est fixé à 97%. Il en sera de même dans ce mémoire. L'OTU permet de regrouper des séquences similaires. Il est ainsi possible d'extraire une séquence représentative par groupe. Ceci permet de diminuer drastiquement le nombre de séquences à analyser lors de la réalisation d'un arbre phylogénétique (31; 32).

Pour réaliser un OTU, nous avons utilisé l'outil informatique Usearch qui est disponible sur le site : https ://drive5.com/usearch/ (33).

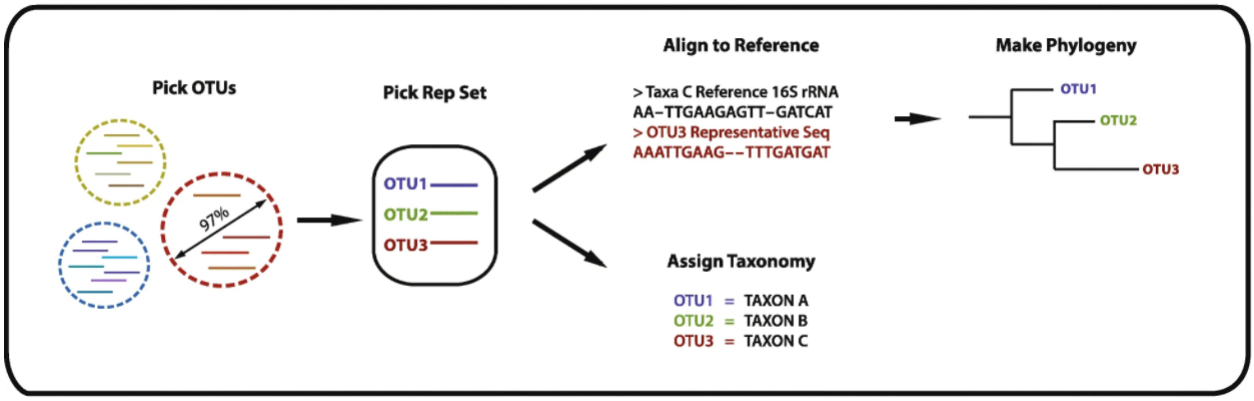


Figure 1.15: Explication de l'utilisation d'OTU(34)

1.3.6 Arbre phylogénétique

Un arbre phylogénétique représente les relations entre des espèces biologiques qui ont évolué à partir d'un ancêtre commun. Un arbre phylogénétique peut être représenté de différentes manières (dendrogramme, diagramme, . . .). Dans ce mémoire, nous avons privilégié la forme dendrogramme pour son aspect visuel. Il comprend différentes ramifications qui représentent les changements nucléotidiques dans les gènes. L'arbre phylogénétique permet également de suivre l'ordre dans lequel ces changements se sont produits. Il est

composé de 3 éléments : les nœuds, les feuilles et les branches (35).

Chaque nœud représente un ancêtre commun. Chaque feuille de l'arbre représente un taxon. Dans ce mémoire, chaque taxon est une séquence spécifique d'une souche d'une bactérie étudiée. Nous nous intéressons particulièrement à la longueur des branches entre les nœuds. Cela permet en effet de déterminer s'il y a d'importantes différences ou non entre les séquences : au plus la longueur d'une branche est grande, au plus la différence entre l'ancêtre commun et les séquences est grande (35).

Pour créer un arbre phylogénétique, il faut créer une matrice de distance entre les séquences étudiées. Cette matrice de distance est une matrice qui contient le score entre les différentes séquences : au plus le score est grand, au plus grande est la différence entre les séquences et inversement. Il existe différentes méthodes pour effectuer la matrice. Dans ce mémoire, la distance de Hamming est celle qui a été choisie car elle est la plus simple. L'arbre phylogénétique peut alors être créé. Il existe également différentes méthodes pour réaliser l'arbre phylogénétique. La méthode UPGMA est la méthode choisie pour ce mémoire car elle donne un arbre phylogénétique en dendrogramme (35).

1.3.7 Classification de séquences

L'objectif de la classification des séquences d'ADN est de comparer des séquences d'ADN entre elles afin de détecter les différences et les similarités entre les séquences. Cette notion est particulièrement importante dans de nombreux domaines biologiques. Cette classification est d'habitude réalisée sur un alignement des séquences d'ADN. Cependant, dans ce mémoire, nous allons nous focaliser sur les k-mers pour discriminer les groupes. Il sera dès lors possible de classer des séquences inconnues. Cela permettra de détecter des bactériocines dans des génomes ou de discriminer des séquences de chromosomes ou de plasmides (22).

1.4 Outils de classification

Nous allons maintenant aborder les différents points théoriques nécessaires pour la bonne compréhension de ce travail au niveau mathématique. Tout d'abord, nous allons commencer par une brève explication de l'analyse en composantes principales (ACP). Ensuite, nous expliquerons les deux outils de classification supervisée utilisés lors de ce mémoire. Il s'agit du SVM et de la régression logistique (dans sa version classique ainsi qu'avec des pénalités sur les paramètres dans les versions de Ridge et de Lasso). Nous terminerons par une explication du phénomène d'overfitting, phénomène important dans la classification supervisée.

1.4.1 Analyse en composantes principales (ACP)

« L'ACP est utilisée pour réduire la dimensionnalité d'un ensemble de données, qui consiste en un grand nombre d'attributs interdépendants, tout en conservant autant que possible la variation présente dans l'ensemble de données original » (36). Il s'agit d'un outil statistique utilisé dans les domaines de recherche utilisant un grand nombre de variables. L'ACP permet de réduire ce grand nombre de variables et de limiter la redondance de l'information des variables. En effet, l'ACP utilise une transformation orthogonale, ce qui aura pour effet de transformer des variables corrélées en de nouvelles variables non corrélées. Ces nouvelles variables sont appelées composantes principales. Chaque composante principale rend compte, de manière décroissante, d'un maximum de variabilité. Les premières composantes principales auront dès lors la plus grande variabilité parmi le jeu de données. De plus, toutes les composantes principales sont orthogonales entre elles (36).

Dans ce mémoire, seules les deux premières composantes principales de l'ACP seront analysées afin d'observer si une tendance se dégage entre les séquences de chromosomes et de plasmides.

1.4.2 SVM

Nous avons utilisé deux outils de classification. Le premier outil de classification utilisé est le Support Vector Machine. Les SVM sont utilisés pour la classification binaire et multiclasse et pour faire de la régression.

Dans le cadre de ce mémoire, le SVM binaire a été utilisé. En effet, les séquences d'ADN proviennent soit du chromosome soit du plasmide.

La classification peut être modélisée dans un espace. Il s'agit de l'espace de représentation. Chaque observation du jeu de données sera dès lors un point dans l'espace. Les coordonnées des observations sont définies par leurs caractéristiques. Dans ce mémoire, les caractéristiques sont l'occurrence des différents k-mers.

L'objectif du SVM est de chercher un hyperplan séparant les points de données associés aux différentes classes. Comme il est possible d'observer sur la figure 1.16, il peut exister une infinité de frontières de décision séparées pour des données. Le SVM va alors tenter de fournir l'hyperplan qui a la marge maximale (37).

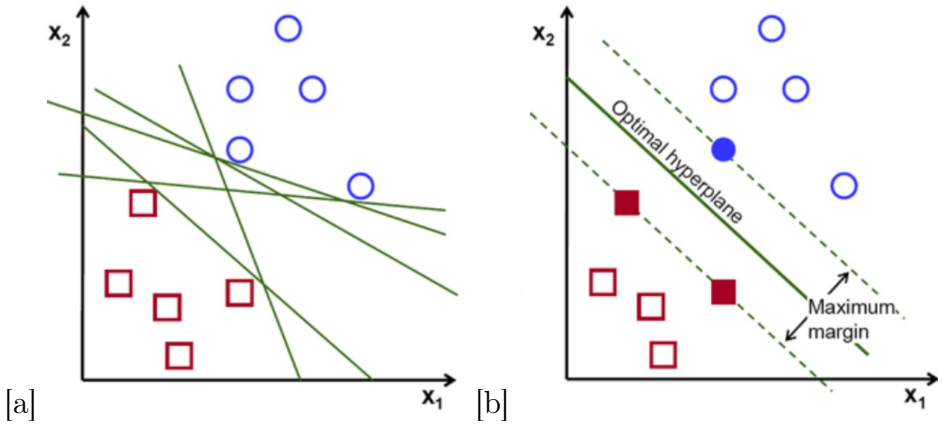


Figure 1.16: a) Exemple d'hyperplan possible b) l'hyperplan prit par le svm (38)

L'hyperplan séparateur du SVM $g(x)$ est défini par une fonction linéaire donnée par : $g(x) = w^T X + w_0$, où le vecteur w et la constante w0 sont les paramètres du modèle. La marge géométrique est définie comme $\frac{1}{||W||}$. Plus w est petit, plus la marge est grande.

Le SVM va dès lors chercher le plus petit w afin d'avoir la marge maximale. Pour cela, il faut minimiser $\frac{1}{2||W||^2}$. Cette minimisation est une tâche mathématique facile car il s'agit d'une forme quadratique (39).

Lorsque les données ne sont pas linéairement séparables, le SVM est capable de modifier l'espace de données afin de le rendre linéaire. Cet espace est de plus grande dimension et est appelé espace des caractéristiques. Pour obtenir ce nouvel espace, le SVM applique une projection des données de l'espace de représentation sur l'espace caractéristique. Plus la dimension de l'espace des caractéristiques est grande, plus la probabilité de trouver un hyperplan séparateur est grande. La modification de l'espace non linéaire en un espace linéaire est réalisée grâce à une fonction noyau (37).

Il existe différentes fonctions noyau : polynomiale, gaussienne (RBF), sigmoïdale et laplacienne. Dans ce mémoire, la fonction noyau utilisée est la gaussienne. Cette fonction est également appelée RBF (Radial basis function). Elle est caractérisée par la fonction $K(x, y) = e^{(-\gamma||x-y||^2)}$. Le gamma est le paramètre du noyau. Le gamma doit être défini au préalable (40).

Cependant, il n'est pas toujours possible de trouver un hyperplan qui sépare exactement les données. Il va dès lors falloir modifier l'équation d'optimisation de sorte que les données ne se trouvant pas du bon côté de l'hyperplan soient tout de même prises en compte afin de trouver un hyperplan. Le nombre de ces données doit être le plus petit possible. La distance entre ces données et l'hyperplan est mesurée par des variables souples. La modification de l'équation d'optimisation consiste donc en la modification de la marge initiale (« dure ») en marge « souple ». Pour la marge « souple », l'équation d'optimisation devient alors (37) :

$$min_{w,\xi} \frac{1}{2}||w||^2 + C \sum_{i=1}^{n} \xi_i$$

où C est le coût de misclassification et est responsable de l'épaisseur de la marge souple et où les ξ_i sont les variables ressorts (slack variables). Ces variables permettent un relâ-

chement des contraintes sur les vecteurs d'apprentissage.

Les paramètres à ajuster avec le SVM sont : la fonction noyau, le coût de misclassification et le paramètre de la fonction du noyau (gamma avec le RBF). Dans le cadre de ce mémoire, nous allons nous focaliser sur le coût de misclassification et le gamma afin de trouver les paramètres optimaux tout en prenant comme fonction noyau le RBF.

1.4.3 Régression logistique

Le deuxième outil de classification utilisé dans ce mémoire est la méthode utilisant la régression logistique.

La régression logistique permet d'associer à un vecteur de variables aléatoires, une variable binomiale Y (soit chromosome = 1 soit plasmide = 0). Il s'agit en effet d'un modèle largement utilisé lorsque la réponse est binaire. L'objectif de ce modèle est de modéliser π qui est la probabilité a posteriori que la réponse soit positive (égale à 1) sachant les prédicteurs X_j . Si $P(Y = 1|X_1, .., X_p) = \pi$ et $P(Y = 0|X_1, .., X_p) = 1 - \pi$, on déduit alors :

$$\text{si } \tfrac{\pi}{1-\pi} > 1 \text{ alors Y} = 1.$$

Cette quantité est appelée le rapport de chances. Le modèle de la régression logistique fait l'hypothèse que la transformation logit de π peut être modélisée par une combinaison linéaire des prédicteurs :

$$logit(\pi) = ln(\tfrac{\pi}{1-\pi}) = \beta_0 + \textstyle\sum_{j=1}^{P} \beta_j X^j$$

La méthode du maximum de vraisemblance permet de donner l'estimation du vecteur $\beta \in R^{p+1}$ des coefficients. Etant donné que le Y n'est pas directement modélisé, la vraisemblance correspond à une vraisemblance conditionnelle (41).
Si $\pi_i = P(Y_i = 1|X_i)$, la probabilité que la réponse soit positive pour l'individu i sachant X_i, $P(Y_i = y_i|X_i) = \pi_i^{yi}(1 - \pi_i)^{1-y_i}$, la vraisemblance s'écrit alors :

$$L(\beta) = \prod_{i=1}^{n} \pi_i^{yi}(1-\pi_i)^{1-y_i}$$

Cependant, nous avions dans notre échantillon 1024 variables. Or, l'utilisation du modèle de régression logistique est à exclure lorsque le nombre de variables est trop excessif. En effet, les systèmes d'équations qui permettent d'estimer les paramètres du modèle sont surdéterminés et présentent dès lors une infinité de solutions (41; 42).

Pour pallier ce problème, il existe des méthodes de régressions pénalisées.

Les méthodes de régressions pénalisées

Ces méthodes vont introduire une pénalité sur le vecteur des coefficients, c'est-à-dire qu'un certain nombre de coefficients vont devenir nuls (Lasso) ou proches de zéro (Ridge). Cela va impliquer la diminution de la variance des estimateurs (41; 42).

Cela permet de pallier le problème des paramètres surdéterminés ou de réduire la dimension d'un problème en tenant compte d'un nombre réduit de variables (41).

Les pénalités Ridge et Lasso portent sur la log-vraisemblance. Cela demande la résolution du problème d'optimisation suivant :

$$\arg \max_{\beta} \{LL(\beta) - g_\lambda(\beta)\}$$

LL est la log-vraisemblance du modèle, et $g_\lambda(\beta)$ est soit la pénalité de Ridge($g_\lambda(\beta) = \lambda||\beta||_2^2$) soit la pénalité de Lasso($g_\lambda(\beta) = \lambda||\beta||_1$). Cependant, la résolution de ce problème n'est pas possible. En pratique, ce problème peut être approché par différentes méthodes d'optimisation (41).

Lorsque le lambda = 0, il n'y aura pas de pénalité sur les coefficients, tandis que si le lambda est très grand, la pénalité sera grande. Il ne restera plus que la constante β_0 (41).

La régression Lasso est quant à elle privilégiée par rapport à la régression de Ridge lorsque la dimension des données est importante. La pénalité Lasso en norme l_1 implique qu'un

certain nombre de variables vont être nulles, impliquant ainsi une réduction du nombre des variables. Cela permet de réduire la complexité du modèle et donc de diminuer le risque d'overfitting (41; 42).

1.4.4 Overfitting

Le phénomène d'overfitting (sur-apprentissage) (voir figure 1.17 à droite) survient lorsqu'un modèle correspond trop précisément à l'ensemble des données. De par sa complexité trop élevée, ce modèle prédira moins bien de nouvelles données qu'un modèle moins complexe.

Le phénomène d'underfitting (sous-apprentissage) (voir figure 1.17 à gauche) survient, par opposition, lorsqu'un modèle est trop simple. Ce modèle ne prédira pas précisément de nouvelles données. Ce modèle est trop simplifié.

L'objectif va être de trouver un compromis entre ces deux phénomènes afin d'obtenir un modèle qui corresponde au mieux à la réalité (voir figure 1.17 au centre).

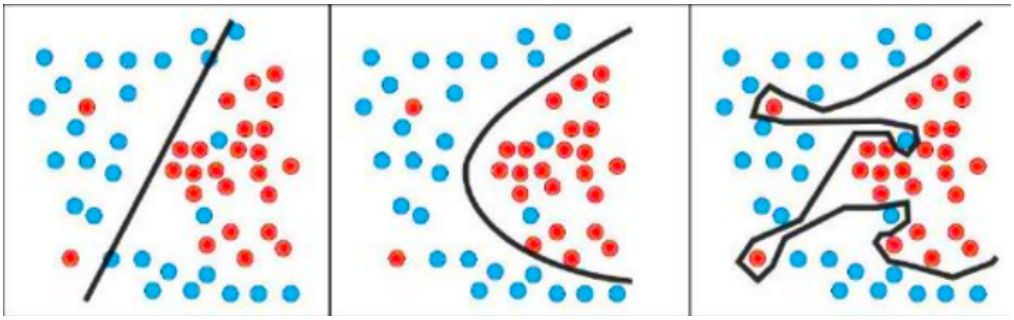


Figure 1.17: Exemple d'un underfitting, d'un goodfitting et d'un overfitting (43)

Comme on peut l'observer sur la figure 1.18, au plus un modèle devient trop complexe, au plus les données qui ont servi à entrainer ce modèle seront bien prédites par ce modèle (Courbe training error). Cependant, il y aura plus d'erreurs pour un jeu de données qui n'a pas servi à entrainer ce modèle trop complexe qu'avec un modèle moins complexe (Courbe test error). L'objectif sera de trouver la meilleure complexité afin que le taux d'erreurs sur le nouveau jeu de données soit le plus bas possible (Best Fit) (44).

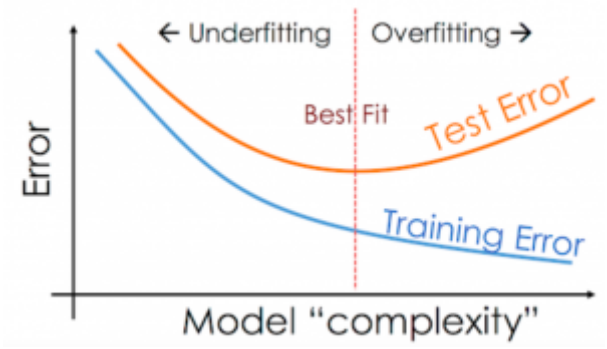


Figure 1.18: Courbe caractéristique : le taux d'erreur sur la complexité du modèle (45)

Dans le cadre de ce mémoire, nous allons effectuer ce type de graphique. La courbe test error sera calculée en utilisant la méthode de cross-validation et la courbe training error sera calculée en utilisant le même jeu de données qui a servi à entrainer le modèle.

Cross-validation

Cette méthode permet de se rapprocher de la performance réelle du classificateur. Concrètement, cette méthode va séparer des observations d'entrées en k groupes distincts. Elle va ensuite créer le classificateur sur base de k-1 groupes et utiliser le dernier groupe comme test. Cette expérience va être réalisée k fois, de manière à ce que chaque groupe puisse être le groupe test. Ensuite, la moyenne des performances sur les différents tests sera calculée. Cette moyenne se rapproche de la performance réelle du classificateur (46).

Figure 1.19: Exemple d'une cross-validation (47)

Chapitre 2

Méthodologie

L'objectif de ce mémoire est de tenter de découvrir de nouvelles bactériocines grâce à l'utilisation de la méthode basée sur des k-mers et d'outils de classification supervisée. Dans ce chapitre, nous allons dès lors aborder les différentes méthodologies utilisées. Nous allons commencer par expliquer l'origine des données que nous avons utilisées. Puis, vu la complexité du domaine de recherche de nouvelles bactériocines, nous avons commencé par nous baser sur ce qui existait déjà dans la littérature scientifique. En effet, nous expliquerons la manière dont Bagel4 et la méthode des k-mers ont été utilisées. Enfin, nous aborderons deux classificateurs utilisés avec les k-mers : SVM et régression logistique.

2.1 Sources des données

Dans le cadre de ce projet, nous allons travailler sur base de génomes de bactéries de V. Cholerae et d'E. Coli. Pour cela, il est nécessaire de trouver des bases de données dans lesquelles se trouvent des numéros d'accession de génomes complets pour les plasmides et les chromosomes du V. Cholerae et E. Coli.

Une fois que les numéros d'accession sont trouvés, les différents génomes doivent être téléchargés. Pour cela, la base de données de Genbank avec le Package «ape » permet de

télécharger les génomes sur R.

Afin de recenser des numéros d'accession, deux bases de données ont été utilisées dans ce mémoire.

La première est la base de données GenBank (48). Cette base de données est la plus fournie au monde. Elle est composée de 61 millions de séquences d'ADN. Elle permet également de recenser un grand nombre de numéros d'accession. Cette base de données permet d'obtenir facilement un grand nombre de numéros d'accession pour des séquences précises (chromosome, plasmides). Cependant, les numéros d'accession sur des génomes entiers ne permettent pas de distinguer les chromosomes des plasmides dans le génome. Cela implique que, si nous trouvons des résultats, il sera presque impossible de pouvoir localiser précisément le lieu de la séquence. Cette base de données fournira une grande quantité de numéros d'accession de chromosomes et plasmide séparément.

La deuxième base de données provient du package R « Genomes ». Cette base de données n'est composée que des numéros d'accession de différentes bactéries. L'avantage de cette base de données est qu'elle fournit les numéros d'accession d'un génome entier en séparant les numéros d'accession des différents chromosomes et plasmides.

Il a été possible de télécharger des génomes complets avec Genbank :

	E. Coli	V. Cholerae
Génomes (NCBI)	100	93
Chromosome 1(Genomes)	/	139
Chromosome 2 (Genomes)	/	145
Plasmides (NCBI)	8000	40

Table 2.1: Nombre de génomes téléchargés

Nous avons observé de la redondance dans les numéros d'accession entre les bases de données.

2.2 Bagel4

L'objectif initial de ce mémoire était de découvrir de nouvelles séquences potentielles de bactériocines dans les génomes du Vibrio Cholerae.

L'utilisation de Bagel4 est simple. Etant donné qu'il s'agit d'un serveur web, il suffit d'aller sur le lien suivant :`http://bagel4.molgenrug.nl/index.php` . Les génomes téléchargés précédemment sont alors uploadés pour être analysés par Bagel4.

Une fois que Bagel4 a terminé de parcourir tous les génomes et de sortir toutes les régions potentielles, des analyses plus détaillées ont été réalisées.

Bagel4 indique les classes des bactériocines des régions potentielles. La première chose qui a été effectuée est l'analyse des différentes régions une par une. Cette analyse a permis de déterminer si les régions de même classe de bactériocines sont similaires entre elles ou non. Une fois cela fait, les régions potentielles dans les différents génomes ont alors été extraites.

Ensuite, il a été possible de télécharger sur Genbank d'autres bactériocines de classes similaires afin de comparer les séquences extraites et les bactériocines venant de Genbank.

Afin d'alléger le nombre de séquences, la méthode OTU a été utilisée. Elle a permis de supprimer toutes les séquences trop proches phylogénétiquement et de n'en retenir qu'une seule.

Enfin, un arbre phylogénétique a été réalisé afin d'observer la proximité phylogénétique ou non de ces séquences. L'arbre a été réalisé sur les nucléotides ainsi que sur les acides aminés.

Comme nous allons le voir plus précisément dans la suite de ce mémoire, les analyses réalisées sur une région potentielle par Bagel 4 ont donné des résultats peu concluants. C'est pourquoi nous avons décidé de modifier la tournure de ce mémoire.

2.3 K-mers

La méthode basée sur des k-mers va être utilisée afin de générer un tableau qui sera alors utilisé par des classificateurs.

Comme mentionné dans la littérature sur l'utilisation des k-mers, l'objectif va être de prendre des séquences de 1000 nucléotides. Cela permettra de prendre en compte tous les plasmides supérieurs à 1000 nucléotides. Afin de minimiser le taux d'erreur, chaque espèce a été étudiée séparément. Les séquences téléchargées précédemment seront choisies aléatoirement. Une fois la séquence choisie, un échantillon de 1000 nucléotides de la séquence sera alors extrait. Ce morceau d'ADN fera office de contig. Afin d'obtenir un nombre d'observations convenable, cette action sera répétée 1000 fois sur les chromosomes et 1000 fois sur les plasmides. Il y aura ainsi en tout 2000 observations.

Une fois que les contigs ont été générés, le tableau des k-mers est alors créé :

	AAAAA	AAAAC	AAAAG	AAAAT	AAACA	AAACC	AAACG	AAACT	AAAGA	AAAGC	
1	3	2	1	1	1	0	0	3	0	1	Chromosome
2	2	1	2	1	0	2	2	2	0	2	
3	0	1	0	1	1	1	1	0	0	1	
4	0	1	0	0	0	3	1	1	0	1	
5	5	3	4	1	5	0	1	1	2	1	
6	5	0	4	3	0	0	2	1	3	3	
7	2	7	3	3	2	1	3	2	4	2	
8	6	2	3	2	2	2	0	0	2	2	Plasmide
9	3	1	1	2	2	1	1	1	0	0	
10	3	5	0	1	0	4	4	3	0	0	
11	2	0	2	3	3	0	1	0	3	1	
12	8	3	6	2	2	1	0	1	3	4	

Figure 2.1: Exemples de tableau obtenu avec l'utilisation des k-mers

Une colonne a été ajoutée à la droite du tableau afin de spécifier si les séquences viennent d'un chromosome ou d'un plasmide. Une fois le tableau terminé, différents classificateurs ont été testés afin de comparer leur efficacité. Les performances des classificateurs ont été réalisées via deux méthodes. La première méthode a été réalisée en Cross

validation avec un nombre de « KFolds » égal à 10. Cette méthode est représentée par la ligne rouge présente dans tous les graphes. La deuxième méthode a été réalisée sur le set d'entrainement et est représentée par la ligne bleue présente dans tous les graphes. La comparaison des classificateurs se fait sur base de leur pourcentage d'erreurs.

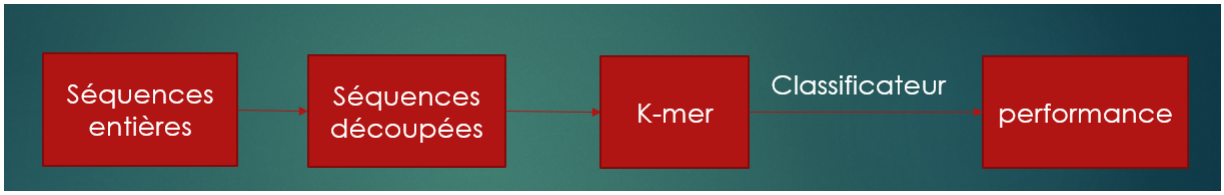


Figure 2.2: Méthodologie effectuée

2.4 ACP

Une ACP a été réalisée sur chaque espèce avec un k=5 grâce au package R «Factominer». Ces ACP ont été effectuées afin d'observer si une tendance se dégage entre les séquences de chromosomes et de plasmides et si cette méthode serait potentiellement intéressante dans la réduction des variables.

2.5 SVM

Le premier classificateur testé est le SVM (Support Vector Machine). Les fonctions utilisées se trouvent dans le package R « e1071». Le choix du noyau est RBF. De manière générale, les paramètres choisis pour le classificateur sont le coût de misclassification(C) de 1 et le paramètre de la fonction noyau : gamma de $\frac{1}{nombredevariables}$. Seuls ces paramètres seront modifiés lors de la dernière analyse. L'objectif est de trouver les paramètres qui vont donner la meilleure précision du classificateur afin de gérer au mieux le phénomène d'overfitting. Pour cela, différentes analyses ont été réalisées :

— La première analyse s'est faite sur un k = 5. Il y a donc 1024 variables. Ce nombre de variables est important pour un classificateur. L'objectif est de diminuer les variables en ne prenant que les plus discriminantes selon un t-test sur le set d'en-

trainement. Les variables ont été analysées par tranches de 50, en commençant à 10. Cette méthode permet de diminuer considérablement le temps de calculs réalisés par l'ordinateur.

— Le SVM est capable de gérer un grand nombre de variables en entrée avant d'avoir de l'overfitting. L'objectif de cette deuxième analyse est de changer le k afin de faire varier le nombre de variables en entrée. Les k choisis vont varier entre 3 et 7.

— La troisième analyse consiste à faire changer le paramètre du coût de misclassification avec un k = 5. En effet, la modification de celui-ci permettrait de gérer l'overfitting. Afin de prendre un grand panel de coûts, les valeurs sélectionnées sont :

0.001 - 0.01 - 0.1 – 1 – 10 – 100 – 1000 – 10000 – 1000000

— La quatrième analyse consiste à faire changer le paramètre gamma avec un k = 5. En effet, la modification de celui-ci permettrait de gérer l'overfitting. Afin de prendre un grand panel de gamma, les valeurs sélectionnées sont :

0.0001 – 0.0002 – 0.001 – 0.002 – 0.004

— La dernière analyse consiste à modifier les deux paramètres (gamma et le coût de misclassification). Grâce à cette analyse, il sera possible de choisir les paramètres optimaux pour le SVM.

2.6 Régression logistique

Le deuxième classificateur testé est un classificateur basé sur la régression logistique. Les fonctions utilisées se trouvent dans le Package R «e1071» et dans le Package R « glmnet ». L'objectif est de trouver les paramètres qui vont donner la meilleure précision du classificateur afin de gérer au mieux le phénomène d'overfitting. Différentes analyses ont été réalisées :

— La première analyse s'est faite sur un k = 5. Il y a donc 1024 variables. Ce nombre de variables est important pour un classificateur. L'objectif est de diminuer les

variables en ne prenant que les plus discriminantes selon un t-test sur le set d'entrainement. Les variables ont été analysées par tranches de 50, en commençant à 10.

— La deuxième analyse consiste à appliquer la régression de Lasso afin de diminuer le nombre de variables en entrée. Les lambda seront choisis par la fonction cv.glmnet.

— La dernière analyse consiste à appliquer la régression de Ridge afin de diminuer l'impact des variables peu discriminantes. Les lambda seront choisis par la fonction cv.glmnet.

Chapitre 3

Résultats et analyses

Dans ce chapitre, nous allons présenter les résultats obtenus grâce aux différents outils bio-informatiques utilisés dans le cadre de ce mémoire. Tout d'abord, nous allons examiner les résultats générés par Bagel4 pour tenter de détecter de nouvelles bactériocines dans des génomes complets de Vibrio cholerae. Face aux résultats peu concluants, nous avons été amenés à modifier l'axe de recherche de ce mémoire. Nous continuerons donc par la présentation et l'explication des résultats générés par ce deuxième axe de recherche qui s'est focalisé sur la discrimination entre les séquences plasmidiques et chromosomiques. Cette discrimination plasmidique - chromosomique a été effectuée par une méthode qui génère des k-mers pour caractériser des séquences d'ADN, suivie par une méthode de classification supervisée. Dans ce contexte, deux outils de classification ont été testés : le « Support Vector Machine » et la régression logistique (dans sa version classique ainsi qu'avec des pénalités sur les paramètres dans les versions de Ridge et de Lasso).

3.1 Bagel4

Comme expliqué dans la méthodologie, nous allons à présent utiliser Bagel4 afin de tenter de trouver de nouvelles séquences de bactériocines. Pour cela, nous allons analyser les différents génomes téléchargés dans les bases de données du package R « Genomes »

et NCBI (Genbank).

3.1.1 Bases de données

Package R « Genomes »

Dans cette base de données, nous avons téléchargé les 93 génomes complets (chromosomes et plasmides) de la bactérie V. Cholerae. Nous les avons ensuite uploadés dans Bagel4 afin de générer le tableau des résultats suivant :

(Sous-classe de) bactériocines	Nombres de génomes	Chromosomes
Microcin (sous-classe)	90	1
Colicin E9 (bactériocine)	3	1
Sactipeptide (sous-classe)	1	?

Table 3.1: Nombre de régions découvertes par Bagel4 avec le package « Genomes »

Tout d'abord, la première colonne recense les différentes sous-classes de bactériocines ou classes de bactériocines que comportent ces 93 génomes trouvés par Bagel4. Ensuite, la deuxième colonne recense le nombre de génomes sur lesquels cette région potentielle de bactériocine se localise. Enfin, la troisième colonne recense la localisation de la région sur laquelle se trouve la bactériocine. Il s'agit soit du chromosome 1 soit du chromosome 2.

Nous pouvons constater que la majorité des génomes comportent une région potentielle de bactériocine de sous-classe Microcin se trouvant sur le premier chromosome. De plus, une petite minorité des génomes comportent une région potentielle de bactériocines similaire à la Colicin E9. Celle-ci se trouve sur le chromosome 1. Enfin, un seul génome comporte une région potentielle d'une sous-classe de bactériocines Sactipeptide. Il est impossible de déterminer sur quel chromosome se situe la région. En effet, l'encodage de ce génome réalisé par les scientifiques ne permet pas de différencier les chromosomes.

NCBI (Genbank)

Dans cette base de données, nous avons téléchargé un grand nombre de génomes de la bactérie V. Cholerae.
Tout d'abord, la première analyse a été effectuée sur 139 génomes différents de chromosome 1. Une fois uploadés dans Bagel4, nous obtenons le tableau suivant :

(Sous-classe de) bactériocines	Nombres de génomes
Microcin (sous-classe)	132
Colicin E9 (bactériocine)	5

Table 3.2: Nombre de régions découvertes sur le premier chromosome par Bagel4 avec les séquences provenant de Genbank

Nous pouvons constater que la majorité des génomes comportent également une région potentielle de bactériocines de sous-classe Microcin se trouvant sur le premier chromosome. À nouveau, une minorité des génomes comportent une région potentielle de bactériocines similaire à la Colicin E9.

Ensuite, la deuxième analyse a été effectuée sur 145 génomes de chromosome 2. Une fois uploadés dans Bagel4, nous obtenons le tableau suivant :

(Sous-classe de) bactériocines	Nombres de génomes
Microcin (sous-classe)	3

Table 3.3: Nombre de régions découvertes sur le deuxième chromosome par Bagel4 avec les séquences provenant de Genbank

Nous pouvons constater que seule une région potentielle a été trouvée sur 3 génomes de chromosome 2 par Bage 4.

Enfin, la dernière analyse a été effectuée sur 40 génomes de plasmides. Une fois uploadés dans Bagel4, nous obtenons le tableau suivant :

(Sous-classe de) bactériocines	Nombres de génomes
Colicin E9	3

TABLE 3.4: Nombre de régions découvertes sur les plasmides par Bagel4 avec les séquences provenant de Genbank

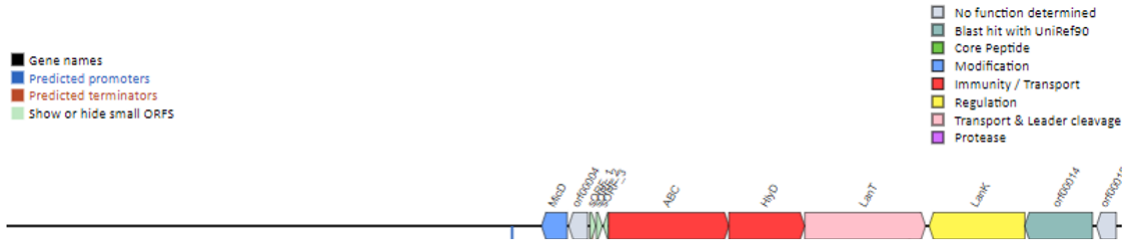
Nous constatons que seulement une région a été trouvée sur 3 plasmides.

3.1.2 Comparaison des régions

Dans l'étape suivante, nous allons comparer les régions qui ont la même sous-classe de bactériocines afin de voir si ces régions sont similaires ou si elles sont différentes. Si les régions sont différentes, cela augmenterait le nombre de régions potentielles de bactériocines découvertes.

Pour cela, nous avons observé séparément toutes les régions données par Bagel4.

— Dans les 225 (90+132+3) régions potentielles de Microcin, nous constatons que toutes les régions semblent similaires. En effet, l'agencement des différents ORF se situe au même endroit sur toutes ces régions (ORF = Opening Reading Frame).

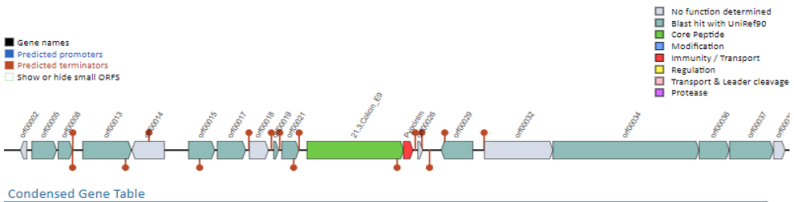


Condensed Gene Table

Name	Function	Motifs
MicD	Cytolysin-activating lysine-acyltransferase RtxC OS=Vibrio cholerae serotype O1 (strain ATCC 39315 / El Tor Inaba N16961) OX=243277 GN=rtxC PE=3 SV=1	PF02794
orf00004		
ABC	ABC-type bacteriocin/lantibiotic exporter	PF00005
HlyD	Leukotoxin secretion protein D OS=Pasteurella haemolytica-like sp. (strain 59438) OX=53500 GN=lktD PE=3 SV=1	PF13437
LanT	ABC-type bacteriocin/lantibiotic exporters	PF00005;PF03412
LanK	Autoinducer 2 sensor kinase/phosphatase LuxQ OS=Vibrio parahaemolyticus serotype O3:K6 (strain RIMD 2210633) OX=223926 GN=luxQ PE=3 SV=1	PF02518;PF00072;PF00512
orf00014	Uncharacterized protein MJ1623 OS=Methanocaldococcus jannaschii (strain ATCC 43067 / DSM 2661 / JAL-1 / JCM 10045 / NBRC 100440) OX=243232 GN=MJ1623 PE=4 SV=1	
orf00015		

FIGURE 3.1: Région potentielle de Microcin

— Dans les 11 (3+5+3) régions de Colicin E9, nous observons que toutes les régions semblent également similaires. En effet, l'agencement des différents ORF se situe au même endroit sur toutes ces régions.



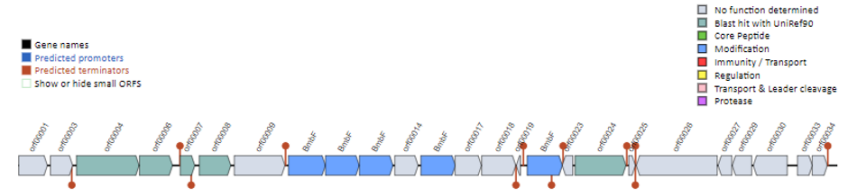
Condensed Gene Table

Name	Function	Motifs
orf00002		
orf00005	HTH-type transcriptional repressor FabR OS=Salmonella paratyphi A (strain ATCC 9150 / SARB42) OX=295319 GN=fabR PE=3 SV=1	
orf00008	Inner membrane protein YijD OS=Escherichia coli O6:H1 (strain CFT073 / ATCC 700928 / UPEC) OX=199310 GN=yijD PE=3 SV=1	
orf00013	Prophage integrase IntZ OS=Escherichia coli (strain K12) OX=83333 GN=intZ PE=3 SV=2	
orf00014		
orf00015	Uncharacterized protein YagK OS=Escherichia coli (strain K12) OX=83333 GN=yagK PE=4 SV=1	
orf00017	Uncharacterized protein YagK OS=Escherichia coli (strain K12) OX=83333 GN=yagK PE=4 SV=1	
orf00018		
orf00019	DNA-binding transcriptional activator AlpA OS=Escherichia coli (strain K12) OX=83333 GN=alpA PE=4 SV=1	
orf00021	Ribonuclease H OS=Cupriavidus necator (strain ATCC 17699 / H16 / DSM 428 / Stanier 337) OX=381666 GN=rnhA PE=3 SV=1	
21.3;Colicin_E9	21.3;Colicin_E9	
PyocinIm	Colicin-E9 immunity protein	PF01320
orf00026		
orf00029	Uncharacterized protein YagA OS=Escherichia coli (strain K12) OX=83333 GN=yagA PE=2 SV=3	
orf00032		
orf00034	DNA topoisomerase 1 OS=Rickettsia conorii (strain ATCC VR-613 / Malish 7) OX=272944 GN=topA PE=3 SV=1	
orf00036	DNA topoisomerase 1 OS=Escherichia coli (strain K12) OX=83333 GN=topA PE=1 SV=2	
orf00037	Putative ribonuclease YeeF OS=Bacillus subtilis (strain 168) OX=224308 GN=yeeF PE=1 SV=2	
orf00039		

Figure 3.2: Région potentielle de Colicin E9

Sur base de cette image générée par Bagel4, la région en vert représenterait une bactériocine Colicin E9 et la région en rouge représenterait son immunité. Bagel4 fournit également la position de ces régions. Il est dès lors possible d'extraire facilement les séquences de leur génome afin de réaliser des analyses plus poussées. Ces analyses détermineraient s'il s'agit effectivement d'une nouvelle bactériocine semblable à la Colicin E9.

— Une seule région de sactipeptide a été trouvée dans tous les génomes.



Condensed Gene Table

Name	Function	Motifs
orf00001		
orf00003		
orf00004	Toxin coregulated pilus biosynthesis protein T OS=Vibrio cholerae serotype O1 (strain ATCC 39315 / El Tor Inaba N16961) OX=243277 GN=tcpT PE=1 SV=2	
orf00006	Toxin coregulated pilus biosynthesis protein E OS=Vibrio cholerae serotype O1 (strain ATCC 39315 / El Tor Inaba N16961) OX=243277 GN=tcpE PE=1 SV=1	
orf00007	Major structural subunit of bundle-forming pilus OS=Escherichia coli O127:H6 (strain E2348/69 / EPEC) OX=574521 GN=bfpA PE=1 SV=1	
orf00008	Twitching motility protein OS=Neisseria gonorrhoeae OX=485 GN=pilT PE=3 SV=1	
orf00009		
BmbF	Anaerobic sulfatase-maturating enzyme OS=Clostridium perfringens (strain 13 / Type A) OX=195102 GN=CPE0635 PE=3 SV=1	PF04055
BmbF	Anaerobic sulfatase-maturating enzyme homolog YdeM OS=Escherichia coli (strain K12) OX=83333 GN=ydeM PE=3 SV=2	PF04055
BmbF	Anaerobic sulfatase-maturating enzyme OS=Klebsiella aerogenes OX=548 GN=atsB PE=3 SV=1	PF04055
orf00014		
BmbF	Anaerobic sulfatase-maturating enzyme homolog YdeM OS=Escherichia coli (strain K12) OX=83333 GN=ydeM PE=3 SV=2	PF04055
orf00017		
orf00018		
orf00019		
BmbF	Anaerobic sulfatase-maturating enzyme OS=Klebsiella aerogenes OX=548 GN=atsB PE=3 SV=1	PF04055
orf00023		
orf00024	Exodeoxyribonuclease I OS=Haemophilus influenzae (strain ATCC 51907 / DSM 11121 / KW20 / Rd) OX=71421 GN=sbcB PE=3 SV=1	
orf00025		
orf00026		
orf00027		
orf00029		
orf00030		
orf00033		
orf00034		

Figure 3.3: Région potentielle de sactipeptide

Cette étape nous permet de constater que les régions qui ont la même sous-classe de bactériocines sont similaires. Dès lors, seules 3 analyses doivent être effectuées.

3.1.3 Analyse des 3 régions

Nous avons décidé de commencer par l'analyse de la Colicin E9 pour deux raisons. Tout d'abord, il s'agit d'une bactériocine et non d'une sous classe de bactériocines. Cette analyse sera dès lors plus précise. De plus, cette bactériocine se trouve dans la bactérie E. Coli. Un grand nombre de données existe donc sur cette bactériocine.

Différentes recherches effectuées lors de mon stage montrent que la Colicin E9 est proche phylogénétiquement d'autres bactériocines qui se trouvent également dans E. Coli. Il nous semble dès lors surprenant que Bagel4 soit aussi précis dans la classe de la bactériocine trouvée.

Le téléchargement des séquences proches phylogénétiquement de Colicin E9 a été effectué. Un arbre phylogénétique entre ces séquences est donné ci-dessous afin de montrer la proximité génétique de ces séquences.

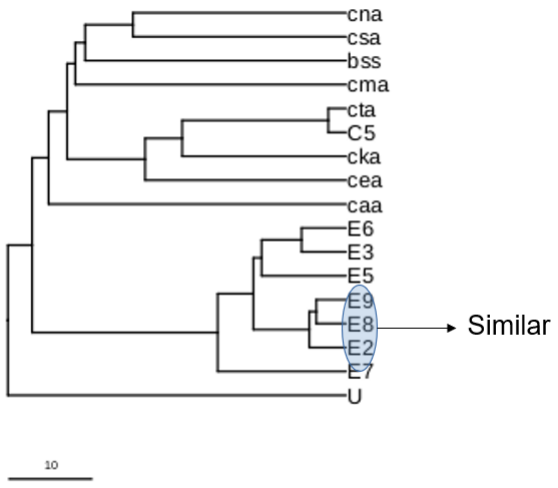


Figure 3.4: Arbre phylogénétique de bactériocines proches de Colicin E9

Nous allons dès lors télécharger plusieurs séquences de chaque sorte de bactériocines proches phylogénétiquement de la Colicin E9 afin d'effectuer un OTU sur les séquences

téléchargées. Nous avons paramétré l'OTU à 97% de similarité car la performance de l'outil est optimale à ce pourcentage-là. Cette même manipulation a été effectuée sur les 11 séquences extraites que Bagel4 considère comme Colicin E9.

Deux arbres phylogénétiques ont alors été construits sur l'analyse des séquences sorties par l'OTU : l'un sur les acides aminés et l'autre sur les nucléotides :

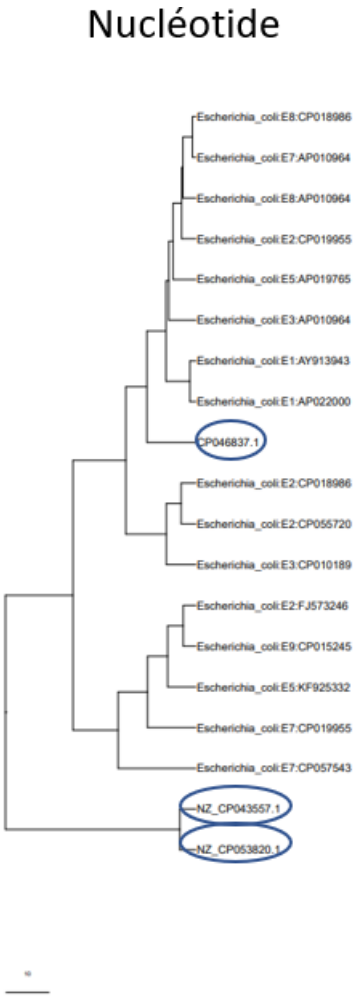


Figure 3.5: Arbre phylogénétique effectué sur les nucléotides

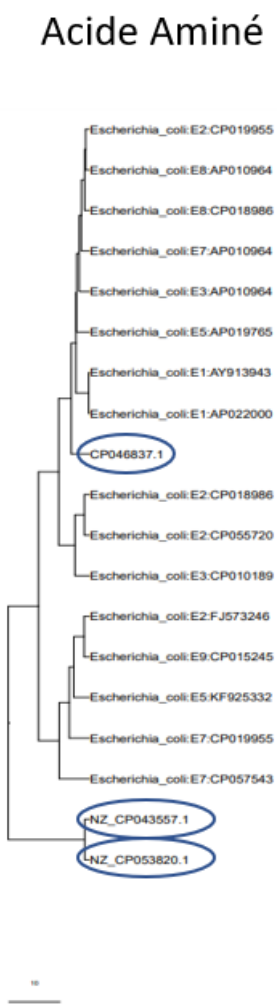


Figure 3.6: Arbre phylogénétique effectué sur les acides aminés

Dans chaque arbre, les séquences entourées par un cercle bleu représentent les séquences provenant de V. Cholerae. Celles qui restent représentent celles de E. Coli. Nous pouvons conclure que les séquences sorties par Bagel4 semblent fortement éloignées des séquences de Colicin E9 dans E. Coli.

3.1.4 Conclusion

L'objectif de ce mémoire était de tenter de découvrir de nouvelles bactériocines grâce à l'utilisation de la méthode basée sur des k-mers et d'outils de classification supervisée. Cependant, les analyses peu concluantes des séquences de Colicin E9 sorties par Bagel4 (outil pourtant investi depuis plus d'une dizaine d'années) confirment que la recherche de nouvelles bactériocines est un domaine complexe qui dépasse la cadre d'un mémoire. C'est pourquoi nous allons modifier l'axe de ce mémoire. Nous gardons la méthode basée sur des k-mers et d'outils de classification supervisée, mais nous allons désormais nous intéresser à la classification de séquences entre les chromosomes et les plasmides.

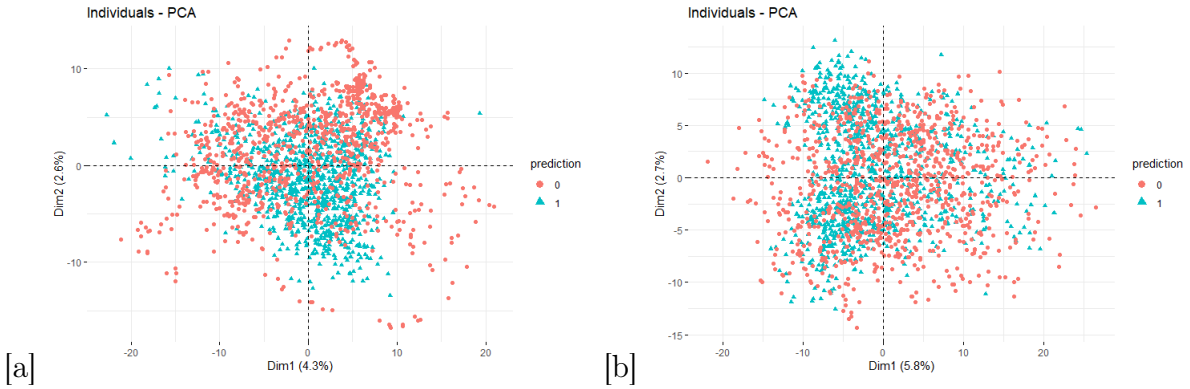
3.2 Discrimination de séquences plasmidiques vs chromosomiques par la méthode basée sur des k-mers et des outils de classification supervisée.

Afin de déterminer si une séquence d'ADN fait partie d'un plasmide ou d'un chromosome, nous avons utilisé la méthode basée sur des k-mers, expliquée dans la partie méthodologie. Cette méthode permet de caractériser chacune des séquences d'ADN en fonction de l'occurrence des k-mers dans les séquences. Grâce à ces caractéristiques, nous pourrons déterminer l'origine des séquences (chromosome ou plasmide).
Etant donné que nous avions pris 1000 séquences (de 1000 nucléotides) de chromosomes et de plasmides, nous avons généré différents tableaux avec 2000 lignes et dont le nombre de colonnes varie en fonction de la taille du k.

Nous allons donc prendre la matrice générée par la méthode basée sur des k-mers et l'insérer dans deux outils de classification supervisée.

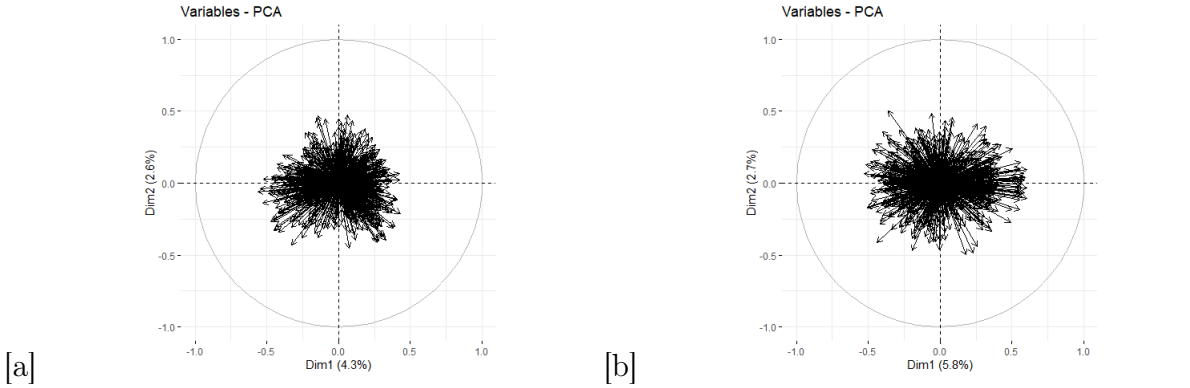
3.2.1 Analyse en composantes principales (ACP)

L'analyse en composantes principales (ACP) avec un k=5 (1024 variables) (fig : 3.7 et 3.8) montre que, comme attendu avec autant de variables, le pourcentage d'informations des deux premières dimensions est très faible (moins de 10%). De plus, aucune tendance à du regroupement n'est observée entre les séquences de même origine sur ces dimensions. Nous n'avons donc pas opté pour cette méthode afin de diminuer le nombre de variables.



[a] [b]

Figure 3.7: Première dimension et deuxième dimension de l'ACP avec un k=5 sur les individus (a = E. Coli et b = V. Cholerae). La valeur 0 représente les plasmides et la valeur 1 les chromosomes



[a] [b]

Figure 3.8: Première dimension et deuxième dimension de l'ACP avec un k=5 sur les variables (a = E. Coli et b = V. Cholerae).

3.2.2 SVM

Plusieurs analyses ont été réalisées avec le SVM.

1. L'objectif de la première analyse est de filtrer les variables en ne prenant que les plus discriminantes selon un t-test sur le set d'entrainement. Cela va permettre de vérifier si le nombre de variables en entrée (très important sans étape de filtre) a un impact sur les performances prédictives du SVM. En effet, un nombre trop important pourrait, par exemple, créer des problèmes d'overfitting si le paramètre de coût de misclassification n'est pas adapté.

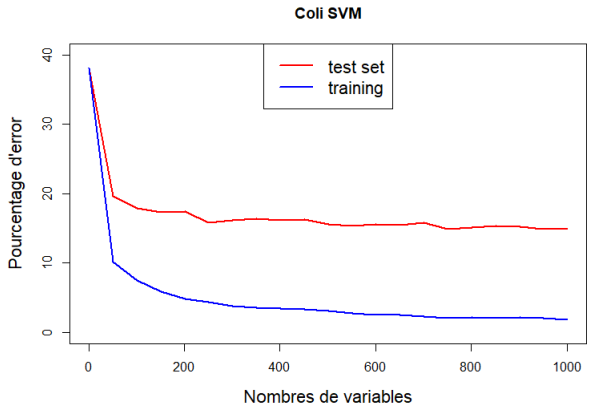


Figure 3.9: La performance du classificateur SVM en diminuant le nombre de variables par un T-test (E. Coli)

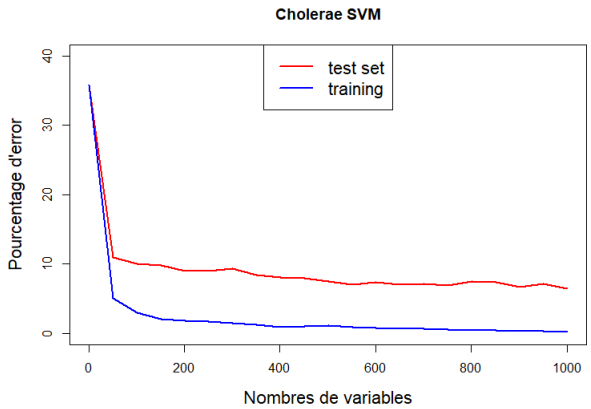


Figure 3.10: La performance du classificateur SVM en diminuant le nombre de variables par un T-test (V. Cholerae)

Nous pouvons remarquer que tant dans le graphe sur E. Coli que dans le graphe sur V. Cholerae, la courbe rouge (test set) ne fait que décroitre. Cela n'est pas étonnant car il est reconnu dans la littérature que le SVM peut gérer un grand nombre de variables. Nous pouvons conclure que la quantité de variables utilisées lorsque k=5 (1024) n'affecte pas négativement les performances prédictives du SVM.
En effet, nous observons que l'erreur est minimale lorsque le nombre de variables est maximal.

	E. coli	V. Cholerae
Set entrainement	1.85%	0.25%
Set test	14.95%	6.50%

Tableau de l'erreur minimale

Cependant, l'utilisation d'un t-test permettant de sélectionner les variables les plus discriminantes n'est pas la plus optimale. En effet, si deux variables sont fortement corrélées entre elles, le t-test aura tendance à donner des résultats similaires et la même décision (d'élimination ou non) sera appliquée à ces deux variables. Il y aura dès lors une redondance dans les variables utilisées. Cette approche univariée générera une diminution des performances prédictives du SVM.

2. Etant donné que le SVM est capable de prendre en entrée 1024 variables, l'objectif de cette deuxième analyse va être d'augmenter le nombre de variables. Nous avons fait varier le k des k-mers de 3 à 7. Le nombre de variables va dès lors varier de 64 (4^3) à 16384 (4^7).

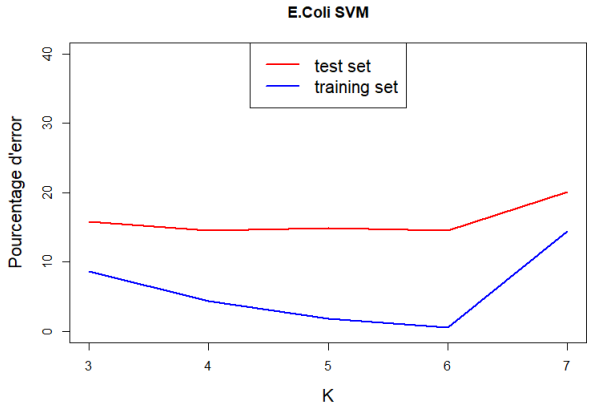


FIGURE 3.11: La performance du classificateur SVM en utilisant des k différents (E. Coli)

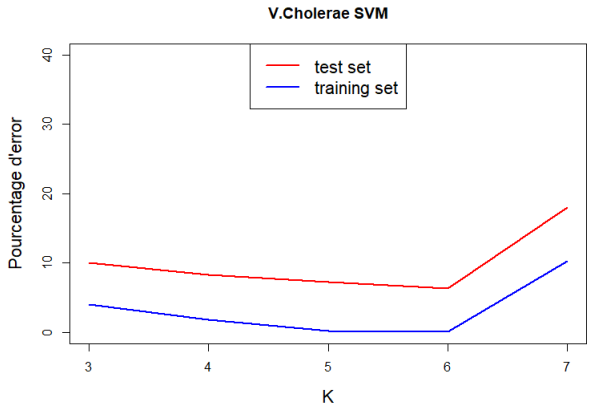


FIGURE 3.12: La performance du classificateur SVM en utilisant des k différents (V. Cholerae)

Les courbes bleues (test entrainement) ne font que décroître pour les k allant de 3 à 6. Nous observons cependant que, lorsque k = 7, les courbes remontent pour les deux espèces. Cela pourrait signifier qu'il y a trop de variables en entrée.

Les courbes rouges (test set) ont l'allure qui est généralement observée lors d'un phénomène d'overfitting. Un k = 5 ou un k = 6 semblent être les paramètres pour les meilleurs résultats. Cependant, les courbes bleues remontent également à partir de k = 7. Cela suggère qu'il ne s'agit en réalité pas d'un overfitting mais plutôt que le SVM ne parvient pas à gérer autant d'entrées.

	E. coli	V. Cholerae
Set entrainement	0.55%(k=6)	0.25%(k=6)
Set test	14.55%(k=4)	6.30%(k=6)

Tableau de l'erreur minimale

3. Dans cette troisième analyse, le paramètre du coût de misclassification sera modifié afin de trouver celui qui donnera les meilleures performances.

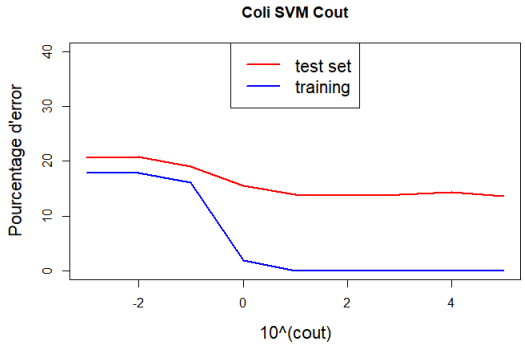


FIGURE 3.13: La performance du classificateur SVM en modifiant le cout de misclassification (E. Coli)

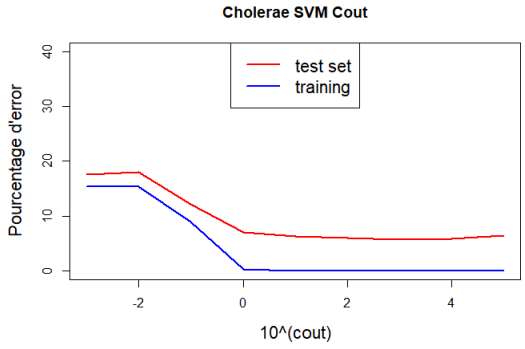


FIGURE 3.14: La performance du classificateur SVM en fonction du coût de misclassification (V. Cholerae)

Les courbes bleues (test entrainement) semblent atteindre le "sans faute" à partir d'un coût égal à 10 pour les deux espèces.

Les courbes rouges n'ont pas l'allure caractéristique observée lors s'un phénomène d'overfitting. En effet, on constate que, lorsque la complexité du classificateur aug-

mente, ces courbes ne remontent pas. Il y a un plateau à partir d'un coût de misclassification égal à 10.

	E. coli	V. Cholerae
Set entrainement	0%	0%
Set test	13.65% (coût =100000)	5.65% (coût =100)

Tableau de l'erreur minimale

4. Dans cette quatrième analyse, le paramètre gamma va être modifié afin de trouver l'optimum des performances pour gérer l'overfitting.

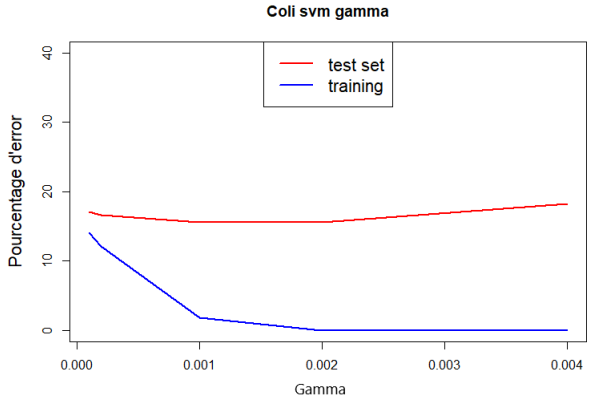


FIGURE 3.15: La performance du classificateur SVM en fonction de gamma (E. Coli)

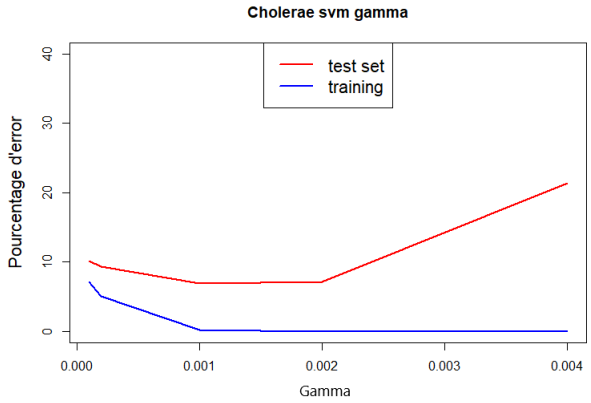


FIGURE 3.16: La performance du classificateur SVM en fonction de gamma (V. Cholerae)

Les courbes bleues atteignent le "sans faute" à partir d'un gamma égal à 0.002

pour les deux espèces.

Les courbes rouges ont l'allure caractéristique d'un overfitting. Les gamma qui donnent les meilleurs résultats sont de 0.001 et 0.002. On peut constater que, lorsque la complexité augmente au-delà de ces gamma, les résultats obtenus sur l'ensemble du set test sont moins bons.

	E. coli	V. Cholerae
Set entrainement	0%	0%
Set test	15.55% (gamma=0.001)	6.95% (gamma=0.001)

Tableau de l'erreur minimale

5. Dans cette dernière analyse, nous allons faire varier les deux paramètres : le gamma et le coût de misclassification. Grâce à cette analyse, il sera plus facile d'observer quelles sont les combinaisons de paramètres qui sont optimales afin de maximiser les performances prédictives.

   La complexité du classificateur est minimale en haut à gauche des tableaux et est maximale en bas à droite. De plus, le paramètre gamma ainsi que la complexité du classificateur augmentent de gauche à droite. De haut en bas, le paramètre de coût de misclassification ainsi que la complexité du classificateur augmentent également.

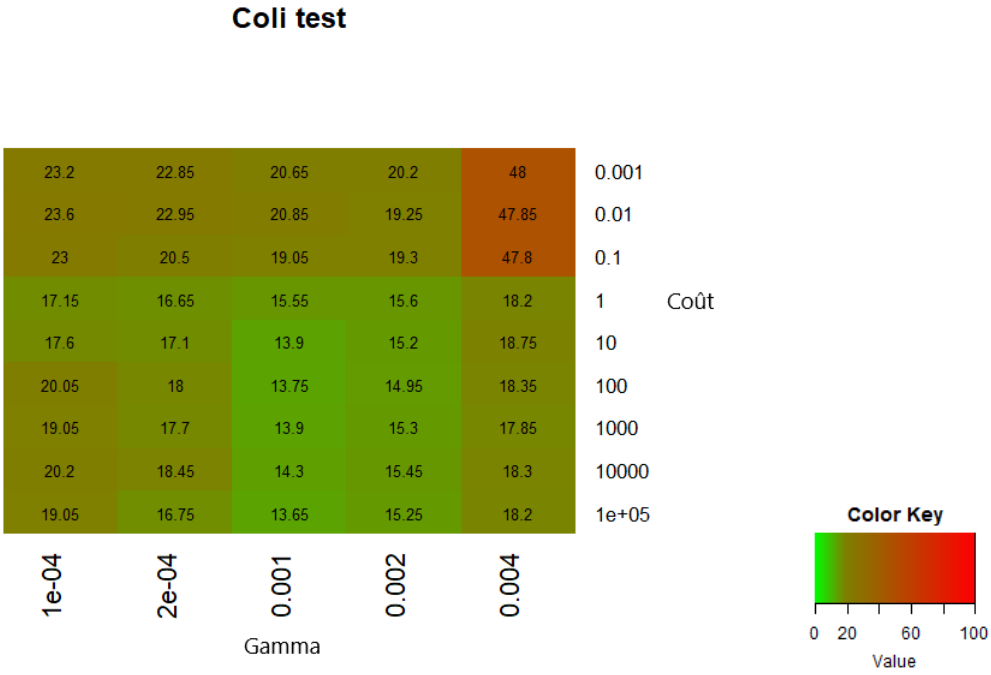


Figure 3.17: La performance du classificateur SVM en fonction de gamma et du coût de misclassification sur le test set (E. Coli)

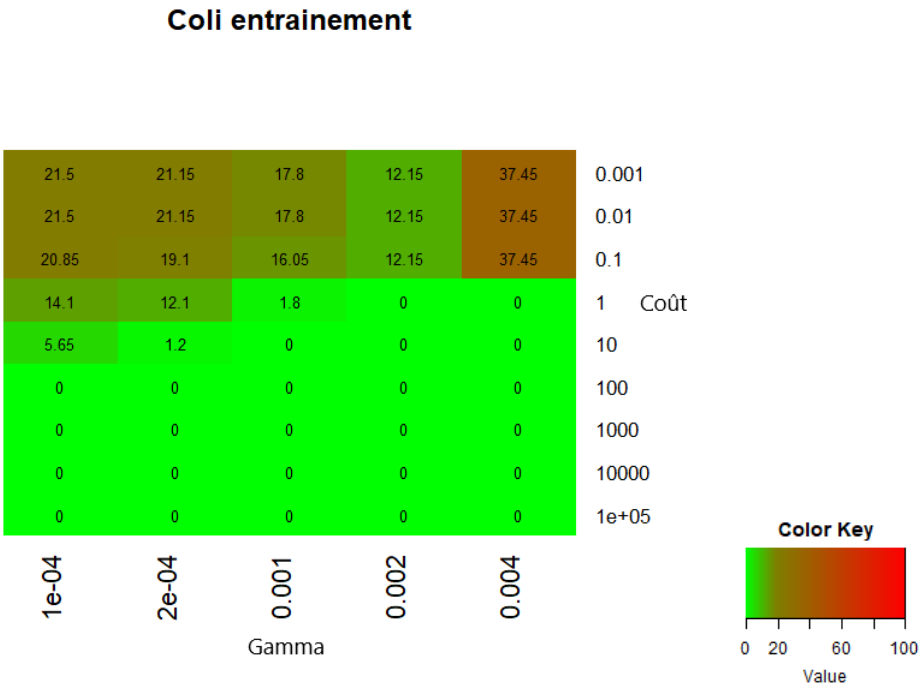


Figure 3.18: La performance du classificateur SVM en fonction de gamma et du coût de misclassification sur le set d'entrainement (E. Coli)

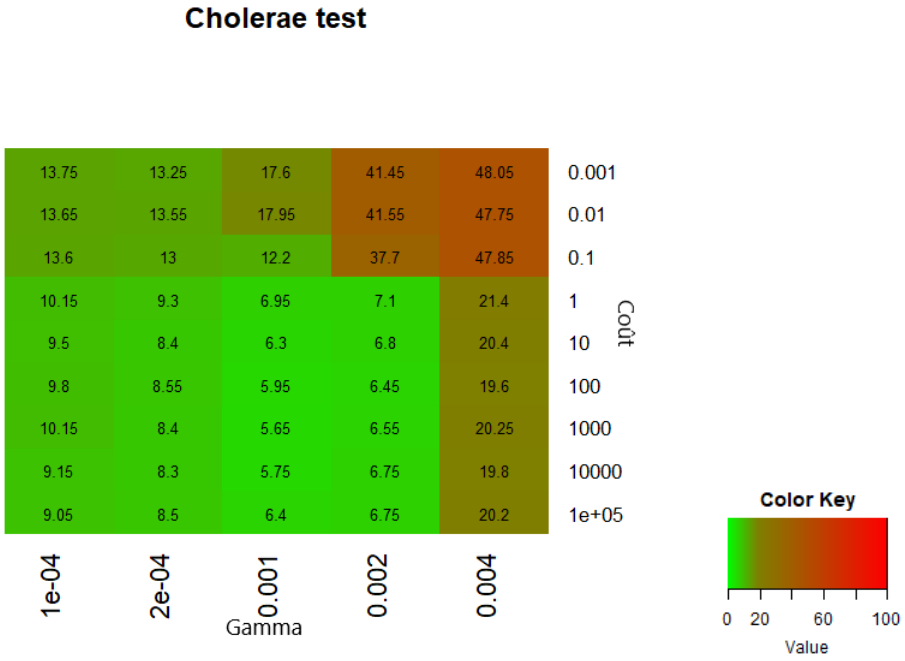


Figure 3.19: La performance du classificateur SVM en fonction de gamma et du coût de misclassification sur le test set (V. Cholerae)

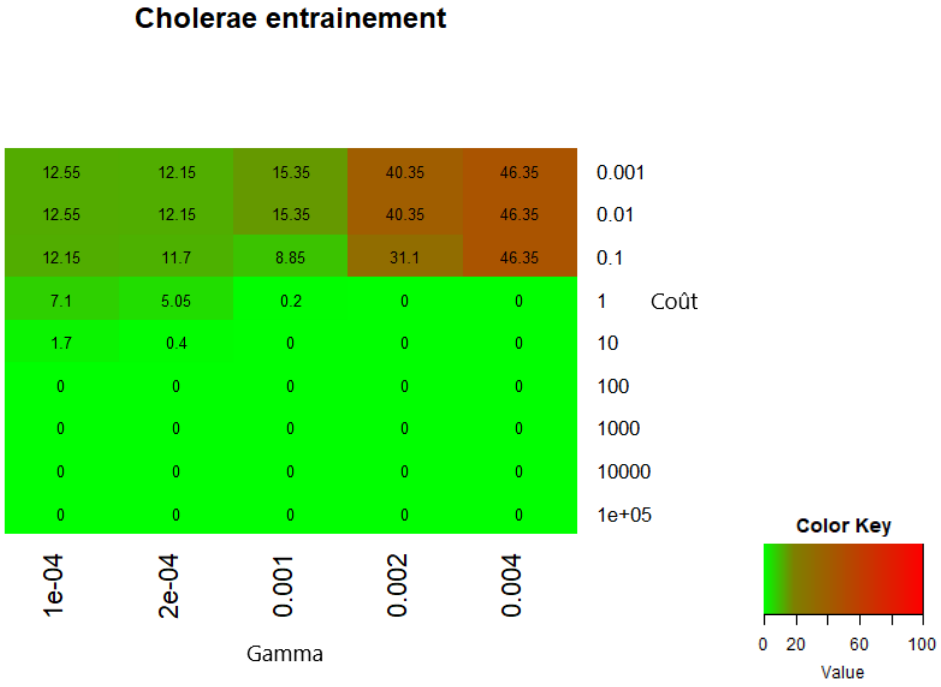


Figure 3.20: La performance du classificateur SVM en fonction de gamma et du coût de misclassification sur le set d'entrainement (V. Cholerae)

L'analyse des tableaux tests, tant pour E. Coli que pour V. Cholerae, nous permet de constater que les extrémités (cases plus rouges), c'est-à-dire quand la complexité du classificateur est trop faible ou trop complexe, génèrent plus d'erreurs. En effet, nous remarquons que les cases centrales sont plus vertes, ce qui signifie qu'un modèle à complexité moyenne donnera de meilleurs résultats.

L'analyse des tableaux entrainement, tant pour E. Coli que pour V. Cholerae, permet de constater que, plus nous nous dirigeons vers le coin inférieur droit, plus l'erreur diminue. Cela correspond parfaitement à la théorie.

	E. coli	V. Cholerae
Set entrainement	0%	0%
Set test	13.65% (gamma=0.001,coût = 100000)	5.65% (gamma=0.001,coût = 1000)

Tableau de l'erreur minimale

6. Conclusion

En conclusion, si nous utilisons la méthode basée sur des k-mers avec l'outil de classification SVM, les paramètres à utiliser afin d'avoir les meilleures performances sont un k = 5 ou k = 6. Nous pouvons noter qu'il n'y a pas de différence significative si le k est de 5 ou s'il est de 6. Cependant, la différence va se faire ressentir sur la durée de calcul qui sera plus longue si le k = 6.

Dans le cas de E. Coli et V. Cholerae, le paramètre optimal minimum à prendre pour le coût de misclassification est de 10. Le paramètre optimal à prendre pour le gamma pour les deux espèces est de 0.001.

Pour E. Coli, en modifiant les paramètres, nous avons réussi à diminuer l'erreur jusqu'à 13.65%. Pour V. Cholerae, nous avons réussi à diminuer l'erreur jusqu'à

5.65%.

De manière générale, l'analyse sur les séquences du V. Cholerae donne de meilleurs résultats que celle sur les séquences d'E. Coli.

3.2.3 Régression logistique

Plusieurs analyses ont été réalisées avec l'outil de classification qui utilise la régression logistique.

1. La première analyse va se faire sur un k = 5. Il y aura donc 1024 variables. Ce nombre de variables est important pour un classificateur. L'objectif est de diminuer les variables en ne prenant que les plus discriminantes selon un t-test sur le set d'entrainement.

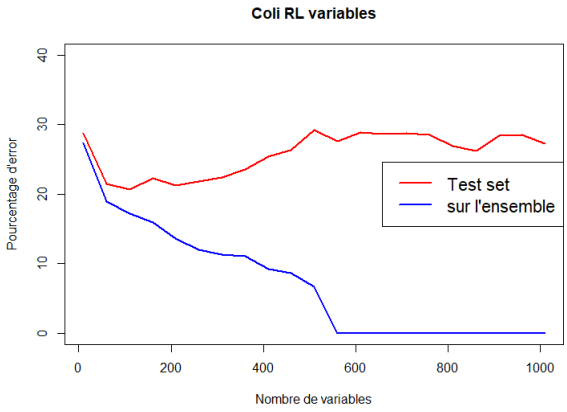


Figure 3.21: La performance du classificateur Régression logistique en diminuant le nombre de variables par un T-test (E. Coli)

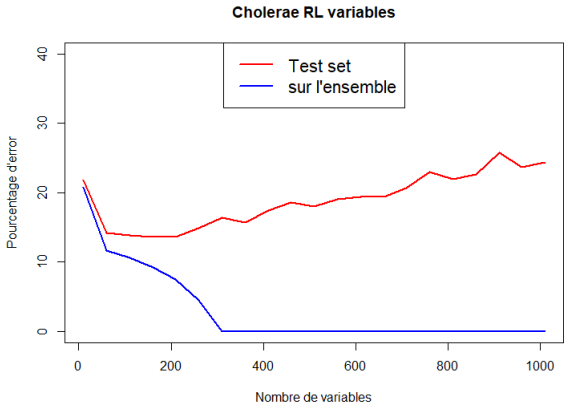


FIGURE 3.22: La performance du classificateur Régression logistique en diminuant le nombre de variables par un T-test (V. Cholerae)

Tant pour E.Coli que pour V. Cholerae, les courbes bleues diminuent avec la complexité du modèle jusqu'à atteindre un taux d'erreur de 0%.

L'allure des courbes rouges semble montrer un overfitting. Selon cette analyse, la performance maximale est atteinte lorsque l'on prend un peu moins de 200 variables.

Nous pouvons observer que, contrairement au SVM, la régression logistique n'a pas de sélection embarquée des variables. En effet, elle n'est pas capable de gérer autant de variables en entrée.

	E. coli	V. Cholerae
Set entrainement	0%	0%
Set test	20.65% (variable=110)	13.65% (variables = 160 et 210)

Tableau de l'erreur minimale

À nouveau, l'utilisation d'un t-test permettant de sélectionner les variables les plus discriminantes n'est pas la plus optimale. En effet, si deux variables sont fortement corrélées entre elles, le t-test aura tendance à donner des résultats similaires et la même décision (élimination ou pas) sera appliquée à ces deux variables. Il y

aura dès lors une redondance dans les variables utilisées. Cette approche univariée générera une diminution des performances prédictives de la régression logistique.

2. Régression de Lasso

   La méthode de Lasso permet de diminuer le nombre de variables en faisant une sélection des variables. Le paramètre à modifier est le lambda. Plus le lambda est grand, plus le nombre de variables avec un coefficient égal à zéro sera grand. Le modèle sera dès lors moins complexe. De même, plus le lambda est petit, plus le nombre de variables avec un coefficient égal à zéro sera faible. Le modèle sera dès lors plus complexe.

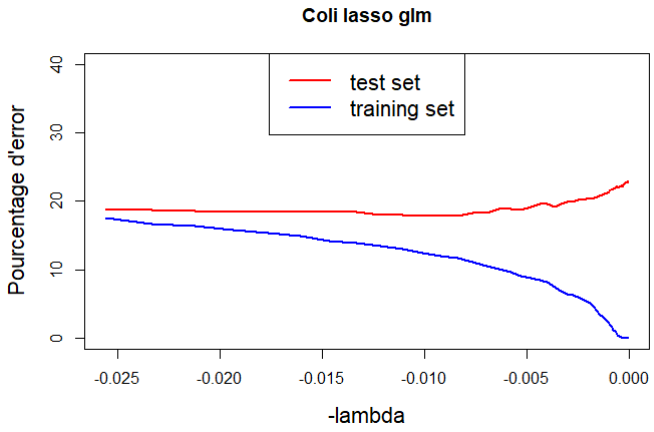


Figure 3.23: La performance du classificateur Régression de Lasso en modifiant le lambda (E. Coli)

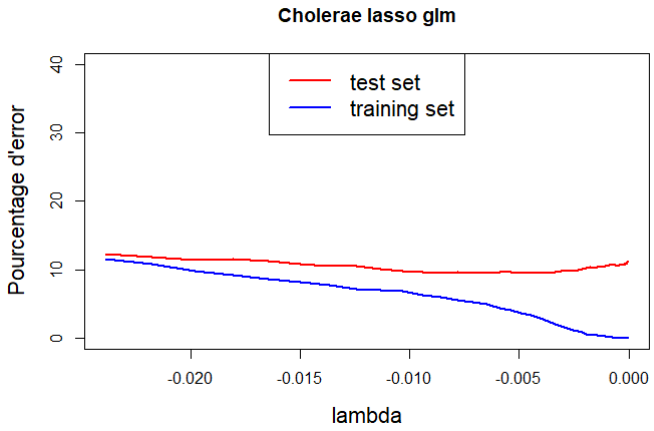


FIGURE 3.24: La performance du classificateur Régression de Lasso en modifiant le lambda (V. Cholerae)

Comme nous aurions pu nous y attendre, les courbes bleues décroissent avec l'augmentation de la complexité du classificateur.

L'allure des courbes rouges semble montrer un phénomène d'overfitting. Le lambda à prendre afin d'avoir le minimum d'erreurs se trouve aux alentours de 0.0065. Cela correspond à prendre les 350 variables les plus discriminantes selon la régression de Lasso.

	E. coli	V. Cholerae
Set entrainement	0%	0%
Set test	17.85% (lambda=0.00838)	9.45% (lambda = 0.00447)

Tableau de l'erreur minimale

3. Régression de Ridge

   La régression de Ridge nous permet de réduire l'amplitude des coefficients. Cela nous permet également d'éviter le sur-apprentissage. Le paramètre à modifier est le lambda. Plus le lambda est grand, plus le nombre de variables avec un coefficient de faible amplitude sera grand. Le modèle sera dès lors moins complexe. De même,

plus le lambda est petit, plus le nombre de variables avec un coefficient de plus grande amplitude sera grand. Le modèle sera dès lors plus complexe.

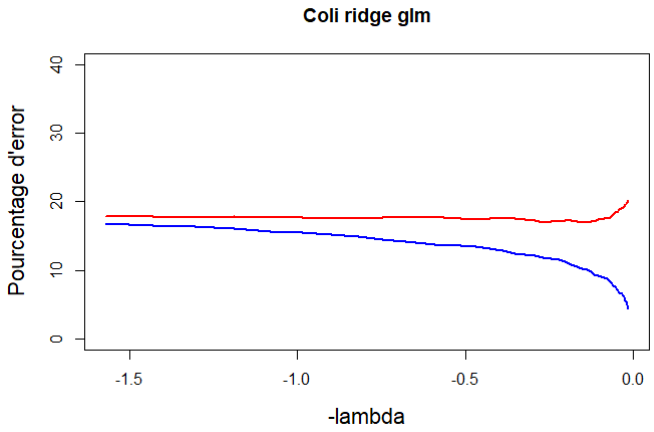


FIGURE 3.25: La performance du classificateur Régression de Ridge en modifiant le lambda (E. Coli)

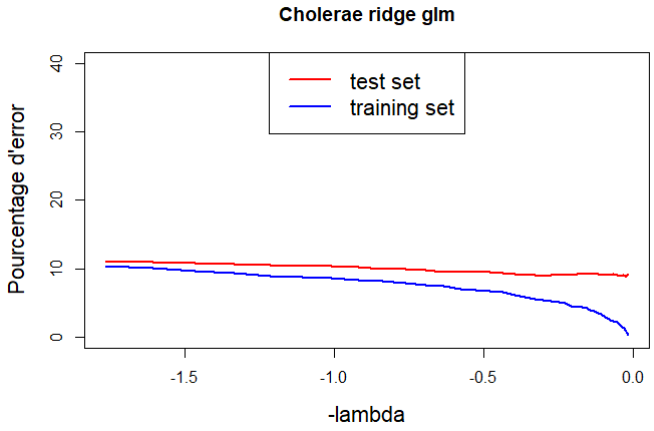


FIGURE 3.26: La performance du classificateur Régression de Ridge en modifiant le lambda (V. Cholerae)

Comme nous aurions pu nous y attendre, les courbes bleues décroissent avec l'augmentation de la complexité du classificateur.

L'allure des courbes rouges semble montrer un problème d'overfitting. Le lambda à prendre afin d'avoir le minimum d'erreurs est égal à 0.1398 pour E. Coli et 0.02442

pour V. Cholerae.

	E. coli	V. Cholerae
Set entrainement	4.3%	0.3%
Set test	16.95% (lambda=0.1398)	8.75% (lambda = 0.02442)

Tableau de l'erreur minimale

4. Conclusion

   Comme nous avons pu le voir avec la première analyse, la régression logistique est soumise à des problèmes d'overfitting lorsque le nombre de variables d'entrée est trop important (par exemple, 1024 variables). Une des solutions possibles face aux analyses effectuées est de faire une sélection de variables en amont (par exemple via un t-test comme démontré dans ce mémoire). Nous pouvons aussi constater que les régressions de Ridge et de Lasso semblent être une bonne alternative car elles permettent d'éviter le phénomène d'overfitting en ajustant un paramètre qui va impacter la complexité du modèle.

   Le minimum d'erreurs qu'il a été possible d'atteindre avec la régression logistique est de 16.95% pour le E. Coli et 8.75% pour le V. Cholerae. Ces deux résultats ont été obtenus avec la méthode de Ridge.

   Comme avec le SVM, les résultats de l'analyse des séquences pour le V. Cholerae sont également meilleurs que ceux obtenus pour E. coli.

3.2.4 Conclusion

Les analyses réalisées sur base des méthodes de SVM et de régression logistique ont pu démontrer leur efficacité. Les performances des classificateurs sont reprises ci-dessous.

	E. coli	V. Cholerae
SVM	13.65%	5.65%
Régression logistique	16.95%	8.75%

Tableau de l'erreur minimale

Sur base de ce tableau, nous pouvons conclure que tant pour le E. Coli que pour le V. Cholerae, la méthode du SVM est la plus performante.

Chapitre 4

Conclusion

L'objectif initial de ce mémoire était, grâce à l'utilisation des k-mers et d'outils de classification supervisée, de découvrir de nouvelles bactériocines dans les génomes du Vibrio Cholerae. Une analyse avec l'outil le plus utilisé dans la recherche de nouvelles bactériocines -Bagel4- a été au préalable réalisée afin de s'assurer que ce domaine n'était pas trop complexe dans le cadre d'un mémoire. Or, les résultats de Bagel4 n'ont pas été concluants. En effet, les analyses plus poussées effectuées sur une région potentielle de bactériocines trouvée par cet outil ont démontré qu'il y avait très peu de correspondance avec la bactériocine jugée similaire par Bagel4. Face à ce constat, nous avons modifié l'axe de ce mémoire.

Dès lors, l'objectif final de ce mémoire maintenait l'utilisation des k-mers et d'outils de classification supervisée pour cette fois-ci prédire si une séquence fait partie d'un chromosome ou d'un plasmide. En effet, le séquençage actuel donne des séquences (contigs) dont l'origine plasmidique ou chromosomique est inconnue. Or, si un gène est découvert dans un de ces contigs, il est important de savoir si ce gène fait partie d'un chromosome ou d'un plasmide. Grâce à cette information, nous pourrons être capables de connaitre le mode de transmission de ce gène qui diffère en fonction de sa localisation. Nous nous sommes focalisés sur deux espèces différentes : Escherichia Coli et Vibrio Cholerae. Ces

deux espèces ont été analysées séparément.

Les outils de classification utilisés sont le SVM dont le noyau est le RBF et la régression logistique. Nous avons réalisé différentes analyses afin de déterminer les paramètres optimaux de ces outils de classification pour traiter au mieux le phénomène d'overfitting. Pour l'outil de classification SVM, deux paramètres sont à prendre en compte : le gamma et le coût de misclassification. Selon nos résultats, le gamma optimal à prendre est de 0.001 et le coût de misclassification doit être supérieur à 10 et cela, pour les deux espèces. Concernant la régression logistique, nous avons effectué deux pénalités sur le vecteur des coefficients : Ridge et Lasso. Pour la pénalité de Ridge, le lambda optimal concernant le V. Cholerae est de 0.02 et pour E. Coli, il est de 0.14. Pour la pénalité de Lasso, le lambda optimal concernant le V. Cholerae est de 0.00447 et pour E. Coli, il est de 0.00838.

Les analyses réalisées sur base des méthodes de SVM et de régression logistique ont pu démontrer des taux d'efficacité différents en fonction des deux espèces. Pour V. Cholerae et E. Coli, il s'est avéré que l'outil de classification SVM est plus performant que celui qui utilise la régression logistique même en appliquant les pénalités de Lasso et de Ridge. En effet, pour V. Cholerae, le SVM a un taux d'erreur de 5.65% tandis que la régression logistique a un taux d'erreur de 8.75%. Pour E. Coli, le SVM a un taux d'erreur de 13.65% tandis que la régression logistique a un taux d'erreur de 16.95%.

Enfin, nous avons comparé les résultats apportés par mlplasmid et ceux obtenus dans ce mémoire. Pour Escherichia Coli, mlplasmid atteint un taux d'erreur de 5%, ce qui est beaucoup plus faible que celui atteint par nos méthodes. Mlplasmid est dès lors plus performant que nos méthodes pour cette espèce de bactéries. Mlplasmid n'a pas effectué de tests sur le Vibrio Cholerae. Cependant, nos résultats pour V. Cholerae (5.65% obtenus par le SVM et 8.75% obtenus par la régression logistique avec une pénalité de Ridge) sont dans la même fourchette que les résultats obtenus par mlplasmid sur d'autres espèces

(5%-8%-6%).

Nous observons que les résultats obtenus sur le V. Cholerae avec nos deux outils de classification apportent un taux d'erreur semblable à ceux obtenus par mlplasmid sur les autres espèces. Nous pouvons dès lors conclure que l'utilisation de k-mers avec les outils de classification SVM et régression logistique est performante dans la discrimination chromosomique ou plasmidique de contigs provenant de V. Cholerae. Cependant, les résultats obtenus dans ce mémoire concernant Escherichia Coli sont moins concluants que ceux obtenus par mlplasmid.

Chapitre 5

Pistes d'amélioration

De manière générale, nous nous sommes intéressés uniquement à la précision et non pas à la sensibilité et la spécificité des classificateurs. Afin d'améliorer les performances des classificateurs, une piste serait de s'intéresser à la modification de la sensibilité et de la spécificité de ces classificateurs. En effet, mlplasmid a modifié ces deux paramètres ce qui pourrait expliquer ses meilleurs résultats pour E. Coli. La modification de ces paramètres lors de futures analyses sur V. Cholerae pourrait dès lors améliorer davantage les résultats.

	Statut Réel		
Résultat		Chromosome	Plasmide
	Chromosome	VC(Vrai chromosome)	FP(Faux plasmide)
	Plasmide	FC(Faux chromosome)	VP(Vrai plasmide)
		Sensibilité	Spécificité

La quantité peu importante de séquences entières de plasmides du V. Cholerae dans les différentes bases de données biaise peut-être les résultats obtenus sur ces séquences. Une deuxième piste d'amélioration dans la recherche de ce mémoire serait de trouver d'autres séquences de plasmides de V. Cholerae afin de diminuer le risque de redondance.

De plus, une troisième piste d'amélioration serait de tester d'autres classificateurs afin de les comparer avec ceux utilisés dans ce mémoire. Ainsi, un classificateur plus performant

pour la discrimination plasmidique - chromosomique pourrait être trouvé.

Enfin, au vu des bons résultats obtenus avec des séquences de 1000 nucléotides, il serait intéressant d'essayer des séquences de moins de 1000 nucléotides car cela permettrait de prendre en compte des plasmides encore plus petits.

Table des figures

Bibliographie

[1] OMS. Résistance aux antibiotiques, 2020. [En ligne ; Page disponible le 31 juillet 2020].

[2] Osnat Gillor, Lisa M Nigro, and Margaret A Riley. Genetically engineered bacteriocins and their potential as the next generation of antimicrobials. *Current pharmaceutical design*, 11(8) :1067–1075, 2005.

[3] Daniel E Dykhuizen. Santa rosalia revisited : why are there so many species of bacteria ? *Antonie van Leeuwenhoek*, 73(1) :25–33, 1998.

[4] T Gregersen. Rapid method for distinction of gram-negative from gram-positive bacteria. *European journal of applied microbiology and biotechnology*, 5(2) :123–127, 1978.

[5] Wikipédia. Gram positif — wikipédia, l'encyclopédie libre, 2020. [En ligne ; Page disponible le 7-juin-2020].

[6] Ping Wang, Lydia Robert, James Pelletier, Wei Lien Dang, Francois Taddei, Andrew Wright, and Suckjoon Jun. Robust growth of escherichia coli. *Current biology*, 20(12) :1099–1103, 2010.

[7] Shah M Faruque, M John Albert, and John J Mekalanos. Epidemiology, genetics, and ecology of toxigenic vibrio cholerae. *Microbiology and molecular biology reviews*, 62(4) :1301–1314, 1998.

[8] Tibor Vellai and Gabor Vida. The origin of eukaryotes : the difference between prokaryotic and eukaryotic cells. *Proceedings of the Royal Society of London. Series B : Biological Sciences*, 266(1428) :1571–1577, 1999.

[9] Laura Fdez. Roldán. Différences entre cellules eucaryotes et procaryotes, 2021. [En ligne ; Page disponible le 25 mars 2021].

[10] Wikipédia. Plasmide — wikipédia, l'encyclopédie libre, 2021. [En ligne ; Page disponible le 4-avril-2021].

[11] Uzbrussel. L'adn nucléaire, 2019. [Online ; accessed 30/10/2019].

[12] MONIKA TUNJIĆ and PETRA KORAČ. Vertical and horizontal gene transfer in lichens. *Periodicum biologorum*, 115(3) :321–329, 2013.

[13] Kartable. Cours de la diversification génétique à la diversification du vivant.

[14] J Lamoril, N Ameziane, J-C Deybach, P Bouizegarène, and M Bogard. Les techniques de séquençage de l'adn : une révolution en marche. première partie. *Immuno-analyse & Biologie Spécialisée*, 23(5) :260–279, 2008.

[15] Sang Tae Park and Jayoung Kim. Trends in next-generation sequencing and a new era for whole genome sequencing. *International neurourology journal*, 20(Suppl 2) :S76, 2016.

[16] Lin Liu, Yinhu Li, Siliang Li, Ni Hu, Yimin He, Ray Pong, Danni Lin, Lihua Lu, and Maggie Law. Comparison of next-generation sequencing systems. *Journal of Biomedicine and Biotechnology*, 2012, 2012.

[17] Wikimedia Commons. File :sanger-sequencing.svg — wikimedia commons, the free media repository, 2020. [Online ; accessed 18-August-2021].

[18] Todd R Klaenhammer. Bacteriocins of lactic acid bacteria. *Biochimie*, 70(3) :337–349, 1988.

[19] Margaret A Riley and John E Wertz. Bacteriocin diversity : ecological and evolutionary perspectives. *Biochimie*, 84(5-6) :357–364, 2002.

[20] Anne de Jong, Sacha AFT van Hijum, Jetta JE Bijlsma, Jan Kok, and Oscar P Kuipers. Bagel : a web-based bacteriocin genome mining tool. *Nucleic acids research*, 34(suppl_2) :W273–W279, 2006.

[21] Paulien Hogeweg. The roots of bioinformatics in theoretical biology. *PLoS computational biology*, 7(3) :e1002021, 2011.

[22] Abdesselem Dakhli. *Classification des séquences d'ADN par les réseaux d'ondelettes.* PhD thesis, ENIS-SFAX, 2017.

[23] Robert C Gentleman, Vincent J Carey, Douglas M Bates, Ben Bolstad, Marcel Dettling, Sandrine Dudoit, Byron Ellis, Laurent Gautier, Yongchao Ge, Jeff Gentry, et al. Bioconductor : open software development for computational biology and bioinformatics. *Genome biology*, 5(10) :1–16, 2004.

[24] Marianna Milano. Computing languages for bioinformatics : R. 2019.

[25] Rayan Chikhi and Paul Medvedev. Informed and automated k-mer size selection for genome assembly. *Bioinformatics*, 30(1) :31–37, 2014.

[26] Wikipédia. K-mer — wikipédia, l'encyclopédie libre, 2021. [En ligne ; Page disponible le 1-mai-2021].

[27] Yuh-Jiuan Tsay and Jiunn-Yann Chiang. Cbar : an efficient method for mining association rules. *Knowledge-Based Systems*, 18(2-3) :99–105, 2005.

[28] Pawel S Krawczyk, Leszek Lipinski, and Andrzej Dziembowski. Plasflow : predicting plasmid sequences in metagenomic data using genome signatures. *Nucleic acids research*, 46(6) :e35–e35, 2018.

[29] Sergio Arredondo-Alonso, Malbert RC Rogers, Johanna C Braat, Tess D Verschuuren, Janetta Top, Jukka Corander, Rob JL Willems, and Anita C Schürch. mlplasmids : A user-friendly tool to predict plasmid-and chromosome-derived sequences for single species. *Microbial genomics*, 4(11), 2018.

[30] Auke J van Heel, Anne de Jong, Chunxu Song, Jakob H Viel, Jan Kok, and Oscar P Kuipers. Bagel4 : a user-friendly web server to thoroughly mine ripps and bacteriocins. *Nucleic acids research*, 46(W1) :W278–W281, 2018.

[31] Fabien Javerliat. *Étude de la structuration des communautés procaryotes des drainages miniers acides par le séquençage haut débit*. PhD thesis, Université de Pau et des Pays de l'Adour, 2014.

[32] Malo Le Boulch. *Taxonomie et inférence fonctionnelle des procaryotes : développement de MACADAM, une base de données de voies métaboliques associées à une taxonomie*. PhD thesis, Toulouse, INPT, 2019.

[33] Robert Edgar. Usearch. Technical report, Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States), 2010.

[34] Qi Zheng, Casey Bartow-McKenney, Jacquelyn S Meisel, and Elizabeth A Grice. Hmmufotu : An hmm and phylogenetic placement based ultra-fast taxonomic assignment and otu picking tool for microbiome amplicon sequencing studies. *Genome biology*, 19(1) :1–20, 2018.

[35] P. Dupont. Lgbio2010 - bioinformatics.

[36] Chamila C De Silva, Scott P Beckman, Shuaishuai Liu, and Nicola Bowler. Principal component analysis (pca) as a statistical tool for identifying key indicators of nuclear power plant cable insulation degradation. In *Proceedings of the 18th International Conference on Environmental Degradation of Materials in Nuclear Power Systems–Water Reactors*, pages 1227–1239. Springer, 2019.

[37] Abdelhamid DJEFFAL. *Utilisation des méthodes Support Vector Machine (SVM) dans l'analyse des bases de données.* PhD thesis, Université Mohamed Khider-Biskra, 2012.

[38] Pier Paolo Ippolito. Svm : Feature selection and kernels, 2019. [En ligne ; Page disponible le 3 juin 2019].

[39] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152.

[40] Thorsten Joachims. Making large-scale svm learning practical. Technical report, Technical report, 1998.

[41] Anne-Claire Brunet. *Développement d'outils statistiques pour l'analyse de données transcriptomiques par les réseaux de co-expression de gènes.* PhD thesis, Université Paul Sabatier-Toulouse III, 2016.

[42] Marius Kwémou Djoukoué. *Réduction de dimension en régression logistique, application aux données Actu-Palu.* PhD thesis, Université d'Evry Val d'Essonne, 2014.

[43] Manpreet Singh Minhas. Techniques for handling underfitting and overfitting in machine learning, 2021. [En ligne ; Page disponible le 6 juin 2021].

[44] Douglas M Hawkins. The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1) :1–12, 2004.

[45] Ajitesh Kumar. Overfitting underfitting concepts interview questions, 2020. [En ligne ; Page disponible le 6 décembre 2020].

[46] Daniel Berrar. Cross-validation., 2019.

[47] Mariia Fedotenkova. *Extraction of multivariate components in brain signals obtained during general anesthesia.* PhD thesis, Université de Lorraine, 2016.

[48] Dennis A Benson, Mark Cavanaugh, Karen Clark, Ilene Karsch-Mizrachi, David J Lipman, James Ostell, and Eric W Sayers. Genbank. *Nucleic acids research*, 41(D1) :D36–D42, 2012.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**