Identification of strains of a bacterial species from long reads

Grégoire Romain Siekaniec

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ RENNES 1

ÉCOLE DOCTORALE Nº 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par

Grégoire SIEKANIEC

Identification of strains of a bacterial species from long reads

Thèse présentée et soutenue à Rennes, le 10 décembre 2021
Unité de recherche : Équipe MicroBio, INRAE, UMR STLO, Agrocampus Ouest
Équipe GenScale, Univ Rennes, Inria, CNRS, IRISA
Thèse Nº : 00000

Rapporteurs avant soutenance :

Helene Chiapello	Ingénieure de recherche, HDR, INRAE, Jouy-en-Josas
David Vallenet	Directeur de recherche, LABGeM, CEA, CNRS, Evry

Composition du Jury :

Président :		
Rapporteurs :	Helene Chiapello	Ingénieure de recherche, HDR, INRAE, Jouy-en-Josas
	David Vallenet	Directeur de recherche, LABGeM, CEA, CNRS, Evry
Examinateurs :	Elisa Fromont	Professeure des universités, Univ Rennes, IRISA, Rennes
	Philippe Glaser	Directeur de recherche, Institut Pasteur, Paris
	Romain Chauvet	Chef de Projet R&D, Eurofins , Nantes
Dir. de thèse :	Jacques Nicolas	Directeur de recherche, Univ Rennes, INRIA, Rennes
Dir. de thèse :	Eric Guédon	Directeur de recherche, INRAE, Institut Agro, STLO, Rennes
Invitée :	Emeline Roux	Maître de conférences, Univ Rennes, Rennes

Attention, en cas d'absence d'un des membres du jury le jour de la soutenance, la composition du jury doit être revue pour s'assurer qu'elle est conforme et devra être répercutée sur la couverture de thèse

Remerciements

Tout d'abord, merci à David Vallenet, Hélène Chiapello, Elisa Fromont, Philippe Glaser et Romain Chauvet d'avoir accepté de faire partie de mon jury de thèse et de l'intérêt qu'ils portent à ce travail de thèse. Je suis honoré d'avoir un panel d'experts si diversifié comme jury de thèse.
Merci particulièrement à David Vallenet et Hélène Chiapello pour leurs rapports détaillés sur mon manuscrit de thèse qui ont soulevé des questions intéressantes.

Ensuite merci à mes deux directeurs de thèse, Jacques Nicolas et Eric Guédon sans qui cette thèse n'aurait jamais abouti. Merci tout particulièrement à Jacques pour l'accompagnement durant cette thèse malgré mes difficultés à la fois en matière d'organisation et de rédaction. Tant qu'on parle de rédaction, un grand merci à Jacques pour le grand nombre de corrections des différents documents impliquant la réalisation d'une thèse avec en tête de liste ce manuscrit de thèse.

Merci également à Emeline Roux sans qui cette thèse aurait manqué cruellement de données biologiques à traiter mais également pour son accompagnement durant cette thèse, ses corrections du manuscrit de thèse et pour avoir été la première personne à m'apprendre qu'un yaourt est composé uniquement de souches de *S. thermophilus* et *L. delbrueckii* subsp. *bulgaricus* (ça fait une anecdote de plus pour présenter mon sujet de thèse).

Je remercie l'Inria/IRISA, l'INRAE (STLO), l'école doctorale MathSTIC et l'université Rennes 1 pour m'avoir donné l'opportunité de réaliser cette thèse.

Merci aux membres du Genoscope : Corinne Cruaud, Stefan Engelen, Jean-Marc Aury... ainsi qu'aux membres de l'I2BC : Delphine Naquin, Erwin van Dijk, Yan Jaszczyszyn... pour leur aide dans la compréhension et la mise en place de l'utilisation du MinION aussi bien du côté des manipulations biologiques que de la bioinformatique associée. Merci aussi à Sophie Lemoine pour l'organisation du NanoClub où j'ai pu assister à des talks très intéressants sur l'utilisation du MinION et le traitement des données nanopore.

Merci aux membres de mon CSI, Alexandre Termier, Jean-Marc Aury et Yoan Augagneur pour leurs idées et conseils sur ce sujet de thèse.

Je tiens à remercier ou re-remercier également toutes les personnes ayant relu mon manuscrit de thèse avec en number 1 Jacques Nicolas suivi par Emeline Roux, Eric Guedon, Clara Delahaye, Olivier Dennler, Victor Epain, Téo Lemane, Sophie Le Bars, Albane Lysiak et Kévin da Silva.

Merci à Téo Lemane pour son aide et ses idées sur le développement de ORI.

Merci à Fabrice Legeai et Dominique Lavenier de m'avoir accueilli dans l'équipe GenScale pour mon stage de fin de Master, sans quoi je n'aurais pas commencé cette thèse. Et merci également au deux directeurs successifs de l'équipe GenScale, Dominique Lavenier et Pierre Peterlongo pour l'accueil dans l'équipe durant cette thèse.

Un grand merci à toutes les personnes des équipes de Symbiose (GenScale, Dyliss, Genouest) pour l'ambiance du laboratoire et les discussions farfelues ou non aux pauses midi/café.

Merci aux deux secrétaires Marie Le Roïc (Inria) et Laurence Adamandidis (INRAE) pour l'accueil dans mes deux laboratoires et la gestion des papiers administratifs, ce qui n'est clairement pas mon fort.

Je tiens également à remercier Nicolas Guillaudeux pour les nombreuses discussions qu'on a pu avoir durant la thèse sur tout un tas de sujets allant des mangas à la transcription d'ARN. J'ai également été très content de collaborer avec toi sur la réalisation d'un "magnifique" film pour Sciences en Cour[t]s (SeC) comportant un chien en général et lors de la mise en place du festival de l'année suivante.
Tant qu'on parle de SeC, je remercie Lucas Bourneuf pour son aide sur le tournage du film mais plus généralement pour les longues discussions qu'on a eues sur tout un tas de sujets du plus loufoque au plus sérieux.

Un grand merci à Méline Wéry pour sa bonne humeur et l'écriture de "la bible de Sciences en Cour[t]s" qui nous a grandement aidés dans l'organisation de l'année d'après.
Merci également Raphaël Truffet pour sa gestion de Nicomaque et lors de l'organisation de SeC. Bravo également pour tes talents en comédie et en écriture qui nous auront bien servi durant l'organisation de ce festival.

Je pense que tout doctorant en fin de thèse le sait. Le bon déroulement d'une thèse passe aussi par un soutien moral en dehors du laboratoire que ce soit la famille ou les amis.

Je souhaite alors remercier la famille Le Bars et particulièrement Jean-Charles et Marie-France pour leur accueil durant les différents confinements.

Ensuite, merci au Slackliners bretons de BZ'Slack, particulièrement à Thomas et Sullivan qui ont été les premiers à me rencontrer et à me faire essayer la 100 mètres. Merci également à Mirentxu, Lucie, Barthélemy, Théophile, Clément et tous ceux que je vais oublier (en partie par flemme d'écrire plus de noms) pour leur bonne humeur et les nombreuses heures passées pieds nus dans l'herbe ou suspendu dans le vide. J'espère vous revoir de temps en temps sur une slackline, à l'escalade ou dans un bar.

Un grand merci également à mes amis de longue date Hugo, Ophélie, Sylvain et Rémi pour leur bonne humeur et pour râler quand je ne revenais pas les voir pour les vacances. Je ne garde pas facilement le contact mais ça fait longtemps qu'on se connaît maintenant et j'espère qu'on continuera à se voir dès qu'une occasion se présentera.

Merci également à mes amis Rennais sans qui ces années auraient été plus moroses et beaucoup moins festives.
Parmi ces personnes je souhaite remercier Alexandre pour ces nombreuses heures passées à l'escalade, dans les bars ou chez l'un de nous deux à discuter, boire, jouer, regarder des séries pas forcément incroyables et à qui je pense que je dois pas mal de verre.
Merci également à Linh-Chi pour les origamis, les tests de Rubik's cube et autres casse-têtes.
Merci à Clara Emery sans qui je n'aurais peut-être pas rencontré la promotion de M1 avec qui j'ai passé beaucoup de temps.
Merci à Olivier et Albane sans qui les aventures du siphon bouché auraient été moins

fun.
Merci également à Olivier pour toute ces soirées ~~quelque peu~~ trop arrosées avec du whisky ~~assez~~ trop cher ou des Bloody Mary ~~un peu~~ trop épicés.
Merci également à Victor pour ces discussions passionnantes et parfois très philosophiques en afterwork.
Merci aussi à tous ceux que j'ai oubliés. Si vous vous sentez lésé, je vous propose de discuter de cet oubli autour d'un verre si cela vous dit ?

Enfin merci à ma famille d'être là pour moi depuis aussi loin que je puisse me souvenir. Merci maman de continuer à t'inquiéter et à prendre soin de moi 8 ans après que j'ai quitté la maison. Merci papa pour les discussions sur le sport, la vie et les jeux vidéo ainsi que pour la découverte du fantastique et de la science-fiction sans qui je ne serais pas celui que je suis actuellement. Merci également à Romain pour les albums de rap et à Caroline pour la tasse Yoshi que j'utilise encore. Merci et pardon de ne pas avoir été aussi présent que vous l'auriez espéré. Promis j'essayerais de vous rendre visite plus souvent par la suite.
Merci également à mes grand-parents qui viennent régulièrement aux nouvelles malgré ma mauvaise manie de ne pas en donner.

Je terminerai ces remerciements avec la plus importante durant cette aventure. La personne avec qui j'ai passé le plus de temps et qui m'a soutenu et poussé dans les nombreux moments ou j'étais à la limite d'abandonner ; merci Sophie. Merci d'être toi avec la cohorte de gentils nanimaux qui t'accompagne.

Info de non-intérêt :
Le mot *Merci* a été utilisé 18 fois dans ces remerciements.

Info intéressante mais qui n'a rien à faire ici :
La phrase : *Les Rats Essayent de Courir là Où Finissent les Grands Espaces* permet de se souvenir de l'ordre des rangs taxonomiques de la classification du vivant : *Règne ; Embranchement ; Classe ; Ordre ; Famille ; Genre ; Espèce.*

Contents

List of Figures

List of Tables

Résumé de la thèse

0.1 Introduction

Une bactérie désigne un organisme vivant unicellulaire, microscopique et procaryote. Cela signifie qu'individuellement (pour la plupart) elles ne sont pas visibles à l'œil nu et que leur structure cellulaire ne comprend pas de noyau. Elles diffèrent des archées (un autre groupe de procaryotes) par la composition de leur membrane et de leur ARN ribosomal. Les bactéries se présentent sous de nombreuses formes : sphériques (cocci), en bâtonnets (bacilles) ou hélicoïdales et une cellule bactérienne a habituellement une longueur comprise entre 0,5 et 5 µm.

On trouve des bactéries dans tous les types d'environnement sur Terre, des profondeurs océaniques au sol, en passant par la roche, l'air, et elles peuplent même les animaux, les plantes et les humains. Leur omniprésence leur permet de jouer un rôle important dans une grande majorité des processus géochimiques, par exemple dans les cycles du carbone et de l'azote. Elles sont nécessaires au bon fonctionnement de l'ensemble de l'écosystème et des organismes qui y vivent.

Malgré cette omniprésence, on connaît relativement peu de choses sur le monde des bactéries et leur classification taxonomique reste compliquée. La taxonomie sert à regrouper les organismes sur la base de caractéristiques communes et ces groupes (les taxons) se voient attribuer un rang taxonomique correspondant à leur niveau supposé d'apparition dans l'arbre de la vie. Comme pour les autres micro-organismes, il est beaucoup plus difficile d'établir une classification naturelle des bactéries que des animaux et des plantes, car elles présentent relativement peu de caractéristiques phénotypiques visibles et se reproduisent par des mécanismes rapides et multiples, tant sexués qu'asexués. Parmi ces mécanismes, on peut citer la division binaire simple et les échanges génétiques par conjugaison bactérienne, qui impliquent un transfert horizontal bidirectionnel de matériel génétique entre des bactéries proches les unes des autres. Tout cela fait que la taxonomie des bactéries est mouvante, rendant la définition d'une espèce bactérienne floue. Les niveaux en dessous de l'espèce tel que ceux de la sous-espèce, de la souche et du clone sont encore moins bien définis et, en fonction des définitions utilisées, peuvent parfois se chevaucher.

Cependant, descendre à un niveau inférieur à l'espèce tel que celui de la souche est très important. Prenons l'exemple de l'espèce bactérienne la plus étudiée, *Escherichia coli.* Cette dernière contient des souches pathogènes [Navarro-Garcia 2014] des souches commensales et d'autres probiotiques [Grozdanov *et al.* 2004]. Être capable de différencier les trois types de souches relève donc d'un problème de santé publique.

A l'origine, l'étude des bactéries était réalisée majoritairement à partir de leurs phénotypes visibles ainsi que de leurs types trophiques (source d'énergie, donneur d'électrons et source de carbone) qui dépend de leur métabolisme. Or, pour réaliser ces études, la bactérie doit être cultivée. Cependant, encore aujourd'hui, seulement un petit nombre de bactéries est cultivable et donc observable scientifiquement. Dans leur grande majorité, les bactéries ne peuvent donc pas être séparées par ces méthodes. Avec les avancées de la génomique, l'étude des bactéries basée sur leur génome a permis d'accéder aux bactéries non cultivables. Depuis une dizaine d'années, le nombre de génomes bactériens dans les bases de données augmente rapidement, rendant les études comparatives entre espèces et au sein d'espèces bactériennes possibles.

La comparaison de deux bactéries peut être réalisée sur la base de leur génome. On peut le faire de façon minimale sur la base d'une séquence partielle du gène de l'ARNr 16S, un gène présent chez tous les procaryotes, contenant des parties variables permettant la distinction entre les espèces et entouré de parties conservées permettant l'accroche d'amorces de PCR (polymerase chain reaction), une technique moléculaire permettant l'amplification du morceau d'ADN présent entre les amorces. Une autre façon de faire, appelée MLST (Multi Locus Sequence Typing), consiste à se baser sur la comparaison de 5 à 7 gènes de ménage (assurant des fonctions indispensables à la survie des cellules et donc très conservés) permettant d'étudier les espèces et souches bactériennes plus en détail.

Cependant ces approches n'utilisent qu'une petite partie de l'information des génomes et manquent donc de robustesse pour discriminer correctement des souches bactériennes.

Pour comparer des génomes de souches bactériennes d'une même espèce, il est possible d'utiliser la totalité du génome de ces souches et calculer des indices de similarité entre ces génomes regroupés en anglais sous le terme OGRI (indices de parenté génomique globale). L'indice le plus couramment utilisé est le taux d'identité nucléotidique moyen (ANI) qui se calcule en divisant un génome en fragments qui sont ensuite recherchés dans l'autre génome à l'aide d'un algorithme d'alignement de séquences. Une valeur d'identité nucléotidique est renvoyée à la suite de ce calcul et la valeur ANI est simplement la moyenne de ces valeurs. Par la suite, [Konstantinidis and Tiedje 2005] a montré que l'ANI restreint aux gènes conservés dans les génomes peut être utilisé comme mesure robuste de leur distance évolutive. En plus de la distance ANI d'autre distances telle que celle de Jaccard peuvent être déterminées en utilisant la composition en k-mers des génomes pour évaluer la proximité des génomes entres eux.

Du fait de l'avancée des technologies de séquençage, on peut identifier des bactéries présentes dans un échantillon à partir des séquences d'ADN qu'il contient. Les séquenceurs fournissent en sortie des lectures (reads) représentant chacune un morceau de l'un des individus présents dans l'échantillon.

Lors de cette thèse, la technologie de séquençage utilisée était la technologie d'Oxford Nanopore produisant des lectures bien plus longues que des technologies précédentes telle qu'Illumina, mais avec un taux d'erreurs beaucoup plus important. Pour séquencer un morceau d'ADN, cette technologie utilise des nanopores protéiques disposés à la surface d'une membrane sous tension à travers laquelle passent les molécules d'ADN [M. Jain, Fiddes, *et al.* 2015]. En fonction des nucléotides de la molécule d'ADN passant par le nanopore, le courant électrique à travers la membrane est modifié, ce qui produit un signal brut qui est ensuite transformé en une séquence par des logiciels dédiés appelés basecallers.

L'avantage de la technologie d'Oxford nanopore, et plus particulièrement du MinION, qui est le séquenceur utilisé pendant cette thèse, outre la taille des lectures en sortie, est le prix du séquenceur ainsi que ses dimensions réduites et donc sa portabilité, ce qui permet de réaliser les expériences directement en laboratoire sans passer par les plateformes de séquençage et ainsi d'accélérer le processus d'identification bactérienne.

Lors de la thèse le modèle d'étude utilisé était les bactéries lactiques, plus précisément, *Streptococcus thermophilus*, une bactérie alimentaire Gram-positive de forme sphérique, d'une taille de 0,7 à 1 µm, formant des chaînes, thermophile (la croissance optimale se situe entre 35 et 42°C selon la souche étudiée). Nous avons choisi cette bactérie comme espèce modèle en raison de son importance dans les industries alimentaires en tant que starter laitier [Martinović *et al.* 2020] et probiotique potentiel [Uriot *et al.* 2017], mais également, car c'est une espèce contenant des souches étroitement apparentées avec une faible diversité génomique [Alexandraki *et al.* 2019; Hu *et al.* 2020], ce qui rend le problème difficile.

Durant la thèse 31 souches de *S. thermophilus* provenant de la collection du Centre International de Ressources Microbiennes - Bactéries Associées aux Aliments (CIRM-BIA) ont été séquencées et assemblées. Ces souches avaient été séquencées avec la technologie Illumina auparavant. L'utilisation de la technologie de séquençage Nanopore lors de la thèse, a permis d'assembler complètement et correctement leurs génomes à l'aide de l'assembleur hybride Unicycler (version 0.4.7) [Wick, Judd, Gorrie, *et al.* 2017].

Enfin, en plus de *S. thermophilus*, des tests sur des échantillons contenant d'autres bactéries lactiques (produits laitiers), des souches de la bactérie *E. coli* et des souches de *Streptococcus pyogenes* ont également été réalisés.

0.2 - État de l'art actuel de l'identification de souches bactériennes à partir des séquences d'ADN

Le but de l'identification bactérienne à partir d'un ensemble de données de séquençage issues d'un échantillon bactérien, est d'attribuer les individus présents dans cet échantillon au niveau taxonomique le plus fin possible. Nous présentons ici l'état de l'art des logiciels d'identification bactérienne travaillant à partir de séquences génomiques.

Les logiciels utilisés pour l'identification bactérienne sont en grande majorité développés pour traiter des lectures courtes, car ce sont encore actuellement les données les plus répandues.

Pour identifier une souche bactérienne, il faut être capable d'assigner les séquences issues du séquençage à leur génome d'origine ou au génome le plus proche actuellement connu. Il existe deux façons de réaliser cette assignation, soit en alignant les lectures sur les génomes soit en recherchant la présence de fragments de taille fixe.

L'alignement de deux séquences consiste à insérer des espaces (gaps) dans les séquences pour que les séquences obtenues aient la même taille et qu'il y ait au moins un caractère qui ne soit pas un espace à chaque position de l'alignement. Un score peut ensuite être associé à un alignement en fonction du nombre d'espaces et des correspondances (match) ou discordances (mismatch) observées entre deux nucléotides alignés. Cependant trouver une solution optimale au problème d'alignement se fait en temps quadratique en fonction de la taille des séquences et des heuristiques ont dû être développées pour accélérer le calcul. La plus connue étant l'heuristique seed-and-extend de BLAST [Altschul *et al.* 1990]. Elle consiste à rechercher une petite séquence (graine) identique entre les deux séquences puis d'étendre l'alignement des deux côtés par programmation dynamique. La recherche de cette graine a par la suite été grandement accélérée par l'utilisation d'une structure compressée, le FM-index [Ferragina and Manzini 2000].

Pour échapper à la complexité de l'alignement, il est possible de décomposer les génomes en fragments nommés k-mers. Un k-mer est une sous-séquence de taille k d'une séquence. Cette décomposition en k-mers fait perdre la notion de continuité entre les k-mers. Pour la conserver, il est possible d'utiliser un graphe de De Bruijn qui est un graphe dirigé dont les nœuds correspondent au k-mers et les arêtes correspondent à un chevauchement de k-1 positions.

Lorsque l'on utilise la décomposition en k-mers, on considère un génome comme un ensemble. Les k-mers de cet ensemble peuvent être intégrés dans un index compressé permettant un gain de place ainsi que la recherche rapide des k-mers communs entre une séquence (la requête) et l'index.

Il existe plusieurs façons d'indexer un génome. On peut par exemple utiliser une structure de données telle que BOSS [Boucher *et al.* 2015] permettent d'indexer des graphes de De Bruijn représentant un génome. Parmi les autres techniques d'indexation sans graphe, on dispose d'index spécialisés dans les requêtes de type présence/absence de k-mers et celles associant de l'information à chaque k-mer. Dans la première catégorie, une structure essentielle est le filtre de Bloom. Les filtres de Bloom peuvent être vus comme des tableaux de bits de taille n initialisés à 0, associés à une ou plusieurs fonctions de hash $h_1..h_m$. Pour insérer un k-mer x dans le tableau, on met toutes les positions $h_1(x)..h_m(x)$ à 1. Ensuite pour rechercher un k-mer y dans le tableau, on regarde si les positions $h_1(y)..h_m(y)$ sont toutes à 1. Cette structure est probabiliste, elle ne produit pas de faux négatif, mais peut produire des faux positifs. Les filtres de Bloom permettent une représentation compacte des génomes et sont à la base de notre méthode. Des techniques de compression sans perte existent pour faire en sorte que ces index prennent le moins de place possible.

Pour indexer plusieurs génomes en même temps, il existe des structures de données agrégatives basées sur les index précédents. Si l'on identifie chaque génome par une couleur, les deux grands groupes de méthodes d'agrégation sont :

- les méthodes d'agrégation par couleur où l'on mélange la totalité des k-mers des génomes et on associe ensuite chaque k-mer à leur ensemble de couleurs.

- les méthodes d'agrégation par k-mer, elles, associent d'abord à chaque couleur leur ensemble de k-mers puis cherchent à classifier les couleurs.

Parmi les méthodes d'agrégation par k-mers, on retrouve la structure de données implémentée dans HowdeSBT [Harris and Medvedev 2020] qui va construire un filtre de Bloom pour chaque génome avant d'agréger ces filtres dans une structure sous forme d'arbre. C'est l'une des méthodes actuelle les plus efficaces en terme de place mémoire utilisée pour indexer les jeux de données et c'est celle que nous avons utilisée pour réaliser l'index de notre propre méthode.

Une fois qu'une lecture a été assignée à un certain nombre de génomes de la base de données initiale, identifier les bactéries d'un échantillon consiste le plus souvent à effectuer l'union des possibilités. Cette stratégie consiste à identifier la lecture par le plus petit ancêtre commun dans la taxonomie des organismes dont le génome a été assigné. Elle a été utilisée par des logiciels tels que Kraken [Wood and Salzberg 2014] et Kraken 2 [Wood, Lu, and Langmead 2019]. Cependant l'utilisation de l'ancêtre commun le plus proche fait que l'identification a du mal à descendre au niveau taxonomique de l'espèce voire du genre bactérien. Il est donc très difficile d'identifier des souches avec cette approche et l'utilisation de lectures courtes rend l'identification de souches bactériennes quasiment impossible.

Il existe cependant des logiciels d'identification de souches bactériennes basés sur les lectures courtes tels que le logiciel StrainSeeker [Roosaare *et al.* 2017], développé pour identifier des isolats bactériens, mais qui fonctionne également sur de petits mélanges de souches bactériennes.

Les lectures longues contiennent plus d'information que les courtes du fait de leur taille, mais leur fort taux d'erreurs rend l'assignation de ces lectures à une souche plus compliquée. Cependant, dans cette thèse, nous partons de l'hypothèse que la taille des lectures longues est assez importante pour ne pas nécessiter de les corriger.

Assez peu de logiciels d'identification de souches bactériennes utilisent des lectures longues. Cependant, certains logiciels fonctionnant avec des lectures courtes fonctionnent également avec des lectures longues, tels que Centrifuge [D. Kim *et al.* 2016], Kraken 2 ou encore StrainSeeker. D'autres comme MEGAN-LR [Huson, Albrecht, *et al.* 2018] ont adapté leur approche à ce nouveau type de données.

Parmi les logiciels spécifiques aux lectures longues, on retrouve Metamaps [A. T. Dilthey *et al.* 2019] et NanoMAP [Hall, Speed, and Woodruff 2020], deux logiciels également basés sur l'alignement de séquences. Le logiciel que nous avons développé durant cette thèse semble cependant être le premier logiciel d'identification de souches bactériennes utilisant les lectures longues fonctionnant avec une approche utilisant les k-mers.

0.3 - Une approche d'identification de souches à l'aide de graines espacées

Ce chapitre présente notre approche pour pallier les erreurs de séquençage lors de l'assignation des lectures aux génomes ainsi que le problème de l'identification de souches d'une même espèce bactérienne.

Lors de l'utilisation de k-mers, si l'on veut assigner une lecture à un génome de référence, un seuil minimum de k-mers communs entre les deux séquences doit être fixé. Ce seuil est régulièrement fixé entre 70% à 90%. Cependant, comme le taux d'erreurs des lectures nanopore est important, nous avons fixé ce seuil à 50% durant la thèse.

Or, si l'on considère un simple modèle binomial et l'hypothèse d'un taux d'erreurs uniforme dans la séquence, il est possible de montrer que si l'on a une lecture de taille 2 000 pb et une taille de k-mer de 15, il faudra en moyenne 4,5% d'erreurs pour que 50% des k-mers de la lecture soient trouvés dans le génome. Ce taux d'erreurs est assez faible si l'on considère ceux de la technologie nanopore. A titre d'exemple le taux d'erreurs moyen dans nos expériences sur *S. thermophilus* était compris entre 5,06% et 4,89% en filtrant pourtant les lectures sur leur qualité et leur longueur. Une expérience sur données réelles a été réalisée pour regarder l'effet de la diminution de la taille des k-mers sur l'assignation de 4 000 lectures de la souche *S. thermophilus* JIM8232 contenant environ 5% d'erreur avec notre approche et en utilisant un index contenant 77 génomes de *S. thermophilus*. Cette expérience montre que plus la taille des k-mers diminue plus le pourcentage de lectures assignées à au moins une souche augmente, mais que le pourcentage de lectures assignées sans ambiguïtés diminue.

Comme le taux d'erreurs des lectures rend leur assignation plus compliquée, en plus de diminuer la taille du k-mer utilisé, il est également possible d'utiliser des graines espacées. Les graines espacées sont des séquences binaires débutant et finissant par un 1 et pouvant être utilisées comme masque lors d'un alignement de deux séquences. Les positions à 1 dans ces graines représentent une correspondance entre les séquences et les 0 représentent des positions joker dont on ne se préoccupe pas. Le pattern comparé en utilisant ce "masque" est appelé q-gram par la suite. Il est également possible d'appliquer cette graine à un k-mer et d'en extraire le q-gram correspondant.

Il a été montré que l'utilisation des graines espacées apportait une amélioration dans la précision des résultats à la fois lors d'alignements de séquences comme avec PatternHunter [B. Ma, Tromp, and M. Li 2002] et lors de l'identification de bactéries non basée sur de l'alignement comme avec Seed-Kraken [Břinda, Sykulski, and Kucherov 2015], une implémentation de Kraken utilisant les graines espacées. Dans notre cas une seule graine 111111001111111 ayant pour but d'être moins sensible aux erreurs de séquençage a été utilisée. Cette graine espacée a été déterminée expérimentalement pour bien fonctionner dans l'identification de souches de *S. thermophilus* parmi un panel de graines de taille et de poids croissants dont le pattern a été optimisé en utilisant le logiciel iedera [Noé 2017].

Une fois les lectures assignées à leur(s) génome(s) de référence ou non assignées (ce qui est le comportement voulu lors de contamination de l'échantillon par d'autres espèces non indexées par exemple), il faut identifier les souches réellement présentes dans l'échantillon. Pour comprendre la difficulté du problème, on a mesuré la proximité des génomes de souches de *S. thermophilus* à deux niveaux :

- le niveau génique avec l'étude du pangénome (ensemble des gènes d'une espèce composé du core génome (gènes communs à toutes les souches) et du génome variable (gènes non présents chez au moins une souche)). L'annotation des génomes et le calcul du pangénome ont été réalisés en utilisant les algorithmes fournis par la plateforme MicroScope [Vallenet *et al.* 2020]. Les noms des gènes ont ensuite été curés manuellement et le pangénome a été utilisé pour réaliser une analyse formelle de concepts [Ignatov 2015] afin de produire les biclusters maximaux $\{souches \times gènes\}$

avec en intention les souches et en extension les gènes. Le tout est calculé avec de l'Answer Set Programming (ASP) [Gebser *et al.* 2012], un langage de programmation déclaratif. Un bicluster maximal peut être vu comme un bicluster dans lequel un ajout dans l'intention provoque forcément une perte dans l'extension.

- le niveau génomique avec la création d'un arbre de classification via un algorithme de neighbour-joining proposé par MicroScope. Les biclusters maximaux ont été appliqués sur cet arbre afin d'obtenir une représentation de l'arbre avec, pour chaque nœud, la liste des gènes spécifiques à ce nœud, le tout disponible en ligne grâce au visualisateur d'arbre phylogénique nommé iTOL [Letunic and Bork 2019] : `https://itol.embl.de/tree/1312541346713115979255 85`. En plus de cela, les distances d'ANI [C. Jain, Rodriguez-R, *et al.* 2018], de Jaccard et de Hamming ont été calculées entre les différentes souches. Nous avons par la suite prouvé que ces distances étaient corrélées et nous avons gardé la distance de Hamming car elle est plus rapide à calculer avec les filtres de Bloom utilisés dans notre méthode. Ces calculs de distances ont permis de produire une heatmap, représentant visuellement les distances entre souches de *S. thermophilus* sur laquelle on observe : (1) certains génomes sont extrêmement similaires et vont donc être très compliqués à identifier les uns par rapport aux autres. Enfin (3), si l'on ajoute les souches d'une autre espèce et d'une autre famille bactérienne ces dernières sont facilement différentiables des souches de *S. thermophilus*. (2) Des clusters de souches plus proches les unes des autres et isolée du reste. On retrouve parmi ces clusters un cluster de souche que nous avions séquencé puis assemblé, et qui est séparé du reste des *S. thermophilus*. Si on compare l'origine de ces souches, on remarque que ces dernières proviennent de produits laitiers italiens traditionnels.

A partir de l'assignation des lectures à un ou plusieurs génomes, nous avons créé la matrice *lectures* $\times$ *génomes* contenant dans chaque case la proportion de k-mers de la lecture retrouvée dans le génome, que nous appellerons α. L'identification est alors réalisée en recherchant un nombre minimum de génomes qui expliquent la totalité des lectures assignées. Comme ce problème est NP-complet, nous avons utilisé l'ASP qui nous permet de trouver une solution exacte tant que le nombre de lectures et de souches n'est pas trop important. Pour s'assurer de cela, un prétraitement des données est réalisé. Premièrement, nous ne gardons que les lectures de bonne qualité en nous basant sur la qualité indiquée dans les fichiers de séquençage bruts ainsi que la longueur de ces lectures. Ensuite, durant l'étape d'identification, les lectures ayant un trop grand nombre d'assignations sont supprimées, car elles ne sont pas assez informatives pour différencier des souches entre elles. Enfin, seuls les génomes des souches avec les meilleures valeurs α sont gardées pour simplifier le calcul.

Dans notre approche, la possibilité est laissée à l'utilisateur de regrouper les souches proches en une seule. Le regroupement de ces souches similaires est réalisé en fixant un seuil sur la distance de Hamming entre les génomes. Dans le cas des souches de *S. thermophilus*, ce seuil a été fixé de manière empirique à $2e^{-4}$ et permet de former 12 regroupements de souche proches. Par la suite dans notre méthode, une visualisation simplifiée de la matrice de distance est fournie sous forme d'histogrammes représentant les paires de distances de Hamming entre souches afin d'aider l'utilisateur à déterminer un seuil de regroupement. Le regroupement des souches proches est réalisé en créant le graphe dont les nœuds correspondent aux souches et les arêtes à une proximité entre deux souches (distance $<$ seuil), puis en réalisant l'union des filtres de Bloom des génomes qui font partie de la même composante connexe.

0.4 - ORI, un nouveau logiciel pour l'identification des souches bactériennes à partir de lectures longues

Ce chapitre présente ORI (Oxford nanopore Reads Identification) [Siekaniec *et al.* 2021] un nouveau logiciel d'identification de souches bactériennes utilisant des lectures longues issues de séquençage nanopore. ORI est disponible sur github à l'adresse : `https://github.com/gsiekaniec/ORI` et peut être installé avec conda.

ORI est basé sur quatre points principaux qui sont :

- l'utilisation de la structure d'index de HowDeSBT ;
- l'utilisation d'une graine espacée permettant d'être moins sensible aux erreurs de séquençage ;
- le regroupement des souches proches ;
- l'identification des souches grâce à une optimisation exacte pour la sélection du nombre minimal de souches permettant d'expliquer la totalité des assignations de lectures observées.

Les trois premiers points étant liés au problème de l'assignation des lectures aux génomes, il a fallu modifier le logiciel HowDeSBT. Les modifications apportées sont :

1. l'insertion à l'intérieur des filtres de Bloom des q-grams à la place des k-mers. Ces q-grams sont utilisés également pendant la requête via l'application d'une graine espacée ;
2. le calcul de la distance de Hamming entre tous les filtres de Bloom, qui est une étape qui utilise beaucoup de temps et de mémoire du fait des complexités respectives en $O(mn^2)$ (temps) et $O(mn)$ (mémoire) avec m la taille des filtres de Bloom et n le nombre de génomes ;
3. la création de l'histogramme permettant le choix du seuil pour le regroupement des souches proches et (4) l'union des filtres de Bloom de ces souches.

Afin de vérifier la compression des génomes ainsi que le temps et la mémoire utilisés lors de la requête de l'index, des expériences ont été réalisées avec des génomes de bactéries lactiques et d'*Escherichia coli*. Les différentes expériences ont montré des taux de compression de plus de 85%. La mémoire maximum utilisée durant la requête est, quant à elle, dépendante du nombre de lectures, mais pas de la taille de l'index alors que le temps de requête est dépendant des deux paramètres, ce qui peut poser problème lorsque le nombre de lectures devient important et que la taille de l'index l'est aussi.

ORI fonctionne en trois parties :

- la création de l'index basée sur la version modifiée de HowDeSBT ;
- la requête de cet index en utilisant les q-grams des lectures. C'est dans cette partie "requête" que les lectures sont filtrées sur leur qualité et leur longueur afin de ne garder que les lectures ayant la meilleure qualité ;
- l'identification des souches présentes dans l'échantillon d'origine à partir des résultats d'assignation de chaque lecture et de l'étape d'optimisation présentée précédemment.

0.5 - Validation de ORI sur les bactéries lactiques et expériences sur *S. pyogenes* et *E. coli*

Ce chapitre présente trois groupes principaux d'expériences visant à identifier des souches bactériennes à partir de longues lectures de nanopores en utilisant ORI. Le premier groupe d'expériences est présenté dans l'article [Siekaniec *et al.* 2021] et correspond

à l'identification de souches de *Streptococcus thermophilus* avec ORI, Kraken 2 et StrainSeeker.
Le deuxième groupe d'expériences correspond à l'identification avec ORI de souches présentes dans différents produits laitiers.
Enfin, le troisième groupe d'expériences correspond à l'identification avec ORI de souches de *Streptococcus pyogenes* puis des souches d'*Escherichia coli* provenant d'un microbiote intestinal de porc.

Les premières expériences consistaient à identifier des souches de *S. thermophilus* dans différents échantillons simulés à partir de mélanges des lectures issues du séquençage nanopore réel de souche de *S. thermophilus* de la collection du CIRM-BIA. 180 expériences ont été réalisées en simulant des mélanges de lectures basées sur 4 paramètres :

- le nombre de souches (4 ou 6) ;
- la proximité des souches (distantes, moyennement proches et proches). Les expériences avec des souches proches contiennent des souches qui possèdent au moins une autre souche qui leur est proche (seuil sur la distance de Hamming entre les filtres de Bloom des deux souches) ;
- le nombre de lectures (1 000, 4 000 et 16 000). Les 1 000 lectures sont sous-sélectionnées aléatoirement parmi les 4 000 lectures qui sont elles-mêmes sous-sélectionnées aléatoirement parmi les 16 000 lectures. Les 16 000 lectures sont sélectionnées aléatoirement en mélangeant le nombre de lectures réelles de chaque souche selon le nombre de souches utilisées et la distribution voulue ;
- la distribution des lectures (distribution uniforme ou distribution avec des souches dominantes et sous-dominantes). Pour les expériences contenant des souches sous-dominantes le nombre de lectures est simplement divisé par deux à chaque souche ajoutée dans l'expérience à part pour la dernière souche. Par exemple pour 4 000 lectures contenant 4 souches on aura 2 000 lectures de la première souche, 1 000 lectures de la seconde souche et 500 lectures des deux dernières.

Pour chaque ensemble de paramètres possible, 5 réplicats ont été réalisés.
Des index contenant les mêmes souches ont été créés pour les trois logiciels comparés, ORI, Kraken 2 et StrainSeeker. Ces index contiennent 77 souches de *S. thermophilus* ainsi qu'un groupe externe composé d'une souche de *L. delbrueckii* subsp. *bulgaricus* et

d'une souche de *Streptococcus macedonicus*. Sur les trois index, ceux d'ORI et de Kraken 2 ont des tailles similaires alors que celui de StrainSeeker est beaucoup plus grand.
Les résultats ont été validés en utilisant trois paramètres :

- la somme des distances de Hamming entre les souches identifiées par les logiciels et leur souche la plus proche. Lors d'une identification complètement correcte cette somme sera égale à 0 ;
- le coefficient de corrélation de Matthews qui mesure le compromis précision/sensibilité de l'identification des souches. Ce coefficient varie entre 1 pour une identification parfaite et -1 pour un désaccord total entre souches identifiées et réelles, 0 signifie que les résultats ne sont pas meilleurs que si les souches avaient été sélectionnées au hasard ;
- un ratio d'ambiguïté consistant à diviser le nombre de souches identifiées par le nombre de souches attendues.

Avant de tester les logiciels dans ces conditions, deux tests d'identification d'isolats ont été réalisés. D'abord l'identification d'une souche simple (*S. thermophilus* JIM 8232) puis d'une souche plus complexe (*S. thermophilus* CIRM-BIA 67). Cette dernière est plus complexe car il existe plusieurs souches proches.
Les résultats d'identification était bons pour les logiciels ORI et StrainSeeker. Kraken 2 a quant à lui donné des résultats peu concluants dès la première identification de souches. Cependant, son identification était correcte au niveau de l'espèce bactérienne.
Sur les expériences de souches en mélange, les résultats d'identification ont été séparés en deux : une distribution de lectures uniforme pour chaque souche versus des souches dominantes et sous-dominantes.
Pour l'ensemble des expériences, les résultats ont été présentés de manière à observer les effets de trois paramètres sur l'identification :

- le nombre de lectures (quantité de données) ;
- le nombre de souches (hétérogénéité) ;
- la proximité des souches (pouvoir de résolution de la méthode).

Avec une distribution uniforme des lectures, Kraken 2 ne donne pas de bons résultats d'identification. StrainSeeker quant à lui donne de bons résultats mais est très sensible au nombre de lectures fournies et donne de mauvaises identifications lorsque le nombre de souches augmente ou diminue trop. ORI semble assez robuste au changement de

paramètres et donne de bons résultats d'identification.
Pour les expériences avec des souches dominantes et sous-dominantes, le but était de regarder si les souches sous-dominantes étaient bien identifiées par les méthodes. Sur ces expériences StrainSeeker (qui est sensible au nombre de lectures) donne de bons résultats d'identification (voir même meilleurs qu'ORI selon les conditions) lorsque le nombre de lectures par souche n'est ni trop élevée ni trop faible. C'est également le cas de ORI qui peine à identifier les souches sous-dominantes lorsque le nombre de lectures par souche devient trop faible. Kraken 2 quant à lui s'améliore un peu et ne se montre pas sensible au nombre de lectures utilisées pour l'identification.
La dernière expérience de cette section était de tester l'effet qu'avait le regroupement des souches proches en une seule souche avec ORI. Globalement, regrouper les souches proches améliore les résultats d'identifications sur l'ensemble des expériences. Les résultats d'identification des souches sous-dominantes reste cependant peu concluants lorsque le nombre de lectures par souche est trop faible ($<$ 125 lectures par souche).

Le deuxième groupe d'expériences a consisté à identifier des souches parmi 5 produits laitiers commerciaux. Ces produits étaient :

- 3 yaourts brassés. Un yaourt est composé obligatoirement et uniquement de souches des deux espèces *S. thermophilus* et *L. delbrueckii* subsp. *bulgaricus.*
- 1 lait fermenté contenant des *Bifidobacterium* en plus des souches de *S. thermophilus* et *L. delbrueckii* subsp. *bulgaricus.*
- 1 lait ribot (produit traditionnel Breton. Babeurre fermenté).

Ce groupe d'expérience avait pour but de montrer la faisabilité de l'utilisation d'un protocole "classique" d'extraction d'ADN et d'une mini flowcell Nanopore, la Flongle (60-80 pores actifs par rapport aux 1 300-1 600 pores d'une flowcell classique) ; à faible coût (80 € / 810 €) ; avec multiplexage des échantillons.

L'ADN des souches de ces différents produits laitiers a été extrait puis séquencé en utilisant le séquenceur MinION.
Afin d'étudier ces échantillons 4 index ORI ont été créés. Ces index contenaient respectivement les génomes complet de :

- 95 souches de *S. thermophilus.*
- 237 souches de *L. delbrueckii.*

- 694 souches de *Bifidobacterium*.

- 272 souches de *Enterococcus*, *Lactobacillus* et *Leuconostoc*. Notez que les nouvelles classifications des *Lactobacillaceae* et *Leuconostocaceae* [J. Zheng *et al.* 2020] n'ont pas été utilisées car la taxonomie de la base de données NCBI n'a pas encore été mise à jour.

Ces expériences ont montré que :

- les yaourts 1 et 3 étaient proche, en se basant sur leur contenu en souches *S. thermophilus* ;

- les yaourts 1 et 2 contenaient la même souche (ou des souches très proches) de *L. delbrueckii* subsp. *bulgaricus* ;

- mis à part la présence de *Bifidobacterium animalis* subsp. *lactis* dans le lait fermenté, son contenu en *S. thermophilus* et *L. delbrueckii* subsp. *bulgaricus* était très proche du yaourt 2 ;

- l'on trouvait majoritairement des souches de *S. thermophilus*, *L. delbrueckii* subsp. *indicus*, *Bifidobacterium breve*, *Lactococcus lactis*, *Lactococcus lactis* subsp. *cremoris* et *Lactococcus cremoris* dans le lait ribot.

Cependant, le nombre de lectures obtenus pour chaque produit laitier était assez faible du fait de l'utilisation de Flongle en multiplexant pour le séquençage. Les résultats d'identification d'ORI sont donc à considérer avec précaution.
De manière générale, cette expérience aura montré qu'il était possible d'identifier des souches bactériennes à partir de produits laitiers en utilisant ORI. Des améliorations pouront être apportées à l'extraction de l'ADN et au séquençage de ce dernier.

Le troisième et dernier groupe d'expériences présente des expériences liées à des collaborations extérieures à la thèse et ayant nécessité la formation d'autres personnes à l'utilisation d'ORI, ce qui a permis d'améliorer le logiciel et sa documentation. On peut séparé les expériences réalisé en 2 :

- l'identification de souches de *Streptococcus pyogenes*, un streptocoque pathogène d'intérêt clinique.

- un test d'identification en cas de contamination d'un échantillon d'une espèce par une autre espèce proche en utilisant ORI.

- un test d'identification de souches d'*E. coli* à partir d'un échantillon métagénomique de microbiote intestinal de porc.

La première expérience consistait simplement à identifier une souche pathogène de *S. pyogenes* STAB14018 dont le génome est connu à partir de lectures longues nanopore et d'un index contenant 244 génomes complets dont les souches proches ont été regroupées dans l'index. Le résultat obtenu est un groupe de trois souches proches regroupées dont fait bien partie la souche *S. pyogenes* STAB14018. Cette expérience est un premier pas permettant de montrer l'intérêt que peut avoir un logiciel tel que ORI dans le secteur médical et permet de valider le fonctionnement d'ORI pour d'autres espèces (chaque utilisateur peut librement créer un index adapté à ses besoins).

La deuxième expérience consistait à étudier les souches de *E. coli* présentes dans un échantillon métagénomique intestinal de porcelet. Le nombre de lectures étant très important et la diversité bactérienne très élevées, le logiciel Kraken 2 a été utilisé pour identifier et trier les lectures provenant uniquement de l'espèce *E. coli.* Ces lectures ont par la suite été utilisées par paquet de 4 000 lectures pour identifier les souches de *E. coli* présentes dans l'échantillon. Pour cela, un index ORI contenant 1 644 génomes complets de *E. coli* a été créé. C'est l'index le plus grand actuellement testé avec ORI (537 Mo). Lors de ce test, 8 souches de *E. coli* ont été identifiées parmi lesquelles 6 souches avaient bien été observées chez le porc [Poulin-Laprade *et al.* 2021; Z. Li *et al.* 2018]. Cette expérience a montré que ORI pouvait être combiné avec Kraken 2 pour identifier des souches d'une espèce d'intérêt dans un échantillon métagénomique.

Enfin, nous avons vérifié la spécificité d'ORI en nous appuyant sur les expériences précédentes réalisées sur *S. pyogenes* et *S. thermophilus.* Cette dernière expérience consistait à requêter un index contenant des souches de *S. thermophilus* avec des lectures longues issus d'un séquençage d'une souche de *S. pyogenes* (espèce proche) afin de vérifier qu'aucune lecture n'était assignée. L'inverse à également été réalisé (lectures de *S. thermophilus* contre un index de *S. pyogenes*). Dans les deux cas aucune lecture n'est assignée à une souche de l'autre espèce, ce qui montre que ORI est robuste en présence d'espèces proches de celles présentes dans l'index.

0.6 - Conclusions et Perspectives

Dans cette thèse, nous avons présenté une nouvelle approche d'identification de souches bactériennes à partir de lectures longues erronées issues de séquençage nanopore ainsi

qu'une implémentation de cette méthode dans un logiciel nommé ORI. Les expériences menées avec ORI montrent que ce dernier est robuste dans ses identifications de souches bactériennes et plus particulièrement lorsque les souches similaires et donc difficilement séparables sont regroupées. ORI identifie correctement les souches présentes, mais peut passer à côté de certaines souches lorsque leur couverture est trop faible.
Le temps de requête de l'index ORI reste relativement élevé et des améliorations pourraient être apportées. Les trois principales étant : (1) de descendre rapidement au niveau espèce ou genre en utilisant une autre méthode avant d'utiliser ORI, (2) d'échanger l'index d'ORI avec un index plus rapide à requêter, mais un peu plus volumineux ou (3) de modifier/améliorer l'index actuel pour le rendre plus rapide.

En ce qui concerne les lectures longues d'Oxford nanopore, leur problème majeur est le taux d'erreurs encore important actuellement. Cependant, ce taux d'erreurs diminue rapidement au fur et à mesure que la technologie (physique, chimie, biologie moléculaire) et les basecallers (bioinformatique) s'améliorent. Dans la suite de cette thèse, trois tests pourraient être intéressants à réaliser : l'utilisation de Taiyaki permettant d'entraîner spécifiquement le basecaller à reconnaître des bactéries lactiques et ainsi diminuer le taux d'erreurs des lectures de ces dernières ; l'utilisation de l'API Read Until de ONT pour sélectionner amplifier les lectures de bactéries minoritaires en écartant les séquences majoritaires et, la réutilisation des flowcells. Cela permettrait un arrêt après un temps court de séquençage (correspondant à 4 000 lectures pour une souche de qualité suffisante) suivie du lavage de la flowcell. Ce lavage permettrait de réutiliser les flowcells pour une nouvelle identification et ainsi réduire le coût des expériences. Ce dernier point rentre dans l'optimisation de la partie en laboratoire attachée à la thèse qui reste actuellement la partie la plus longue lors de l'identification de souches bactériennes (extraction ADN, préparation de la librairie, séquençage) bien qu'accélérée grandement par l'utilisation en local du MinION d'Oxford nanopore (sans passer par une plateforme externe de séquençage).

Chapter 1

Introduction

Contents

Preamble: This chapter aims is a quick introduction to the bacterial domain that focuses on how they can be classified and identified. The thesis studies bacterial identification from long read sequences of lactic acid bacteria and we present the necessary prerequisites for this study. This is done by presenting the characteristics of genomic sequencing data, as well as the main bacterial species on which we worked, the main one being the lactic acid bacteria *Streptococcus thermophilus*.

1.1 Bacterial identification

Since their discovery by Antony van Leeuwenhoek in the 17th century with the first microscopes, bacteria have been studied with increasingly precise means to better investigate and decipher their functions and purpose in the environment. Bacteriology, the study of bacteria is one of the branches at the core of microbiology. In the 19th century, Louis Pasteur played a major role in this domain by understanding the involvement of bacteria in fermentation processes and diseases (pathogenic bacteria), and by developing methods to destroy them (pasteurisation). The 19th century also saw the beginning of medical bacteriology, linked to the work of Robert Koch, for the discovery of pathogenic bacteria which are now extremely well studied. Since then, bacteriology has evolved to allow humans to use bacteria to their advantage, such as in the treatment of wastewater or the processing of food (e.g. the production of dairy products such as yoghurt and cheese). In the 20th century, a major breakthrough was the discovery of antibacterial agents such as penicillin, discovered by Ernest Duchesne and studied by Alexander Fleming.

The term bacteria refers to unicellular, microscopic and prokaryotic living organisms. This means (for the most part) that individually they are not visible to the naked eye and their cell structure does not include a nucleus. They can be separated from archaea (another group of prokaryotes) by differences in their membrane composition and ribosomal RNA. They come in many forms like spherical (cocci), rod-shaped (bacilli) or helicoidal. Usual bacterial cells have a length between 0.5 and 5 µm. However, large bacterial species are known such as *Thiomargarita namibiensis* (from ∼100-300 µm up to 750 µm wide) [Schulz *et al.* 1999] as well as other species of very small size like some Actinobacteria (< 0.1 µm) [M. W. Hahn *et al.* 2003].

Bacteria can be found in all types of environment on Earth, from the ocean depths to the soil, rock, air and even on and in animals, plants and humans. Their omnipresence allows them to play an important role in a large majority of geochemical processes, for example in the carbon and nitrogen cycles. Bacteria are necessary for the proper functioning of the entire ecosystem and the organisms living in it. A change in the set of bacteria present in an environment can have a significant impact. For example, a dysbiosis (imbalance in the biodiversity of our intestinal flora) can lead to health issues [Carding *et al.* 2015] and the presence of commensal *E. coli* in the gut microbiota helps protect against pathogenic *E. coli* [Conway and Cohen 2015]. In the same way [Bell *et al.* 2005] showed the importance of the bacterial community diversification for proper functioning of ecosystems. In the food industry, bacteria are widely used and modifications in bacterial population composition allow the production of a wide variety of foods (e.g. cheese, yoghurt, sauerkraut, vinegar...). For example the bacterial composition of sourdoughs can be associated with particular bread flavours [De Vuyst *et al.* 2002]. All these examples explain the interest of being able to identify bacteria in the most precise possible way. This raises the issue of identifying similarities and differences, that is, classifying bacteria at a fine level. As for other microorganisms, establishing a natural classification of bacteria is much harder than for animals and plants because they have relatively few features and they reproduce via fast and multiple mechanisms, both sexual and asexual. It includes simple binary division and genetic exchanges by bacterial conjugation, which involves bidirectional horizontal transfer of genetic material between bacteria close to each other.

1.1.1 Evolution and taxonomy of bacterial species

The study of the taxonomy and evolution of living organisms is called Systematics. It addresses six key points [Michener *et al.* 1970]: (1) naming rationally biological organisms, (2) describing them, (3) preserving them, (4) giving a classification of organisms and collecting data on their distributions, (5) studying their evolutionary history and (6) studying their environmental adaptation. Taxonomy is the part of systematics studying points (1) to (4). The remaining points concern phylogeny, the study of the relationship between present-day living beings and those that have already disappeared in order to reconstruct the evolution of living organisms and to understand the mechanism of appearance of new species.

Taxonomy clusters organisms into taxa on the base of shared characteristics and these groups are given a taxonomic rank corresponding to their level in a tree supposed to represent the evolution of life forms since the beginning of life. The principal ranks (an example for the bacteria *Streptococcus thermophilus* is given in brackets) in descending order from the root of this tree are domain (Bacteria), kingdom (Eubacteria), phylum (Firmicutes), class (Bacilli), order (Lactobacillales), family (Streptococcaceae), genus (Streptococcus) and species (*Streptococcus thermophilus*).

The current classification of life into three major domains, proposed in 1990 by Carl Woese, contains the Eukaryota or Eukarya and two prokaryotic domains, the Bacteria and the Archaea.

The bacterial phyla are still subject to many controversies. The estimated number of bacterial phyla was about 1 300 in 2014 [Yarza *et al.* 2014] but it drops to only 127 in the Genome Taxonomy Database (GTDB, release 202, June 2021) an Australian initiative to establish a standard microbial taxonomy from genome phylogeny [Parks, Chuvochina, Waite, *et al.* 2018; Parks, Chuvochina, Chaumeil, *et al.* 2020]. So the taxonomy of bacteria is still in flux. In details, the clades below the phylum in GTDB (release 202) contains 360 classes, 1 163 orders, 2 886 families, 12 037 genera and 45 555 species. The real number of bacterial species is still debated, the current estimates ranging from 10^3 to 10^{12} or even higher [Yarza *et al.* 2014; Louca *et al.* 2019; Bodor *et al.* 2020], and a large proportion of these species remains unknown.

1.1.2 Early methods and issues for the identification of bacteria

Bacteria were at first classified based on their shape (morphology), Gram stain, motility, culture conditions (growth requirements) and pathogenic potential [Schleifer 2009]. As explained in [Schleifer 2009] bacterial classification changes have been linked to the introduction of new techniques allowing more precise observations and combine the phenotype and the genotype of the studied organism [Vandamme *et al.* 1996]. Improved accessibility to genetic information due to the decreasing cost and higher throughput of sequencing has allowed a shift from the original molecular biology techniques used for bacterial recognition to computer techniques processing sequencing data directly.

Phenotypic information (bacterial features)

In bacteriology, morphology is the first phenotypic information that is useful for identification since it can be observed with the naked eye from a bacterial colony. The general appearance of the colony, the shape of its relief, its size, its smell, its colour and pigmentation as well as it's transparency and edge can be easily observed.

To study the morphology of the bacterial cell in more detail, the use of microscope is necessary. Microscopy allows in a fresh state (living bacteria) to see the motility of the bacteria. Then, one of the systematic steps for the identification of bacteria is the Gram stain, a technique developed by the bacteriologist H.C. Gram in 1884. Originally, it consisted of attaching the bacteria to the plate by a smear and then staining them with gentian violet in order to reveal the cell wall structure, mostly determine by peptidoglycans. Since then, it is possible to predict the Gram stain of bacteria using other techniques such as a specific polymerase chain reaction (PCR) called Multiplex Gram-Specific TaqMan-Based PCR (MGST-PCR) [Bispo *et al.* 2011].

Gram staining allows a rough classification of bacteria into two types, Gram + and

Gram -. Microscopy can be used to determine other morphological characteristics such as the size of the bacteria, their shape, the way they are grouped together (e.g. grouped in chains for Streptococci) or the presence of capsules or flagella. The formation of a bacterial biofilm can also be used for identification.

Another source of phenotypic information is bacterial physiology and particularly their metabolism, that is, the set of chemical reactions that take place in the cell. The classification of bacteria is based on their *trophic type*, depending of their metabolism. It takes into account three dimensions shown in Table 1.1.

Table 1.1: *Classification of organisms based on their metabolism.*

Energy source	Electron donor coupound	Carbon source	*Trophic type*
Light	Inorganic	CO2	Photolithoautotroph
		Organic compound	Photolithoheterotroph
	Organic	CO2	Photoorganoautotroph
		Organic compound	Photoorganoheterotroph
Biochemical oxidation	Inorganic	CO2	Chemolithoautotroph
		Organic compound	Chemolithoheterotroph
	Organic	CO2	Chemoorganoautotroph
		Organic compound	Chemoorganoheterotroph

In addition to the *trophic type*, it is also possible to study the capacity of a bacterium to degrade certain substrates like glucose in the presence or absence of oxygen or even the nitrate reduction. It is related to the presence or absence of certain enzymes in the bacteria, which allows their classification. Currently, Gram staining associated with tests for the presence of catalases (enzymes allowing oxidation-reduction reactions) and oxidases (enzymes allowing the degradation of hydrogen peroxide into water and oxygen) allow the classification of bacteria into a bacterial family. By performing other tests, often depending on the family, it is possible to go further down the classification. Nowadays, it is possible to use analytical profile index (API) detection systems that quickly perform miniaturised biochemical tests in each well of a gallery to refine the identification of bacteria. The optimal growth medium and culture conditions (e.g. temperature, hygrometry) of a bacterium depend on its trophic type and its ability to degrade certain compounds and not others. It can thus be used for identification purposes.

Finally, some methods of identification in the medical domain are based on serology or resistance to certain antibiotics.

The identification of bacteria based on their phenotypes suffers from a serious drawback: the vast majority of bacteria are not cultivable or have not been described before [Bodor *et al.* 2020]. Fortunately it is now possible to easily access the genetic content of these organisms. One way of identifying non-cultivable species is therefore to go through their genotype.

Genotypic information

The genotype represents the genetic information of an organism, inherited from its parents. The genetic information, together with the environment and epigenetic modifications, explains the phenotype of a living being.

It is present in the genome of living organisms in the form of a deoxyribonucleic acid (DNA) polymer composed of four different types of nucleotides: adenine (A), thymine (T), cytosine (C) and guanine (G). A DNA strand is an oriented chain from a 5' end to a 3'. The DNA of living organisms is double stranded, formed by two antiparallel and complementary strands (a 5'-3' sense strand opposite a 3'-5' antisense strand) forming a double helix. The nucleotides of the two DNA strands form bonds, so A will pair with T and C will pair with G.

The size of genomes varies from species to species, with generally a few megabases (Mb) for bacteria (e.g. 4.6 Mb for *Escherichia coli* [Blattner *et al.* 1997]).

A genome is composed of one or more chromosomes depending on the species. In bacteria, there is often a single circular chromosome, but some may have multiple and/or linear chromosomes. Chromosomes contain genes that code for ribonucleic acid (RNA) molecules. Some of these RNAs can be translated into proteins with specific biological functions, while others play a regulatory role in the functioning of cell metabolism. In bacterial cells, in addition to the chromosome, other DNA molecules can be found, called plasmids, which are capable of autonomous replication but are not essential for the survival of the bacteria (e.g. resistance genes, fertility factor that will allow the bacterial conjugation or completion of new metabolic pathways to use new nutrients).

Part of the bacterial genetic information is specific to a species or even a group of individuals. Other parts can spread in many bacteria without passing through a process of vertical transfer inherited from a common ancestor. This type of genetic material is acquired through three mechanisms of horizontal transfer: (1) *Bacterial conjugation*, which corresponds to the transmission of a conjugative plasmid from a donor bacterium to a recipient bacterium. Some plasmids are episomes, which means that they integrate into the chromosomal DNA and several plasmids can be present in a bacterial cell. The DNA sequence is transferred by contact between the two bacteria and integrated into the DNA of the new bacterium. (2) *Transduction*, which transfers genetic material from a donor bacterium to a recipient bacterium by means of a phage (bacterial virus). (3) *Transformation*, which consists in the integration of an exogenous DNA fragment, often from a dead bacterium, into a living bacterial cell. These three types of transfer play a major role in the diversification of bacteria and make their classification and identification more complicated.

In this thesis, we will not consider the issue of horizontal transfers during identification. We will just discard plasmid sequences from databases and assume that although some reads (pieces of DNA from a sequencing) are not relevant for identification purpose, there is sufficient genetic content for the characterization of each bacterial strain.

The first discriminant genomic information is the genome length. However it assumes that the complete genome sequence is correctly assembled, which implies a rather important sequencing and processing effort and therefore is not always realized.

Assuming that the entire or almost complete genome sequence of an organism is known, several statistics may help its identification. With the global GC rate (percentage of G or C nucleotides in the genome), bacteria can be classified from high to low GC. For example, *Streptococcus thermophilus* has a global GC rate of 38.9% wich is far lower than the 50.8% mid value for *Esherichia coli* or the 66.7% high GC rate of *Deinococcus radiodurans*. The global GC content will affect the use of codons for a given amino acid. Codons are sequences of three nucleotides in a coding gene that will define the amino acids that will be used to produce the protein to be synthesised. However, more than one codon can code for the same amino acid, this is called the codon degeneracy. Bacteria with high GC will tend to use the same codons [L.-L. Chen and C.-T. Zhang 2003]. More precise information can be obtained by looking at the percentage of GC of genes present in the genomes at position one, two and three of codons (GC1, GC2 and GC3). These GC values, especially GC2 and GC3, allow us to observe when there is a bias in codon usage. Locally, this bias can be used to find genes resulting from horizontal transfer [Lawrence and Ochman 1997]. The differential use of codons is also suitable to differentiate between thermophilic bacteria (optimum growth above 50°C) and mesophilic bacteria (optimum growth between 20 and 45°C) [Carbone, Képès, and Zinovyev 2005].

Classifying bacteria can be done by looking at their differences or by looking at their commonality. Several methods are based on whole genome analysis either by biomolecular techniques or on the genome sequence when available also called whole genome sequencing (WGS) analysis. Restriction analysis is a method that has been widely used for a cursory comparison of organisms. Restriction enzymes are proteins that cut DNA at specific positions (restriction sites), depending on the sequence. If the genome sequence is known, it is possible to predict where the restriction enzyme will cut and therefore the size of the expected pieces (restriction length). The restriction length profile for a given set of restriction enzymes is then used to compare genomes [McMaster, Tratschin, and Siegl 1981]. More refined comparison techniques exist that are based on the capacity of two DNA strands to hybridize.

DNA-DNA hybridation (DDH) is a technique for demonstrating genomes proximity (see figure 1.1). The DNA of one bacterium is marked and mixed with the unmarked DNA of another. The mixture is then incubated at high temperature so that the DNA strands dissociate and cooled gently to form a new hybrid double-stranded DNA molecule. In order to hybridize, the sequences must have a high degree of similarity. To separate the two DNAs, DNA melting is used, which consists of heating the DNA until it separates. The higher the dissociation temperature, the closer the DNAs are to each other. Nowadays DDH is computed *in silico* from sequenced genomes. The DDH percentage represents the genomes/sequences similarity and a 'gold standard' DDH value of 70% has been set for the delineation of species [Goris *et al.* n.d.; Moore *et al.* 1987].

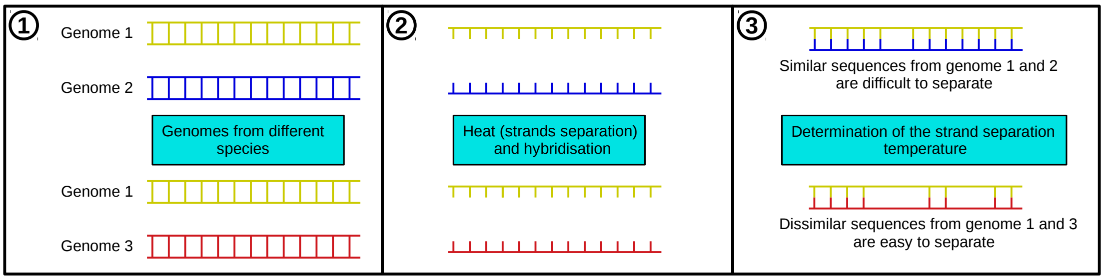


Figure 1.1: *Representation of the DNA-DNA hybridisation technique.* DNA strands from the genomes of different species (1) are dissociated by heat (2) and then reassociated (3), creating hybrids. Depending on the dissociation temperature of the two DNA strands (3), it can be determined a measure of sequence similarity. A high temperature means that the sequences are close, whereas a low temperature means that the sequences are different.

Today, with the advent of high throughput sequencing technology combined with computational analyses, the studies and identification of bacteria have evolved. More systematic identification methods have emerged based on the study of parts of the genome such as conserved genes, with or without amplification by Polymerase Chain Reaction (PCR).

PCR is based, like DDH, on temperature-dependent hybridisation and separation of complementary strands of DNA. It consists of three repeating steps: denaturation (separation of the DNA double strands), hybridisation of specific sequence (primers) on DNA strands and finally elongation of the DNA (synthesis of the complementary strand). Repeating these three steps a number of times exponentially amplifies the number of sequences corresponding to the area between two primers in the genome. If the ends of sequence to be amplified are not in the genome, there will be no amplification, it is a negative PCR test. By choosing the sequences of primers carefully, it is possible to classify bacteria based on the presence of specific marker sequences, a technique known as DNA fingerprinting.

In the last few years one of the most popular identification methods used small subunit ribosomal RNA gene (16S rRNA) amplification (PCR), sequencing and analysis. Indeed, this gene codes for part of the ribosome that is present in all bacterial cells [Woese 1987]. The structure of the 16S rRNA gene consists of a highly conserved region interspersed with genus or species-specific variable regions (see figure 1.2).

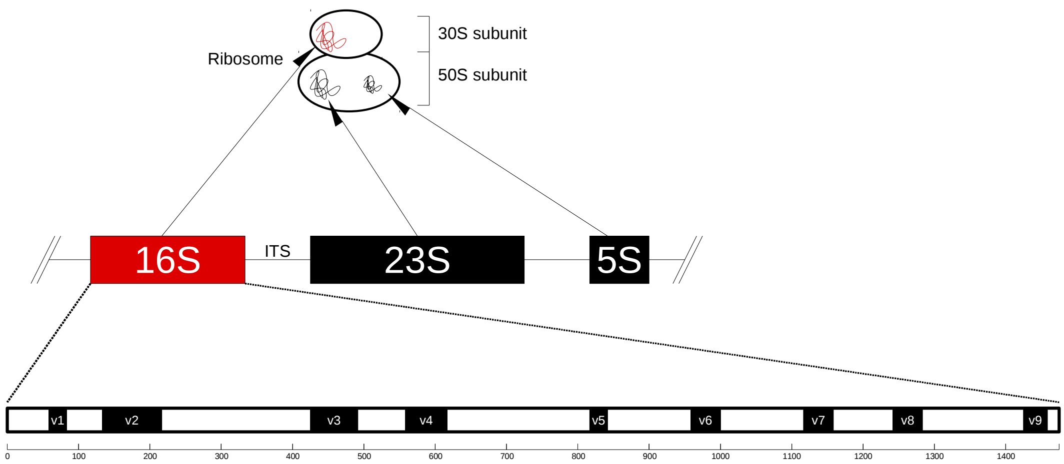


Figure 1.2: *Representation of the variable regions (V1-V9) of the ARNr 16S gene that encodes the 30S subunit of the ribosome.*

It is therefore easy to find PCR primers that bind to the conserved regions, allowing the amplification of the variable sequences of the rRNA gene [Relman 1999]. These analyses have been accelerated by the advent of high-throughput sequencing, associated with the Mothur [Schloss *et al.* 2009] or Qiime [Caporaso *et al.* 2010] pipelines which are the two reference software for studying metagenomes from short reads from 16S rRNA gene sequencing. This method is still widely used for example in the study of marine ecosystems [Muyzer *et al.* 1995] or bacteria of clinical interest [Jenkins *et al.* 2012; Cai, Archambault, and Prescott 2003]. A link was established between DDH and 16S rRNA gene sequence similarity showing that 97% sequence similarity of the 16S rRNA gene corresponded to the 70% DDH, which is the limit between bacterial species [Stackebrandt 2002]. Subsequently, this threshold has been reviewed. For example in 2014 [M. Kim *et al.* 2014] a threshold of 98.65% sequence similarity of the 16S rRNA gene was set and in 2018 this threshold was even higher (99% for the full length rRNA gene similarity) [Edgar 2018]. For example, the 16S rRNA gene of *Streptococcus thermophilus* and *Streptococcus salivarius* species are 99.74% identical. If we take this last value into account it seems that, due to the speed of evolution of the 16S rRNA gene, this technique does not always allows to go down to the species level. So, to separate species, it was shown that the use of DDH (based on the whole genome) rather than 16S rRNA seems more appropriate (from 97% similarity for 16S rRNA) [Tindall *et al.* n.d.].

Other more accurate techniques have therefore emerged such as Multilocus Sequence Typing (MLST). It aims at analysing five to seven conserved housekeeping genes that evolve faster than the 16S rRNA gene, thus allowing a better separation of bacteria. By comparing orthologous genes of the same bacterial species, an allelic profile can be created based on the nucleotide differences of these genes. In this context, orthologous genes represent genes that descent from a common ancestor, in single copy in the genome, and are common to different strains. An approach very similar to MLST is called multilocus sequence analysis (MLSA) [Chun and Rainey n.d.]. MLSA does not assigned alleles, but concatenates housekeeping gene sequences and uses this concatenated sequence to determine phylogenetic relationships between genomes. These two approaches can be used in the study of bacterial strains as in [Delorme, Legravet, *et al.* 2017] which analyzed 178 *Streptococcus thermophilus* strains with MLST or in the Streptococcal taxonomy analysis using MLSA [Thompson *et al.* 2013]. This technique has, like 16S rRNA gene sequencing, also benefited from new sequencing technology allowing WGS which, coupled with MLST analysis, allows for more accurate identification by computer analysis of the sequences [Larsen *et al.* 2012]. Finally, the success of MLST and MLSA also stems from their escape from the difficulty of sequencing and assembling certain complete genomes [Stackebrandt 2002].

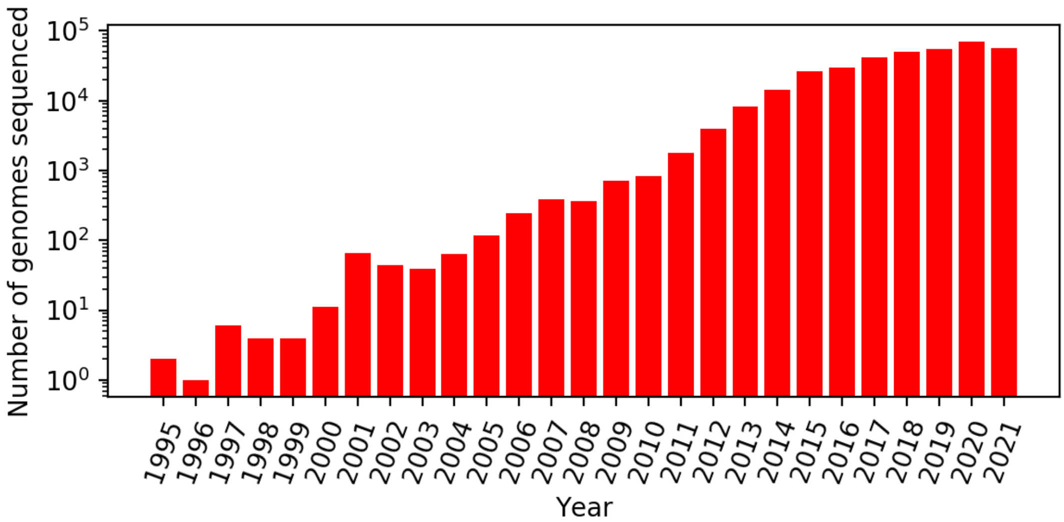


Figure 1.3: *Number of bacterial and archaeal genomes sequenced each year since 1995.* Calculated from the prokaryotes.txt GenBank file, downloaded in September 2021.

However, with the progress of sequencing techniques, the number of complete strain genomes is rapidly increasing (see figure 1.3). In the case of species with known complete genome and for an accurate recognition of strains it seems now reasonable to not limit the search to a few genes but rather make use of the whole genomic sequence. In [Chun and Rainey n.d.] the similarity indices between two genomes based on whole genome sequences are called overall genome relatedness indices (OGRI). The most commonly used index is the average nucleotide identity (ANI). It can also be referred as digital DDH or dDDH since it mimics the process of experimental DDH. The calculation goes like this: one genome is split into fragments, each of these fragments is then searched in the other genome using the BLASTn [Altschul *et al.* 1990] algorithm. A nucleotide identity value is returned as a result of this calculation and the ANI value is simply the average of these values. Furthermore, [Konstantinidis and Tiedje 2005] show that the ANI restricted to conserved genes in genomes is a robust measure of their evolutionary distance. The bacterial species delineation can be associated to an ANI value of 94-96% that corresponds to 70% DDH [Goris *et al.* n.d.].

Genome to genome distances have shown better correlation with 16S rRNA gene similarity values and ANI values than DDH values [Auch *et al.* 2010].

At last, other OGRI indices exist like the average amino acid identity (AAI), the maximum unique matches index (MUMi) or the tetranucleotide regression slope. OGRI have the advantage to produce more accurate and reproducible results than the DDH.

In addition to all the problems that can be encountered in bacterial identification, there are two separate issues that need to be considered: the problem of identifying a bacterial isolate and the more complex problem of identifying bacteria within a metagenome. The metagenome corresponds to the set of genomes present in a population in a given environment. It is obviously more difficult to identify bacteria when their genomes are mixed with those of other bacteria or even with the genomes of archaea or eukaryotes. In this thesis, the study of metagenomes is not addressed and we limit ourselves to the identification of a few strains in a mixture (e.g. fermented food).

1.1.3 Definition of a bacterial species, subspecies, clones and strain

"Which organisms are present in my sample ?" and "what do these organisms do ?" are the two major questions of metagenomics.

In order to correctly define the bacterial composition of a sample, it is often necessary to go down to a taxonomic level lower than the species level, the strain level. Whether at the level of bacterial species or strain, the definitions remain vague and much debated. The difficulty in applying the species principle to bacteria comes from the fact that the species was originally defined for eukaryotes on the basis of a certain number of criteria, including reproduction which is very different in bacteria. The concept of bacterial species is debated [Doolittle 2012; Caro-Quintero and Konstantinidis 2012] but seems to be useful. Furthermore, metagenomic studies tend to show that genetically similar bacterial groups exist. Therefore species definitions have been proposed. A species can be considered as the basic category in biological classification. The definition is still unclear and inconsistencies may arise depending on the chosen definition [Kumar *et al.* 2015]. However, one way of expressing it would be that a species corresponds to isolates having a common origin and being more closely related to each other than to any other isolate [Dijkshoorn, B. Ursing, and J. Ursing 2000]. The strain can be described as a genetic variant within a biological species. From a taxonomic point of view, a strain corresponds to the descendants of a pure bacterial culture isolation from a single initial organism. In 1995, [Tenover *et al.* 1995] defined the notion of strain as a descriptive subdivision of a species, an isolate or group of isolates distinguishable from other isolates of the same genus/species by phenotypic or genotypic characteristics. Furthermore, a strain is different from a subspecies but their definition is quite similar. A subspecies can be defined as a group of strains within species with distinct features such as populations in a particular geographical area that are genetically distinct from other populations of the same species. This notion of distinct characteristics being rather vague, distance values were put on the notion of subspecies, for example, [Meier-Kolthoff *et al.* 2014] define the 79-80% dDDH value as the threshold for delineating subspecies by using *Escherichia coli* strains.

Despite this difficulty in defining the strain as a taxonomic level, it remains important in microbial studies to be able to describe and identify bacteria at a level below the species level. For example, the most studied bacterial species, *Escherichia coli* contains pathogenic [Navarro-Garcia 2014], commensal and probiotic strains [Grozdanov *et al.* 2004]. Another example is the *Streptococcus thermophilus* strain JIM8232, which will produce a yellow pigment that is not usual for this bacterial species [Delorme, Bartholini, *et al.* 2011]. Thus it is possible to observe phenotypic differences between strains that are sometimes very close genetically. These differences can be linked to the addition, loss or modification of one or more genes, the presence of a strain-specific plasmid specific or even a differential genes expression between strains (e.g. *prtS* differential expression in *S. thermophilus* strains [Galia *et al.* 2016]).

The current difficulty in identifying bacterial strains stems both from the lack a precise definition of a strain and from the techniques currently used in bacterial identification. Classification is often achieved on the basis of DNA but can also be done using methods based on the study of RNA and proteins. In the context of this thesis, only the classification at the DNA level will be presented.

Since strains of the same species are genomically very similar organisms, it is hard to base identification techniques on one or more genes. For example, 16S rRNA sequencing, which cannot always allows to distinguish different species [Poretsky *et al.* 2014], becomes useless at the strain level. As for techniques such as MLST and MLSA developed especially for strain identification, these techniques allow the separation of certain strains but this separation depends on the genes used. These genes often come from the core-genome and the allelic profile may be identical between certain strains [Junjua *et al.* 2016]. The core-genome is the part of the pangenome containing the genes present in all strains. The pangenome contains the core-genome as well as genes that are not found in all strains of the species. Thus, in the case of species with a very large core genome, the variables parts of the strain genomes, which contain genes shared by at least two strains and strain-specific genes, will be little or not present at all for some strains.

It is possible to use the whole genome rather than just the genes. By doing this, we no longer compare the genomes with the evolutionary spectrum but it is possible to see differences in non-coding parts of the genomes. At this point another problem arises, which is how to treat two individuals with very few differences between their genomes but which, from the point of view of their definition, are considered to be of different strains. Could these strains be considered as clones ? A clone is a population in which all members come from the same progenitor by asexual reproduction [Dijkshoorn, B. Ursing, and J. Ursing 2000]. For their part, [F. Ørskov and I. Ørskov 1983] defines clones as bacterial cultures isolated from different independent sources but which are so similar in terms of phenotype and genotype that the only possible explanation is that they have a common origin. However, for the purposes of this thesis, clones will be considered to be bacteria that have undergone asexual reproduction and are completely identical. In the case of bacteria with only a few differences these bacteria will be considered as variants.

Although the study of bacteria started rather late due to technological limitations, it is now booming with the advent of high-throughput sequencing. In addition to pathogenic bacteria, it now includes the study of whole communities of bacteria and their role in their environment. Major projects concern the intestinal microbiota for humans or animals, or the large-scale project Tara Ocean [Karsenti *et al.* 2011] aiming at the study of the diversity, the ecology, and the global impact of microorganisms (including bacteria) in the oceans. Furthermore, the identification of bacteria at the species level is actually beginning to yield good results. However, there is still a need for the identification of bacterial strains, both at the level of large metagenomic studies and at the level of use and characterisation of single or small groups of strains. In this thesis we will focus on the identification and characterisation of bacterial strains via the use of genotype and more precisely via the use of the genomic sequence of the organisms studied.

1.2 Sequencing bacterial genomes

We now focus on the identification of bacteria in a sample based on the sequencing of its DNA. Sequencing allows access to the genomic content of bacteria and in some cases to recover their complete genome.

It is much more easy to sequence small parts of genomes (reads) than sequencing a complete genome. The most widely used approach is the shotgun sequencing approach proposed in 1979 [Staden 1979]. The first step of this technique is the fragmentation of the genome. It forms what is called a library composed of random fragments from the initial genomic DNA. There are two ways to perform shotgun sequencing: (1) whole genome sequencing where the genomes are randomly sheared into small fragments and (2) hierarchical sequencing where the genomes are first divided into larger segments and each fragment is then cut to form several libraries. In both cases, these DNA fragments are then sequenced. This raises the computational issue of retrieving the original sequence using the overlap of the sequenced fragments, this is called assembly. The greater the number of overlapping reads, the more the genome is said to be covered. The average number of times each position (nucleotide) of the genome is represented in these reads is the *sequencing depth*. Sequencing with a depth of 100X means that on average each base of the genome is sequenced 100 times. The greater the depth, the easier it is in theory to reassemble the original genome. However, the existence of numerous repeated regions in genomes can prevent the complete recovery of their sequence. Another parameter to take into account is the *coverage* of the genome. The genome coverage corresponds to the percentage of the genome represented by at least one read. It is another possible factor of limitation for a correct assembly. Sequencing depth and coverage are therefore two important parameters in the analysis of a whole genome sequencing.

1.2.1 First generation sequencing

The first generation of sequencer dates back to 1977. This sequencing method, named Sanger sequencing after its creator, is based on the chain termination technique [Sanger, Nicklen, and Coulson 1977]. It proceeds as follows: the synthesis reaction starts with a small sequence (called a primer) complementary to part of the DNA fragment to be sequenced. Elongation from the primer is carried out by a DNA polymerase (the same as for PCR). It uses deoxyribonucleotide triphosphates (dNTPs: dATP, dCTP, dGTP, dTTP) and a low concentration of one of the four dideoxyribonucleotides (ddNTPs: ddATP, ddCTP, ddGTP, ddTTP). These ddNTPs stop the elongation, which explains their "chain-terminating nucleotides" name. Four reactions will be carried out at the same time, one for each letter. For example, in the reaction with ddATP, the elongation will stop randomly at the level of a A because DNA polymerase uses ddATP in a random manner. Thus, at the end of the elongation process, fragments of different sizes are obtained, representing all possible sizes between the primer and any A position present in the sequenced DNA (same for the three other dNTPs). By migrating fragments using electrophoresis on a gel they can be ordered by length. This way, the sequence can be reconstructed (see supplementary figure S1). This technique is laborious and expensive but very accurate and robust, it allows to obtain the sequence of a DNA fragment of about 1 kb long with an accuracy of 99,999% [Shendure and Ji 2008]. An evolved version of this sequencing technique is still widely used in laboratories today for small-scale experiments or to finish regions such as long repeated regions that are difficult to sequence with other sequencing techniques [Koh *et al.* 2021]. Nowadays reactions all take place in a single tube and each ddNTP is labelled with a fluorochrome whose colour depends on the ddNTP. At the end, a capillary electrophoresis is used to separate the pieces of DNA instead of a gel. The fluorochrome is then excited with a laser and the light signal is automatically analysed.

The DNA molecule to be sequenced has to be amplified in order to have a large number of copies. It can be done by PCR or by plasmid cloning. Plasmid cloning works as follows, a plasmid is cut, via a restriction enzyme for example [Nathans and H. O. Smith 1975], and the DNA to be amplified is inserted into the plasmid by ligation [Jackson, Symons, and Berg 1972], which is inserted into bacteria. The bacteria with its plasmid reproduce by binary division. Subsequently, by extracting the plasmid insert (again via a restriction enzyme), the clones of the original sequence are recovered.

The 2000s saw the emergence of the second generation of sequencing technologies, also called next generation sequencing (NGS).

1.2.2 Next generation sequencing: short reads

The main NGS technologies are 454 sequencing, illumina sequencing, SOLiD sequencing and Ion Torrent sequencing. The specificity of these sequencing technologies is that they can sequence small fragments of about a hundred bp but with a high throughput (up to one billion reads per run) [Ardui *et al.* 2018]. The workflow of NGS sequencers is quite similar to Sanger sequencing. The PCR amplicons (clones of a fragment resulting from the amplification) are grouped into clusters using different methods. This makes it possible to amplify the signal in order to facilitate its analysis. Finally, a sequencing process by synthesis is carried out (extension from primers by a polymerase or a ligase) in a cyclic manner and there is, most of the time, an imaging-based data acquisition (one image per cycle) [Shendure and Ji 2008].

The most widely used of all these systems is the Illumina sequencing technology.

The principle is as follows (see figure 1.4): first the DNA is fragmented and denatured and adapter sequences are added to the ends of these fragments. Fragments are amplified by a bridge PCR [Adessi *et al.* 2000; Fedurco *et al.* 2006] that works as follows: A template strand is formed by attaching fragments to a flowcell via their adaptors. A complementary strand is synthesised by a polymerase. The template strand is then removed to leave only the synthesised strand attached to the flowcell. The free end of the complementary strand, which contains a second adaptor sequence, will then hybridize to another complementary oligonucleotide attached to the flowcell. This results in a new complementary strand synthesis identical to the template strand. A further denaturation step separates the two strands. By performing this step a large number of times, clusters of strands corresponding to a starting template strand are created.

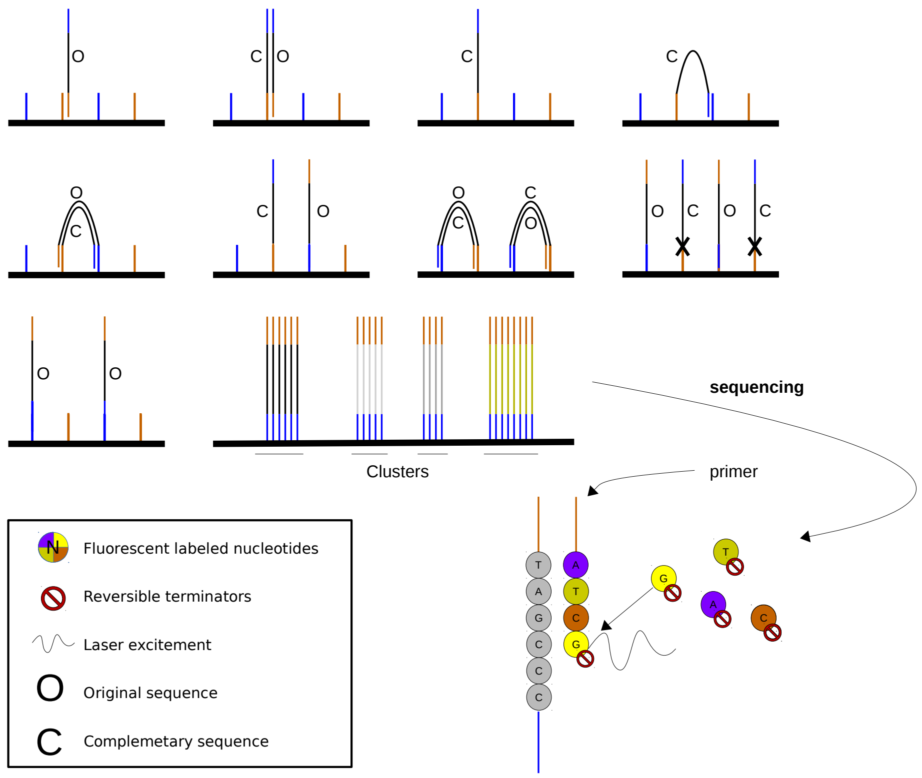


Figure 1.4: *Simplified diagram of bridge PCR followed by Illumina sequencing.* The black line represents the flowcell. The DNA fragment is first amplified by PCR which forms clusters containing many times the same molecule. Then these clusters are sequenced by wave addition of nucleotide modified with fluorochrome which are excited by laser at each sequencing step.

This is when sequencing begins. Sequencing primers are added and hybridize to all strands of all clusters via sequences complementary to the adapters. Then fluorescent labeled nucleotides with reversible terminators are incorporated. In each cycle the four types of nucleotides are added and only one nucleotide is incorporated due to the terminator nucleotide preventing elongation. The nucleotides are then excited with a laser which produces a different light for each nucleotide. At each cycle, image processing enables to infer the incorporated nucleotide [Shendure and Ji 2008].

In addition to single end sequencing, Illumina offers paired end sequencing. Instead of sequencing just one end of each fragment, both ends are sequenced (each on a different strand). Since the two ends are on a same fragment, if the fragment is too long for the sequenced ends to overlap, the fact that the two sequences are close on the genome may be used to limit the assembly combinatorics.

The range of machines offered by Illumina has grown over time and can now generate from 4 million to 20 billion sequences per run, with a maximum length of 150 to 300 bp. However, this size is still too small for some applications such as repeat region assembly, even when using paired-end sequencing. The error rate of these devices is low: in 2021 the mean error rate goes from 0.087% to 0.613% depending on the sequencer used [Stoler and Nekrutenko 2021]. For example, according to the current illumina data (2021), the MiSeq sequencer produces 90% of reads with a Qscore above 30. The error rate corresponding to this Qscore Q can be calculate with the following equation $e = 10^{\frac{-Q}{10}}$, with Q the QScore and e the estimated probability of the basecall being wrong.

Moreover, [Stoler and Nekrutenko 2021] have shown the context dependency of errors to flanking sequences notably in the case of homopolymers.

In order to overcome the small size of their reads, in 2014 Illumina created the Moleculo technology called synthetic long reads (SLRs) [Voskoboynik *et al.* 2013]. A year later, a variant allowing higher partitioning called linked-reads was proposed by 10X Genomics [G. X. Y. Zheng *et al.* 2016]. These technologies based on Illumina sequencing allow for more information than with conventional sequencing. The principle of these technologies is to cut the genome into long DNA molecules ranging from 10 kb for SLRs to 100 kb for linked reads. The long molecules are then cut into fragments for sequencing and linked to a specific barcode for all fragments of the same molecule. Their innovation is therefore based on a step upstream of sequencing that makes it possible to identify reads from the same DNA molecule [Dijk *et al.* 2018].

Knowing which reads come from the same DNA fragment facilitates bioinformatics analyses such as assembly for instance [Z. (Ma *et al.* 2019].

1.2.3 Third generation sequencing: long reads

In the end of the 2000s, the third generation of sequencer has emerged [Check Hayden 2009]. They address some of the major problems of NGS. Indeed, NGS approaches use an amplification step [Dijk *et al.* 2018] whereas the third generation allows direct sequencing of a single DNA molecule in real time. Moreover, the size of the reads has been greatly increased compared to previous generations, giving access to long-range information (at least several kb) [Pollard *et al.* 2018]. The two leading compagnies in this domain are Pacific Biosciences (PacBio) and Oxford Nanopore Technology (ONT). These technologies are still not mature, unlike NGS, their weak point being the quality of their reads.

The Pacific Biosciences sequencing technology (PacBio)

The Pacific Biosciences (PacBio) sequencer works by synthesizing and sequencing single molecules in real time (SMRT) [Eid *et al.* 2009] (see figure 1.5). PacBio uses a nanowell containing a DNA polymerase immobilized in the transparent bottom of the well that allows sequencing of a DNA template molecule [Levene *et al.* 2003]. The template is a double-stranded molecule containing a hairpin adapter at both ends which makes it circular. Sequencing occurs in each well by incorporating fluorescently labeled nucleotides which after excitation via laser produces real-time recorded light signals processed to get the sequence [Rhoads and Au 2015; Dijk *et al.* 2018].

There are several PacBio sequencing protocols with varying results in terms of error rates and lengths. The limiting factor is polymerase lifetime, so two strategies are possible: (1) Continuous Long Reads (CLR), which produces long reads that will have a high error rate or (2) Circular Consensus Sequence (CCS) which produces shorter reads with a lower error rate: thanks to hairpin adapters, the DNA sequence can be read in loop until the DNA polymerase is no longer functioning. A CCS read results from the calculated consensus of all these sequences.

In terms of features, CLRs average 10 kb in size and can approach 100 kb (Sedlazeck et al., 2018a) with an error rate of 8-15% [Rhoads and Au 2015; Dijk *et al.* 2018; Logsdon, Vollger, and Eichler 2020]. The CCS protocol makes it possible to produce High-Fidelity (HiFi) reads that have a high accuracy ($\geq$ 99%) and a length between 10 and 30 kb [Wenger *et al.* 2019; Logsdon, Vollger, and Eichler 2020]. PacBio long reads are already currently used in bacterial genome assembly and identification [Wagner *et al.* 2016].

In the context of the thesis, we have used another technology, Nanopore sequencing which has the advantage of being fast, portable and less expensive than PacBio.

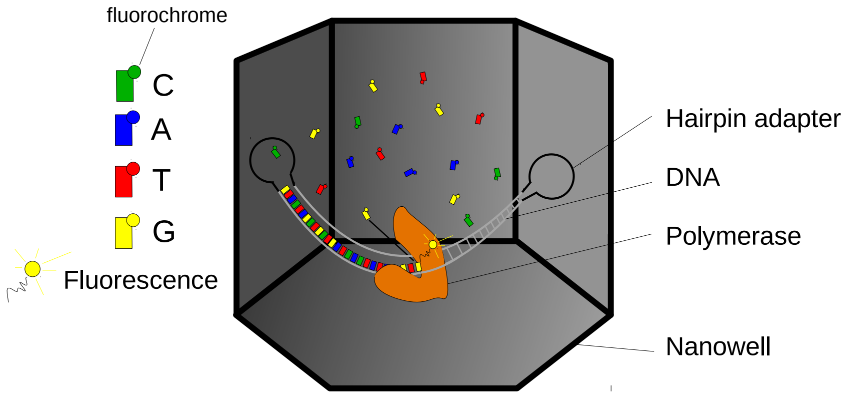


Figure 1.5: *Simplified diagram of a PacBio sequencing in a nanowell containing a DNA polymerase and fluorescently labeled nucleotides.*

The Oxford Nanopore Technology (ONT)

This long read sequencing technology is commercialised by Oxford Nanopore Technologies (ONT) since 2014. It is thus quit recent and subject to multiple changes that were a source of difficulties during the thesis since we had to multiply the experiments. Unlike PacBio or Illumina, it is not based on a synthetic sequencing approach but performs sequencing by passing DNA molecules in protein nanopores arranged on the surface of a voltage membrane [M. Jain, Fiddes, *et al.* 2015]. Depending on the nucleotides passing through the nanopore, the electric current through the membrane is modified (see figure 1.6). The raw signal data is then transformed into a DNA sequence via a basecalling step. The first methods of basecalling were based on a hidden Markov model (HMM) (e.g. Metrichor...) but were quickly replaced by connectionist machine learning approaches (e.g. Albacore, Guppy...), notably deep learning.

During sequencing, double-stranded DNA molecules are transported to the pore by a motor protein which then passes them through the pore base by base. Several types of pore have been developed, the most common being R9.4.1 and R10.3. In R9 pores, the signal is measured over about five bases [Wick, Judd, and Holt 2019] with a speed of 400-450 bases per second. In parallel with the improved chemistry, several protocols ($1D$, $2D$, $1D^2$) have been developed to increase the accuracy of the data. The $1D^2$ protocol is the most recent and uses specific adaptor sequences so that the forward and reverse strands are attached and pass successively through the nanopore allowing for improved accuracy [Lannoy, Ridder, and Risse 2017] by double reading the same sequence. In addition to the classic flowcell, ONT has developed the Flongles (Flow Cell Dongle). It is an adapter that allows direct, real-time DNA sequencing on smaller and single-use flowcells. These single-use flowcells contain fewer active pores but are less expensive than a classic R9.4 flowcell.

The average size of current ONT reads is 10-20 kb long, with an error rate ranging from 5-15% depending on the genome sequenced (%GC, low complexity regions...) and the basecaller used. In addition, a protocol to produce ultra-long reads [M. Jain, Koren, *et al.* 2018] has been developed and allows to obtain reads with an average size of 100 kb and reads up to almost 1 Mb long, particularly useful for genome assembly.

Finally, ONT offers a range of sequencers from the extremely small, cheap and portable (MinION) to the extremely powerful and high-throughput (PromethION). Its high portability (for the MinION) and affordability (see table 1.2) are advantages for ONT compared to PacBio.

The main drawback of the ONT sequences is their error rate, which made them useless at the beginning (around 40%). Even if error rate has dropped to acceptable levels, it remains high in particular sequences such as homopolymeric sequences. This is because the rate of base passage is not constant in the pore and the size of the reading head is 5 bases. Therefore, when dealing with homopolymeric sequences, the electrical signal does not change and these sequences are poorly sequenced (see figure 1.7). However, the technology is evolving very fast, with actually (2020-2021) on average a new version of the ONT basecaller Guppy every 1 to 3 months. Due to all the improvements in sequencing protocols and basecalling methods, it is currently possible to obtain reads with an error rate of less than 5% [Delahaye and Nicolas 2021; Wick, Judd, and Holt 2019].

To overcome the problem of homopolymeric sequences, a new R10 pores based on the use of two signal sensors have been developed by ONT (see figure 1.8). However, the first experiments performed with these R10 pores produced more errors than when using R9 pores. These errors are more random allowing a better correction of the reads but progress remains to be made to improve this error rate.

The ONT technology also offers a sequencing mode called Read Until. Read Until allows to analyse in real time the sequences that pass in the nanopore. It allows to filter sequences of interest to amplify them (enrichment) or on the contrary reject useless reads before the completion of sequencing. It is also possible to use this technology to globally balance the barcodes of reads during sequencing and thus have a correct sequencing

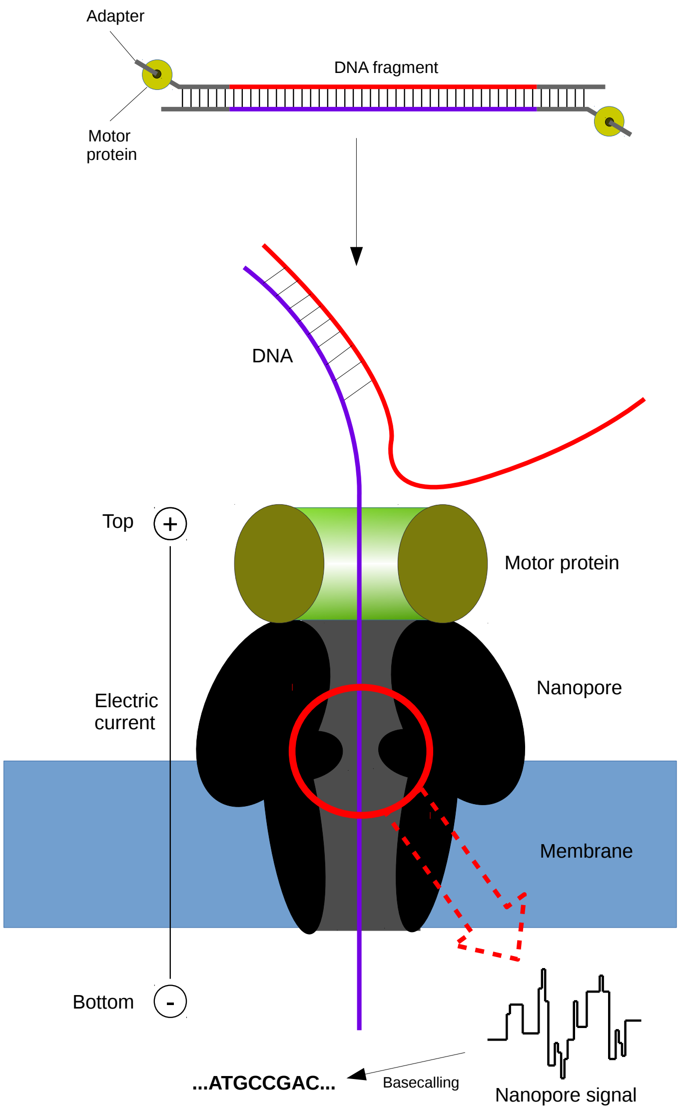


Figure 1.6: *Overview of ONT sequencing.* DNA is labelled with sequencing adapters pre-linked with a motor protein on one or both ends. The motor proteins drive each template to a nanopore that crosses a membrane. The motor protein separates the double-stranded DNA template and a single-stranded DNA passes through the nanopore. An electric current is applied to the membrane. With each passage of a nucleotide a change in the electric current is recorded. Finally, the basecalling step transforms the electric current changes into a DNA base chain.

depth for each sample when multiplexing.

Another benefit of the technology is the sale of a wash kit that allows to wash and

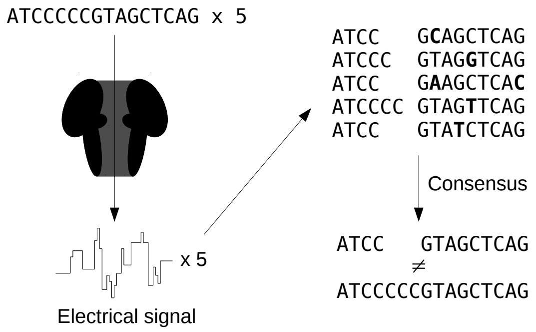


Figure 1.7: *Example of a systematic error in an homopolymeric sequence.* The error in the homopolymer being systematic it is not possible to correct it afterwards

reuse flowcells that has not been used in their entirety (where there are still active pores). Washing the flowcells allows them to be reused and thus saves money and time on experiments.

All these innovations, which are not yet fully developed and exploited, provide a basis for many future developments, notably in bacterial characterization.

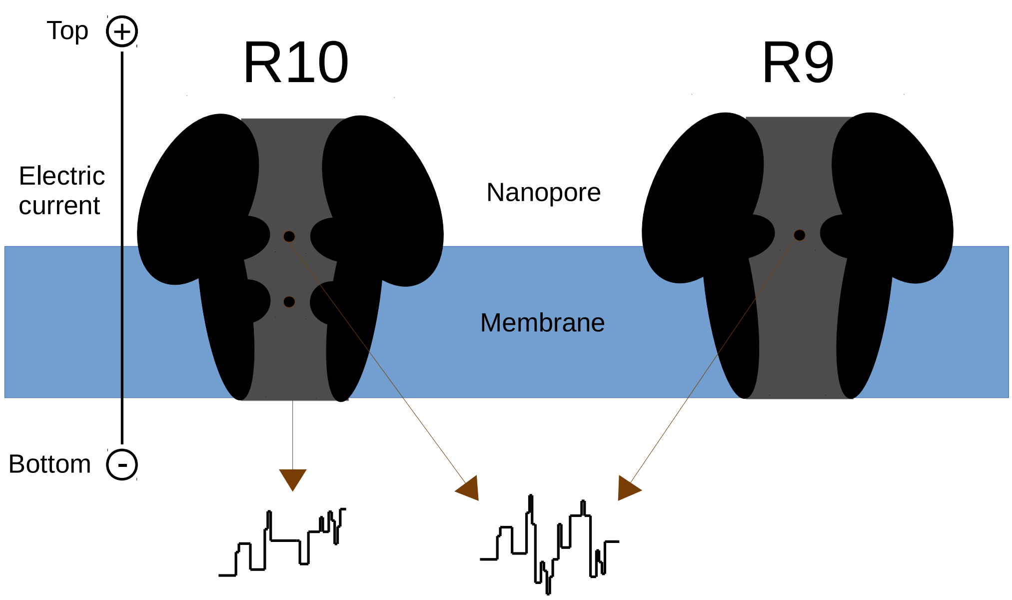


Figure 1.8: *Comparison between an R10 and R9 pore.* In an R10 pore the electrical signal is recorded at two different locations which makes it less sensitive to homopolymer errors.

General conclusion on third generation sequencing

To conclude this section, third generation sequencing allows access to much longer reads potentially containing more information, in particular structural information, which is of interest for the purpose of fine identification of bacteria. However, despite rapid improvement, their error rate is still high. For example in 2020, [Dohm *et al.* 2020] showed an average error rate of 12.72% in third generation sequencing (TGS) reads by aligning raw reads of *E. coli* to their respective genomes using four different long read aligners. Furthermore, unlike NGS data, long reads have more insertion/deletion errors than substitution errors which makes analyses at the gene level more complicated because of shifts in the reading frame.

Table 1.2: *Non-exhaustive list of third generation sequencers and their sequencing prices. Extracted from [Dijk* et al. *2018].* * means that the price of the Illumina sequencer used must be added.

Platform	Sequencer	Device Cost (k$)	Cost/Gb ($)
PacBio	PacBio RSII	700	400
	Sequel	350	85
ONT	MinION	1	24
	GridION	50 – 143	24
	PromethION	135	5
Illumina SLR	Illumina sequencer	*	12-27
10X Genomics SLR	Chromium + Illumina	125*	8-11

For the purpose of this thesis the MinION sequencer was used with R9.4 flowcells or Flongles. The thesis focuses on the use of such long reads to identify isolates or mixtures of bacterial strains. Our choice was largely due to the portability and low cost (see table 1.2) of the MinION sequencer, which allows sequencing to be carried out directly in the laboratory, allowing controlled, rapid and cheaper analysis than through sequencing platforms.

1.3 Study model: lactic acid bacteria & coliforms

This thesis is realized in the context of a collaboration between a computer science team, GenScale (bioinformatics) and a biology team, MicroBio from the STLO (milk and egg science and technology). STLO is a laboratory that studies milk and the microorganisms that are involved in its fermentation. The model of study during the thesis has been a lactic bacterium but tests on *Esherichia coli* have also been carried out.

1.3.1 *Lactobacillales*: an order whose taxonomy is still evolving

Lactobacillales are an order of Gram-positive anaerobic (or aerotolerant anaerobic) bacteria including lactic acid bacteria (LAB). LAB are able to ferment sugars into lactic acid and are therefore widely used as starters in dairy fermentations [El Soda 1993] because acidification prevents the growth of spoilage microorganisms. These bacteria are nutritionally demanding as they are auxotrophic (unable to synthesise) for a number of organic compounds and are therefore obliged to find these compounds in their growth medium.

Despite their worldwide use, *lactobacillales* remain an order whose taxonomy is still evolving and which should be more studied. Recently, great changes in the taxonomy of this order have been presented in [J. Zheng *et al.* 2020]. The genus *Lactobacillus* which contains (in March 2020) very diverse species in terms of phenotype, ecology and genotype has been broken down into 25 different genera and the family *Lactobacillaceae* was also modified to include all the genera of the family *Leuconostocaceae*. Classification errors were mainly due to the use of too few metrics, sometimes not informative enough, which lead either to associate distant bacteria or to dissociate close bacteria. The study has highlighted the value of using the whole-genome information such as in core genome phylogeny to obtain more accurate classifications.

In this thesis we were particularly interested in the *Streptococcus thermophilus* species belonging to these LAB.

1.3.2 *Streptococcus thermophilus*: a species with low genetic diversity

S. thermophilus is a food bacteria in the spherical shape of 0.7 to 1 µm and which forms chains (see figure 1.9). It's a Gram-positive bacterium of the genus *Streptococcus* and a thermophilic bacterium with an optimum growth between 35 and 42°C depending on the strain studied [Radke-Mitchell and Sandine 1986].

In this thesis, *S. thermophilus* was chosen as a model species because of its importance in the food and health industries as a dairy starter [Martinović *et al.* 2020] and potential probiotic [Uriot *et al.* 2017]. This species is closely related to other species of streptococci including deadly human pathogens (e.g. *Streptococcus pyogenes*, *Streptococcus pneumoniae* and *Streptococcus agalactiae*), which can cause diseases such as pneumonia,

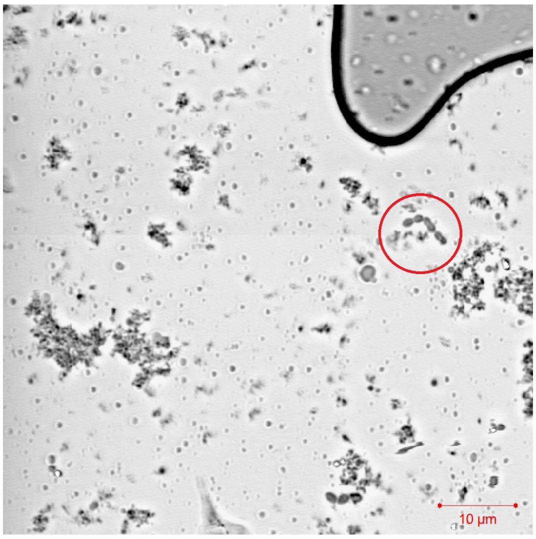


Figure 1.9: *Microscopy image showing a chain of Streptococcus thermophilus bacteria [Benyoussef 2013].*

sepsis, meningitis and pharingitis [Tettelin 2004]. It is also related to other pathogens such as *S. mutans*, which plays an important role in the development of dental decay [Mitchell 2003]. However, due to the loss of the main pathogenic genes (absents or as pseudogenes) [Bolotin *et al.* 2004], *S. thermophilus* is the only streptococcus that has the "Generally Recognised As Safe" (GRAS) status granted by the American Food and Drug Administration [FDA 2002; FDA 2012] and the "Qualified Presumption of Safety" (QPS) granted by the European Food Safety Authority (EFSA). Indeed, *S. thermophilus* is used in food processing and more than 10^{21} live cells are ingested each year by humans [Hols *et al.* 2005]. The strains of *S. thermophilus* are used in the fermentation of milk to produce fermented milk products such as yogurt or some cheeses. It is one of the only two obligatory bacteria in association with the bacterial species *Lactobacillus delbrueckii* subsp. *bulgaricus* for the manufacture of yogurt [Kiliç *et al.* 1996].

Currently (in August 2021) in the public databases, 84 complete genomes of *S. thermophilus* strains can be found. Among these 84 genomes, 31 have been sequenced and assembled in the context of this thesis. These strains come from the collection of the International Centre for Microbial Resources - Food-Associated Bacteria (CIRM-BIA) and have been sequenced with Illumina NGS technology and the Nanopore sequencing technology. Illumina and Nanopore data have been used to fully and correctly assemble their genomes using the Unicycler hybrid assembler (version 0.4.7) [Wick, Judd, Gorrie, *et al.* 2017]. These 31 complete annotated genomes have been made available on the current NCBI database.

The *Streptococcus thermophilus* genome have a median length of 1.83 Mb, a median number of genes of 1619 and a low percentage of GC content (median 39%). In addition, *S. thermophilus* is known as a species that encompasses closely related strains with low genomic diversity [Alexandraki *et al.* 2019; Hu *et al.* 2020]. This genomic proximity can be observed by MLST where some strains remain indistinguishable [Delorme, Legravet, *et al.* 2017]. However, despite this close genomic proximity, *S. thermophilus* strains can have variable phenotypic characteristics, such as their ability to adhere to gastrointestinal cells and survive gastrointestinal stress [Junjua *et al.* 2016], which means that some, but not all, can be considered probiotic. Among those strains with a different phenotype from other strains is the *S. thermophilus* strain JIM8232 [Delorme, Bartholini, *et al.* 2011] which, unlike all currently known strains, produces a yellow pigment acquired by horizontal gene transfer.

As a result, the strain sequences are quite similar, which makes the challenge of identifying them interesting in the context of this thesis. In this thesis we limit ourselves to the use of genomic information only. It is interesting in the context of the *S. thermophilus* species to show which bacterial strains can be distinguished on a genomic basis and which ones need complementary information (e.g. phenotypic) to be distinguished.

1.3.3 *Escherichia coli*: the most studied bacterial model species

During the thesis, in addition to the identification of *Streptococcus thermophilus* strains, identification of *Escherichia coli* strains was also performed. *Escherichia coli* is the most studied bacterial species. It is a Gram-negative, facultative anaerobic rod-shaped bacterium present in particular in the vertebrate intestines, generally commensal, but including pathogenic strains [Denamur *et al.* 2021].

There are currently many *E. coli* strain genomes in the database with over 2 000 fully assembled genomes of *E. coli* and more than 23 000 incomplete genomes (in September 2021).

From a strain identification point of view, there is more variability between *E. coli* strains [Rasko *et al.* 2008] than between *S. thermophilus* strains so the identification should be easier. However, identifying *E. coli* strains remains interesting because of the large number of existing genomes and their interest in health.

Chapter 2

Current state of the art in the identification of bacterial strains from sequences

Contents

Preamble: The goal of bacterial identification from a DNA sequencing dataset of a bacterial sample, is to assign the individuals present in this sample at the lowest possible taxonomic level. This can reach the genus or species level or more recently the strain level. The identification can either provide an assignment of each read or only the global composition of the sample. When studying the identification of bacteria from genomic sequences, several elements must be taken into account. The first one is the sequencing device, because depending on the technology, the DNA fragments will not have the same properties. In this chapter we present methods based on short and long reads sequencing. The second element to take into account is the quantity of information available. This leads to several strategies, using marker genes, protein sequences or complete genomes. Finally, a last point concerns the accuracy/efficiency trade-off when considering large sets of genomes. Methods can use either alignments of sequences or a compressed representation based on their composition into small words (k-mers).

2.1 Identification of bacterial strains using short reads

Identifying a bacterial strain requires the ability to compare it with already known sequences. Two standard ways to do this are by aligning sequences or by checking the presence of fixed-size fragments. Alignment-based methods will assign reads by aligning them to reference genomes (or to their genes) whereas fixed fragments size methods will assign reads by thresholding the percentage of common fragments between a genome and the read. A read can be assigned to several reference genomes. It is then necessary to either assign taxonomically each read or to identify the genomes actually present in a sample. The main bacterial identification softwares are listed at the end of each section (see tables 2.3 and 2.5).

2.1.1 Approaches based on sequence alignment with reference genomes

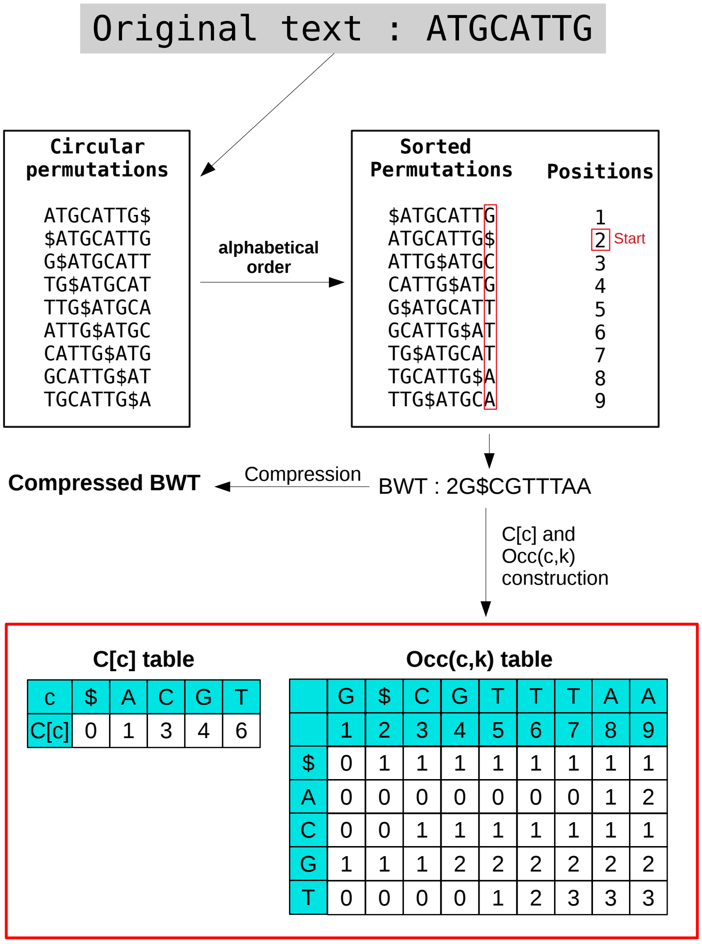
The alignment of two sequences is a way to identify the similar regions of these sequences. It consists in inserting a certain number of a special "gap" character in the sequences at some positions such as the transformed sequences have the same size and there is at least one non gap character at each position of the alignment. Then a score may be associated to a given alignment by counting the number of positions where bases are identical between the two sequences (match), different (mismatch) or one of the two sequences has a gap character (gap in the alignment corresponding to an insertion in one sequence or a deletion in the other). The goal is to get an optimal alignment of the two sequences with respect to the score.

Two algorithms based on dynamic programming have been proposed to solve this problem: the Needleman-Wunsch (NW) algorithm [Needleman and Wunsch 1970] which considers the whole sequences (global alignment) and the Smith-Waterman (SW) algorithm [T. F. Smith and Waterman 1981] which considers only parts of the sequences (local alignment). Both algorithms find an optimal solution to their respective alignment problem (global and local) in quadratic time ($O(mn)$ where m and n are the size of the two sequences).

For bacterial identification, alignments of sequenced reads to known genomic sequences are sought. If a read aligns to a genomic fragment of a unique species with a good score and on a sufficient length, then the species/strain is considered to be present. Due to the size of the genomes and the number of reads (millions/billions), the exact resolution of the alignment problem becomes too hard in practice. It is therefore necessary to design heuristics allowing reasonable execution time. The most famous one is the seed-and-extend heuristic of BLAST (Basic Local Alignment Search Tool) [Altschul *et al.* 1990] which efficiently searches for highly similar regions, called seeds, between the sequences. Then the alignment is extended from the seeds using dynamic programming which stops when the alignment score drops below a threshold.

With the current number of sequences in databases and the size of the sequence datasets, even BLAST has become insufficient, leading to the development of faster but less sensitive aligners used by metagenomic classifiers [Breitwieser, Lu, and Salzberg 2019]. An improvement of the seed-and-extend is the use of adaptive seeds. LAST [Kiełbasa *et al.* 2011] for example, looks for seeds that appear less than a certain number of times in the reference. State-of-the-art techniques speed up the search for seeds by the creation of an index for the reference and/or query sequences. The *FM-index* [Ferragina and Manzini 2000] is often used for this task in the alignment technique (see figure 2.1). This index is based on the BWT (Burrows Wheeler transformation) transform which presents similarities with the *suffix array* that allows the search of a sequence of size m in the index in time O(m).

It works as follows: first, the original sequence to index is transformed using the BWT. This BWT allows to get another representation of the sequence that promotes repetitions (see figure 2.1). This BWT corresponds to the last column when sorting all



c	$	A	C	G	T
C[c]	0	1	3	4	6

	G	$	C	G	T	T	T	A	A
	1	2	3	4	5	6	7	8	9
$	0	1	1	1	1	1	1	1	1
A	0	0	0	0	0	0	0	1	2
C	0	0	1	1	1	1	1	1	1
G	1	1	1	2	2	2	2	2	2
T	0	0	0	0	1	2	3	3	3

Figure 2.1: *Example of indexing the ATGCATTG sequence using an FM index.* The Burrows Wheeler transform is performed by sorting all permutations of the text in lexicographic order, then taking the last column. This column of size k represents the BWT of the sequence. Then, it is used to construct the arrays C[c] and the function Occ(c,k). The array C[c] associates each character c of the alphabet with the number of characters in the text that are lexicographically smaller than c. The function Occ(c,k) calculates the number of occurrences of character c in the BWT from 1 to k and creates the Occ(c,k) table. In the original paper presenting the FM-index [Ferragina and Manzini 2000], in order to save space, the BWT is successively compressed by using three different methods (a move-to-front algorithm (mtf) followed by a run-length encoding algorithm (rle) on which a prefix code (PC) is applied). The final index corresponds to the BWT compressed, the table C[c] and a function to compute Occ(c,k) in O(1).

permutations of the original text in a lexicographic order. The position of the original sequence is also kept so that it can be found again.

Once we have the BWT, it can be used to the create the FM-index corresponding to the array $C[c]$ and the function $Occ(c, k)$, where c is a character of the alphabet (A,C,G,T in the case of a nucleic sequence) and k is the position in the BWT. The array C[c] associates each character c of the alphabet with the number of characters lexicographically smaller than c in the text. The function $Occ(c, k)$ computes the $Occ(c, k)$ table containing the number of characters c at the position ranging from 1 to k in the BWT. It has been shown by [Ferragina and Manzini 2000] that the computation of this $Occ(c, k)$ table can be done in O(1). It is then possible to use lossless data compression techniques on the BWT to save space. In the original paper [Ferragina and Manzini 2000] successively compressed by three different techniques. The first one is a move-to-front algorithm (*mtf*) on the BWT wich consists in starting from the indexes of the characters c of the alphabet in alphabetical order and changing the index value of these characters by putting the current character in first position. For example, if we take the sequence $CTTTTTT$ with the alphabet $\{A, C, G, T\}$ corresponding to $\{0, 1, 2, 3\}$ indices, we will have a 1 (C) then C is changed to 0 so the alphabet becomes $\{C, A, G, T\}$. Then we have a 3 (T), T becomes 0 and then five 0's. This example is presented in figure 2.2. The final sequence is 1300000.

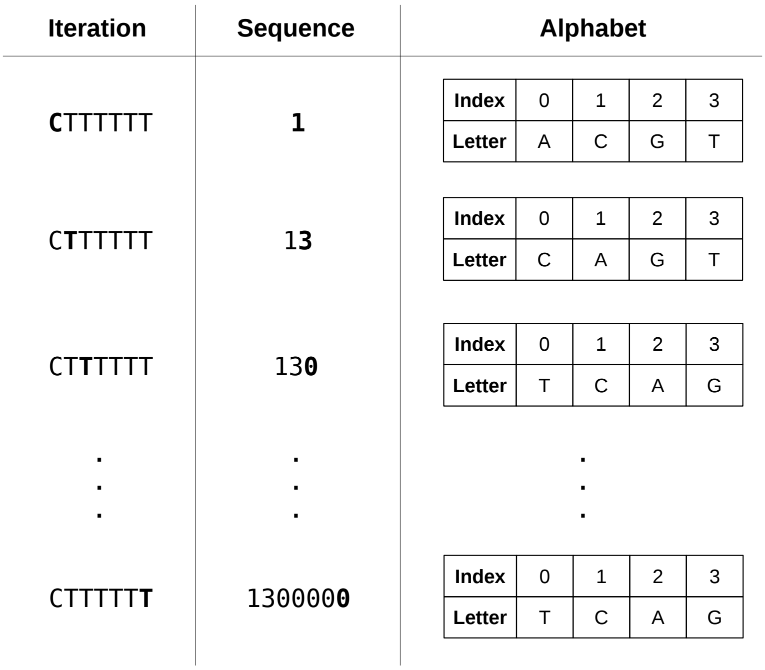


Figure 2.2: *Example of move-to-front algorithm applied to the $CTTTTTT$ sequence on the alphabet $\{A, C, G, T\}$.*

The structural properties of the BWT imply that the $mtf(BWT)$ will be dominated by small numbers. On this *mft* encoding a run-length encoding (*rle*) is applied to compress the 0 range created by the *mtf*. The *rle* principle is to encode the number of characters followed by the character in question. For example 00000111111 will be written 5061. Finally, this encode text is compressed by the use of a variable-length prefix code (PC) which consists of modifying the way the numbers are encoded in memory to use as few bits as possible. The final FM-index correspond to the compress $PC(rle(mtf(BWT)))$ sequence, the C[c] table and the Occ(c,k) function.

The FM-index provides a lossless compressed representation of genomic sequences with a complexity in time and storage space that is sublinear to the size of the input data. It allows the rapid counting of the number of occurrences of a pattern of size m in O(m) and the location of these occurrences can be achieved in a sublinear querytime depending on the size m and the number of occurrences of the pattern. A more recent version further speeds up the search for string positions in this index [Ferragina and Manzini 2005].

In the case of the seed-and-extend heuristic, the FM-index is used to quickly find the positions of the seeds. Techniques based on the use of this index are used by aligners such as BWA-MEM [H. Li and Durbin 2009] and Bowtie2 [Langmead and Salzberg 2012] which are adapted to align a large number of relatively short sequences on a single reference. BWA-MEM simply uses the BWA structure and takes advantage of its similarity to the suffix table, whereas Bowtie2 use a modified FM-index structure. This modified FM-index stores more information than a standard FM-index to allow a quicker localisation of a pattern occurrences, in return the size of the index is therefore larger. However, as both of these aligners are used to align very large numbers of reads against a single medium-sized reference it is normal for their trade-off to be in favour of query speed rather than data compression. In addition, to further accelerate the speed of alignment, these two programs provide the possibility to parallelize the alignment of the reads on the genome and thus to align several reads at the same time. A comparison of the index size of both software is available in the table 2.1.

Other alignment software that work on protein sequences can include Hidden Markov chains for search of proteins motifs or a preprocessing step to reduce the alphabet size. Table 2.2 lists different short read alignment software and their strategy.

Table 2.1: *Comparison of BWA and Bowtie2 index size using Streptococcus thermophilus and human genome.*

Software	Genomes	Genomes size	Index	Index size
BWA-MEM	*S. thermophilus* JIM8232	2 Mb	BWT	3.3 Mb
	Human	3.1 Gb		5.3 Gb
Bowtie2	*S. thermophilus* JIM8232	2 Mb	FM-index	11 Mb
	Human	3.1 Gb		7.1 Gb

The choice of the compromise between speed and space used (storage and memory) depends on the desired alignment, so software specialising in the alignment of two whole genomes will not use the same strategy than software for aligning reads on genomes. FM-index is the most widely used strategy technique for aligning reads to genomes but the BWT compression steps are sometimes not done and strategies are added to speed up the pattern search at the expense of the space used to store the sequences. In contrast MUMmer4 [Marçais, Delcher, *et al.* 2018] a software allowing the alignment of two genomes, uses suffix trees to index the two genomes as these generally do not require specifically to be compressed.

Table 2.2: *Non-exhaustive table of short read alignment software and their main search strategy.*

Software	Search strategy	Reference
BLAST	First implementation of the seed-and-extend heuristic	[Altschul *et al.* 1990]
BWA-MEM	Seed-and-extend with BWT allowing mismatch and gap in the seed	[H. Li and Durbin 2009]
LAST	Seed-and-extend with use of adaptive seeds	[Kiełbasa *et al.* 2011]
Bowtie2	Seed-and-extend with FM-index	[Langmead and Salzberg 2012]
HMMER	Use of hidden Markov chain for protein alignment	[Eddy 1998]
DIAMOND	Seed-and-extend using reduced alphabet and spaced seed for protein and translated DNA	[Buchfink, Xie, and Huson 2015]
MUMmer4	Complete genomes alignment using suffix trees	[Marçais, Delcher, *et al.* 2018]

We present in table 2.2 a list of important alignment software used in bacterial identification software. The early read classifiers used the presence of marker genes to identify bacteria like for example MethaPhlAn [Segata *et al.* 2012]. The use of genes allows to be faster than with whole-genome alignments. These approaches choose single copy gene sets present in all organisms below a certain level of taxonomy. These genes are searched, at the protein or nucleic level. However, it does not work well for strains due to their proximity since they often differ by presence/absence and duplication of a genes.

Even among genes that are common between strains, these are often well conserved and do not necessarily show variation. More generally, differences between strains of the same species will often be related to differences in the genes under selection, which depends on the environment of the strain. For example the strains of the *Pseudomonas aeruginosa* species diverge mostly because of their accessory genomes linked in particular to transposable elements [Subedi *et al.* 2018]. Another example can be the evolution of the genome composition of *Lactobacillus salivarius* strains to adapt to their specific host [J.-Y. Lee *et al.* 2017] known as the host adaptation mechanism that is present in some bacterial species [Toft and Andersson 2010]. Other approaches implement techniques that use unique genomic fragments (often k-mers) specific to a taxon rather than a set of unique genes (see GOTTCHA [Freitas *et al.* 2015]) or even techniques that use species-specific marker genes variants (see the strain level version of MethaPlAn called StrainPhlAn [Truong *et al.* 2017]).

Reads can also be assigned to genomes by alignment on the complete genome. Many software make local alignment of reads on complete genomes before assigning them to a taxonomic rank. However, most of them do not reach the strain level either by choice or because of the lack of accuracy of aligners. In addition, most identification software use the lowest common ancestor (LCA) in case of ambiguity in the alignment of a read: if a read aligns to different genomes with a good score, it will be assigned to the LCA of those genomes. A drawback of this strategy is that identification tends to stop before the species level, especially as the read is short and therefore contains less information because it will align itself more easily with distant genomes. The DUDes software [Piro, Lindner, and Renard 2016] uses another notion called the deepest uncommon descendant (DUD) which corresponds to a top-down approach going down the taxonomy. In addition to finding the DUD (which corresponds to the LCA), DUDes uses alignment score to solve certain ambiguities in the taxonomic rank. However it remains insufficient to go down below the species level.

Among the identification software based on sequence alignment, Centrifuge [D. Kim *et al.* 2016] is a fast metagenomic classifier descending to the strain level. It uses an FM index to align reads on full genomes from databases. It also compresses shared sequences from nearby genomes using a genome-genome aligner allowing the size of the index data structure to be further reduced. However this compression rate of course depends on the size of the species core genome. For example, in [D. Kim *et al.* 2016] the compression of 131 *E. coli* genomes in one sequence allows a 16% gain in space with respect to the space taken by the 131 initial sequences. Another example is the compression of the index containing about 4 300 bacterial and archaeal genomes from 6.9 GB to 4.2 GB. However, this compression limits Centrifuge to the species level classification while without this compression the results are at the strain level. As with other sequence alignment based identification software, Centrifuge faces the problem of reads assigned to multiple genomes. To resolve these ambiguities Centrifuge then uses a modified version of the LCA approach. This approach simply leaves a certain amount of ambiguity where it exists. By default it is possible for a sequence to be associated with five taxonomies (modifiable threshold). If there are more than five taxonomies associated with a read, the number of taxonomies is reduced by grouping this taxonomies at higher taxonomies (LCA approach) until there are five taxonomies associated with the read.

Another approach using an FM-index table similar to that used in Centrifuge is Kaiju [Menzel, Ng, and Krogh 2016] which works with a protein database.

Finally, there are approaches allowing strains identification that assign reads to genomes by using sequence quality and alignment quality associated with statistical models such as Bayesian statistical model (Pathoscope [Francis *et al.* 2013]). Other software such as Sigma [Ahn, Chai, and Pan 2015] use the alignment of reads to genomes to define a probabilistic model to test for the presence of genomes in the sample.

Following taxonomic assignment some software (e.g. Centrifuge, Pathoscope) quantify the strains present in the sample, most of the time, using an Expectation-Maximisation (EM) algorithm. As the aim of this thesis is simply to identify the strains present in a sample we will not discuss this genome quantification step further.

Table 2.3: *List of important short read identification software based on the use of sequence alignment.*

<table>
<tr><th></th><th>Identification software</th><th colspan="2">Alignment software</th><th>Taxonomic assignation</th><th>Reference</th></tr>
<tr><td rowspan="5">Alignment on genome fragments</td><td>MethaPhlAn</td><td colspan="2">Bowtie2</td><td rowspan="4">Clade-specific unique marker genes</td><td>[Segata *et al.* 2012]</td></tr>
<tr><td>StrainPhlAn</td><td colspan="2">Bowtie2</td><td>[Truong *et al.* 2017]</td></tr>
<tr><td>mOTU</td><td colspan="2">HMMER</td><td>[Sunagawa *et al.* 2013]</td></tr>
<tr><td>Phylosift</td><td colspan="2">HMMER</td><td>[Darling *et al.* 2014]</td></tr>
<tr><td>GOTTCHA</td><td colspan="2">BWA-MEM</td><td>Clade-specific unique k-mers</td><td>[Freitas *et al.* 2015]</td></tr>
<tr><td rowspan="7">Local alignment on complete genomes</td><td>MEGAN</td><td rowspan="3">Any short read aligner</td><td rowspan="2">default: BLAST/LAST</td><td rowspan="2">LCA</td><td>[Huson, Beier, *et al.* 2016]</td></tr>
<tr><td>Taxator-tk</td><td>[Dröge, Gregor, and McHardy 2015]</td></tr>
<tr><td>DUDes</td><td>None</td><td>DUD</td><td>[Piro, Lindner, and Renard 2016]</td></tr>
<tr><td>Kaiju</td><td colspan="2" rowspan="2">Own aligner based on FM-index</td><td rowspan="2">LCA</td><td>[Menzel, Ng, and Krogh 2016]</td></tr>
<tr><td>Centrifuge</td><td>[D. Kim *et al.* 2016]</td></tr>
<tr><td>Pathoscope</td><td colspan="2">BLAST+GNUMAP (probabilistic NW)</td><td>Bayesian statistical framework</td><td>[Francis *et al.* 2013]</td></tr>
<tr><td>Sigma</td><td colspan="2">Bowtie2</td><td>Statistical framework</td><td>[Ahn, Chai, and Pan 2015]</td></tr>
</table>

2.1.2 Approaches based on genome fragment indexing

In order to escape the intrinsic quadratic complexity of alignment, most recent methods are based on the decomposition of genomes into fragments which are then integrated into an index. This ensures a good compression of data and a fast search for the presence of fragments common to genomes and reads.

Genomes indexing and read assignation

An index is a structure whose purpose is to order and sort data using the less possible space in order to be able to find them more quickly. In genomics, an index can be used for example to search for a specific gene in thousands of metagenomic samples. In our case the aim is to search for a read representing a small part of a genome in a more or less large list of genomes. To do this, the genomes are fragmented into small fragments (e.g. k-mers) which are then entered into the index; this is called *genome indexing*. In this case, the index has two main roles: (1) the first is to save the presence information and possibly the counting of genomes fragments by using the least amount of space possible. (2) the second role of the index is to allow when a read is given to quickly find in which genome(s) this read is present; this is what we call *querying the index*.
This query part allows a read to be assigned to its reference genome(s) and thus replaces the read alignment step.

Indexing based on k-mers

The main approach for genome indexing is based on k-mers. A k-mer is a substring of length k of a string. In our case, it is a fragment of DNA (read, genome...) or RNA, and there are therefore 4^k possible k-mers.
Unlike alignment-based approaches which will search for approximate sequence matches, k-mer based approaches compare sequences only on the basis of the set of common fragments. The contribution of each possible k-mer to the sequences similarities may be based simply on its presence/absence or on a count of its occurrences.

This k-mer set representation of sequences is more compact due to the disappearance of redundant k-mers occurrences. However, the indexing based on the decomposition into k-mers will most of the time cause the notion of continuity between k-mers to be lost. When we search for the presence of a read R in a genome G, if R is of size k (size of the k-mers) then there will be no problem. In the case of a larger sequence R, it is split into k-mers and each k-mer is searched in G. However, there is no guarantee that even if all the k-mers of R are present in G, the unique continuous sequence R would also be present (see figure 2.3). However, the assumption used in most methods is that k is large enough that this does not (or very rarely) happen.

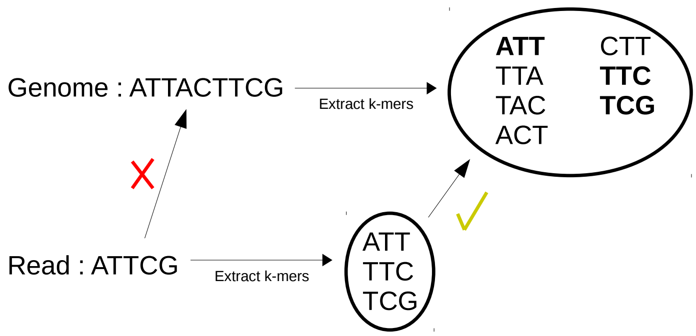


Figure 2.3: *Example of read whose 3-mers are present in the genome but which is not actually present in this genome.* This error is related to the loss of continuity in the set of k-mers. However, in reality, the size chosen for the the k-mers is large enough that this type of false positive rarely occurs.

Many of the methods presented below were originally developed for other purposes than bacterial strain identification (e.g. metagenomics, RNA-seq...). In the context of this thesis we will consider the study of bacterial strain genomes. Thus these bacterial genomes are the basic data to be indexed and the sequences searched in these genomes are the long reads.

A set of k-mers can be simply represented by a bit vector that stores the presence/absence of a k-mer in the data. To save the k-mers continuity it is possible to use a more advanced data structure called a *De Bruijn* graph [Chikhi, Holub, and Medvedev 2021]. We recall that a graph G is a pair (V, E) where V corresponds to the vertices (also called nodes) and E to the edges of the graph (see A in figure 2.4). We will now give a short review of some of the graph structures and types of graphs that will be used later in this thesis.

Graph structures:

- A graph can be traversed via *paths* which are represented by a sequence of edges connecting nodes together (see numbered nodes in figure 2.4).
- A graph is *connected* if each of its vertices can be connected to another by a path, otherwise the graph is *disconnected*.

- A subgraph is a subset V' and E' of a graph $G = (V, E)$.
- A *connected component* is a maximal connected subgraph of a graph.
- A graph is *directed* if edges are oriented (see B in figure 2.4).
- A *cycle* (in undirected graph) or *circuit* (in directed graph) is a path whose two vertices at the ends are identical.

Specific type of graphs:

- A *complete graph* is a graph where all the nodes are connected between them (see C in figure 2.4).
- A *clique* of a graph is a complete subgraph of that graph. The maximal cliques of a graph are the cliques that are not contained in any other clique.
- A *tree* is an acyclic connected graph in which two nodes are thus connected only by one path (see D in figure 2.4).
- A *subtree* is a subgraph of a tree.

The most widely used graph for storing genomic information based on k-mers is the *De Bruijn graph*. A *De Bruijn* graph is a specific type of directed graph whose vertices are k-mer and edges link k-mers overlapping on k-1 positions (see E and F in figure 2.4). Recently [Bowe *et al.* 2012] showed that it is possible to represent a De Bruijn graph in a compact way using a BWT representation of the graph that allow indexing and compression called the BOSS structure (BWT-based De Bruijn Graphs) [Boucher *et al.* 2015].

In their review, [Marchet, Boucher, *et al.* 2020] divide the other data structures that allow to index a genome into two categories. The first one is dedicated to membership queries: they store the presence/absence of k-mers. A typical structure in this category is the Bloom filter (BF). This probabilistic data structure has been invented by Burton Howard Bloom in 1970 (see figure 2.5) and we give some details on it since it will be used in our approach. A BF can be seen as an array of bits (initially set to 0) of fixed size n. In addition to the array the filter contains a collection of m hash functions $h_1 ... h_m$. A hash function is a particular function which, from an input data, computes a digital fingerprint used to quickly identify this initial data. While using with Bloom filters, the hash functions compute from a k-mer, a number representing its position in an array.

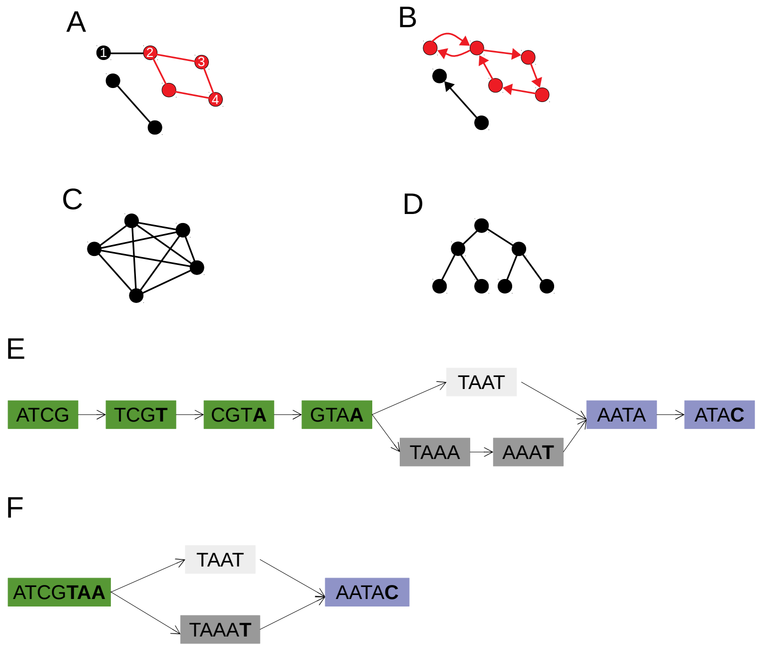


Figure 2.4: *Different representations of graphs.* In A we have a disconnected undirected graph with in red a cycle and an example of a path from node number 1 to 4. In B we have a disconnected directed graph with a circuit in red (there are actually three circuits in this graph). In C there is an example of a complete graph. In D there is an example of a tree. Finally graph E represents a De Bruijn graph f the 4-mers set ATCG, TCGT, CGTA, GTAA, TAAT, TAAA, AAAT, AATA, ATAC and graph F represents the corresponding compacted De Bruijn graph.

To insert a k-mer x into the array, all positions $h_1(x)...h_m(x)$ are set to 1 in the array. Then, to find out if a k-mer y is present in the BF, we check if positions $h_1(y)...h_m(y)$ in the array correspond to a value 1. An advantage of this data structure is that it requires fixed space to store the k-mers (n is fixed and defined beforehand). However, it can only answer with certainty to queries about the absence of an element (no false negative) or with a certain probability for the presence of the element (there can be false positives). If we assume that the probability of each bit being equal to 1 is independent of other bits, then the probability of a false positive can be estimated by using the following equation:

$$P_{false\_positive} \approx (1 - e^{\frac{-lm}{n}})^m \quad \text{(A)}$$

where l is the number of inserted elements.
Even if the independence assumption is false, the estimate of the false positive rate is still close to reality if $\frac{n}{l}$ is high and, due to its simplicity, this equation is therefore often used to approximate the false positive rate of a Bloom filter.

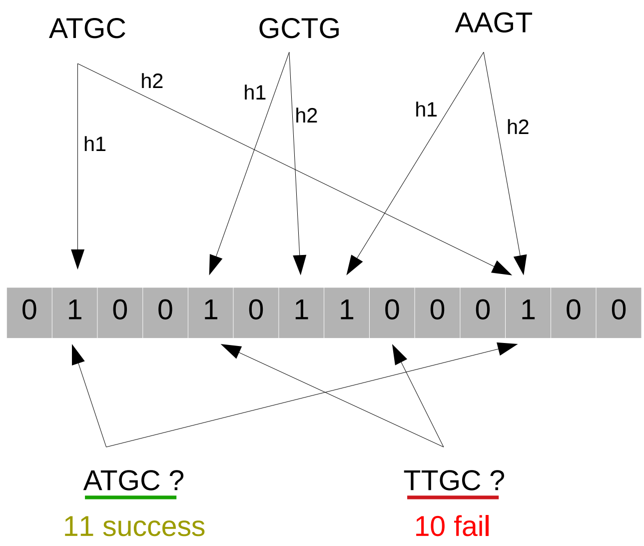


Figure 2.5: *Bloom filter of size 14 with 2 hash functions* $h1$ *and* $h2$*.* Three 4-mers (ATGC, GCTG, AAGT) are inserted in the filter. The 4-mers ATGC and TTGC are searched in the filter. The probability of having a false positive using this filter and equation A is about 12,15%. But the $\frac{n}{l} = 4.7$ ratio is a bit low, so the false positive probability is therefore probably underestimated.

The second category of index uses associative data structures, which associate information with k-mers. For instance, the hash table used in the Othello hashing method [Yu *et al.* 2018] is a MPHF (Minimal Perfect Hash Function) that allows to index n keys on n consecutive integers without any collision. This permits to be more compact than a conventional hash table while maintaining a constant time to search for the origin genomes of a k-mer. The counting quotient filter (CQF) [Pandey, Bender, *et al.* 2017] has strategy comparable to Bloom filters but with a different hashing strategy that allows to perform element counts in addition to presence/absence queries. CQF cuts the k-mer in two parts and uses the hash of the first part to find the position in the array and store the second part associated with its count.

Most of the data structures that enable genome indexing can be compressed to save space via various compression techniques. BFs are amenable to some of these compression techniques because of their bit representation with long ranges of 1's or 0's. In their review, [Marchet, Boucher, *et al.* 2020] separate k-mers set compression techniques into three main types: the bit-vector compression, delta-based compression and hybrid techniques. In bit vector compression we find methods such as RRR [R. Raman, V. Raman, and Satti 2007] and Elias-Fano [Elias 1974; Ottaviano and Venturini 2014] whose overall principle is to find the ranges of 0s and encode them in the most efficient way possible. Delta-based compression techniques are based on the principle that if two sets of k-mers are close it is more cost effective to store only one set and a representation of the differences of this set with the other set. Finally the hybrid techniques are based on the separation of vectors of bit in buckets. The compression technique used for each bucket depends on characteristics of the vectors contained in the buckets. However, these data compression techniques are complex and quite distant from the strain identification goal of this thesis. Therefore, they will not be discussed in detail in this thesis. We will simply note that basic data structures such as BF can be compressed to optimize space usage.

All these methods represent a genome as a set of k-mers. A more difficult problem is to index *sets* of genomes. For this purpose, it is possible to assign distinct color to each genome. There are different data structures based on aggregative methods [Marchet, Boucher, *et al.* 2020] that allow to index sets of genomes. These methods can be classified into two categories, *color-aggregative methods* and *k-mer aggregative methods* (see table 2.4 at the end of this section).

Color-aggregative methods

Color-aggregative methods first index the union set of all k-mers then associate original genome(s) information with each k-mers (see figure 2.6). The first color-aggregative method was a *colored DBG* proposed by [Iqbal *et al.* 2012]. The principle is quite simple, a colored DBG represents multiple genomes in a single DBG by coloring the nodes of the graph with the colors of the genomes from which the k-mer comes.
Later, more compact structures were developed such as the Bloom Filter Trie (BFT) [Holley, Wittler, and Stoye 2016] which stores the k-mers and their corresponding colors in a tree-like structure. Going down this tree, we find the prefixes of the k-mer. The leaves contain the suffixes corresponding to the prefixes of the previous nodes and the colors of the genomes containing the k-mer. Another structure allowing to save space compared to colored DBG is implemented in the Vari software [Muggli, Bowe, *et al.* 2017]. Vari compresses the DBG by using a BOSS structure associated with a compressed color

matrix.
Improvements were made over times but in general, the basic strategy of all the methods is as follows: the set of k-mers from all genomes is indexed using either a software-specific representation (e.g. hash table) or a DBG. Then a correspondence is established between this indexed k-mers and the colors (genomes). The association of colors to k-mers is done using one of the three following structures:

- A color matrix (e.g. Vari, Vari-Merge, BLight) where a k-mer corresponds to a row of the matrix that encodes the colors (genomes) corresponding.
- A color classes matrix (e.g. BFT, Rainbowfish, Mantis, Mantis+MST) where the k-mers which have the same origins (same colors) point to the same row of the matrix. This representation reduces space used compare to the use of color matrix.
- Several color matrices, this strategy is used in SeqOthello [Yu *et al.* 2018] and depends on the k-mer occurrence in the reference genomes.

K-mer aggregative methods

Unlike color-aggregative methods, k-mers aggregative methods first index all k-mers sets separately then aggregate them (see figure 2.6). In this category, we find many methods having the Bloom filters as basic structure to store data. Indeed, the Bloom filter is very efficient in terms of space to store DNA sequences and the false positive rate of these filters is reduced by the search of the many k-mers of a query with a DNA sequence. [Solomon and Kingsford 2016] has proven that even with a very high false positive rate of the filters (e.g. 50%) there is still no performance degradation without errors on a sequence if at least 50% of its k-mers are found. In practice, false positives can be linked to sequencing errors in reads. In the context of bacterial strain identification from nanopore reads, which is the aim of this thesis, this error rate is quite high and therefore requires a lower false positive rate of the filters.

There are different data aggregation structures used in this category to regroup BFs. The first one is the use of matrices. For example in BIGSI [Bradley *et al.* 2019], the BFs are concatenated in order to obtain a matrix where each column corresponds to a color (genome) which allows for very fast queries. COBS [Bingmann *et al.* 2019] subsequently improved BIGSI method's in terms of time (C++ implementation) and space by using variable BF size depending on the genomes size instead of fixed size BFs. There are similar works such as BioBloom Tools [Chu, Sadeghi, *et al.* 2014; Chu, Mohamadi, *et al.*

2020] which evolved from the use of multiple hash functions in 2014 [Chu, Sadeghi, *et al.* 2014] to a multiindex Bloom Filter (miBF) structure [Chu, Mohamadi, *et al.* 2020] allowing the storage of multiple spaced seed sequences (see section 3.1.2 for further details) instead of multiple hash functions. Finally, structures containing BFs representing several data sets also exist such as in RAMBO [Gupta *et al.* 2020] (matrix form) or DREAM-YARA [Dadi *et al.* 2018]. RAMBO merges the initial BFs randomly and creates a matrix containing all combinations of the merge of the initial BFs. This matrix is queried row by row and the intersection of the results is used to find where the k-mer comes from. For DREAM-YARA, the data structure is based on the Interleaving Bloom filters (IBF). The IBF is a large BF that interleaves the bits of the BFs corresponding to the n different genomes. So, for example, the first n bits of the IBF correspond to the first bit of the n initial BFs which allows to find very quickly in which genomes a k-mer is present.

The second data aggregation structure is the search tree. In this case the initial datasets are BFs at the leaves of the tree and the internal nodes represent groups of datasets (union of the BFs) with the root node representing all the initial datasets. The first implementation of this kind of structure is the Sequence Bloom Tree (SBT) [Solomon and Kingsford 2016]. In this SBT a node in the tree represents the k-mers present in this sets of the subtree under the node. The tree structure represents a hierarchical clustering of BFs (datasets). It can be obtained by using for example the k-mers composition of the datasets. Improvements were made later, Split-Sequence Bloom Trees (SSBT) [Solomon and Kingsford 2018] and AllsomeSBT [Sun *et al.* 2017] are the first to implement the use of two filters at each node of the tree. This allows for faster queries on the one hand, and a more space-efficient index on the other. Finally, HowDeSBT [Harris and Medvedev 2020] later improved the use of space and time by mixing ideas from the previous softwares. It is currently, in August 2021, the most space-efficient tool for indexing genomic k-mers datasets, and has been chosen for this reason as the basis of our own method.

HowDeSBT works as follows: first, the set of datasets to be indexed are represented by their Bloom filters in leaves of the tree (as with SBT). HowDeSBT uses only one hash function to hash the k-mers and insert them into their BF. The tree topology is determined by clustering using the Hamming distance between BFs (see section 3.2.1). The bottom-up clustering method used in HowDeSBT is the same agglomerative hierarchical clustering as in AllSomeSBT based on [Hoon *et al.* 2004]. Grouping the BFs close

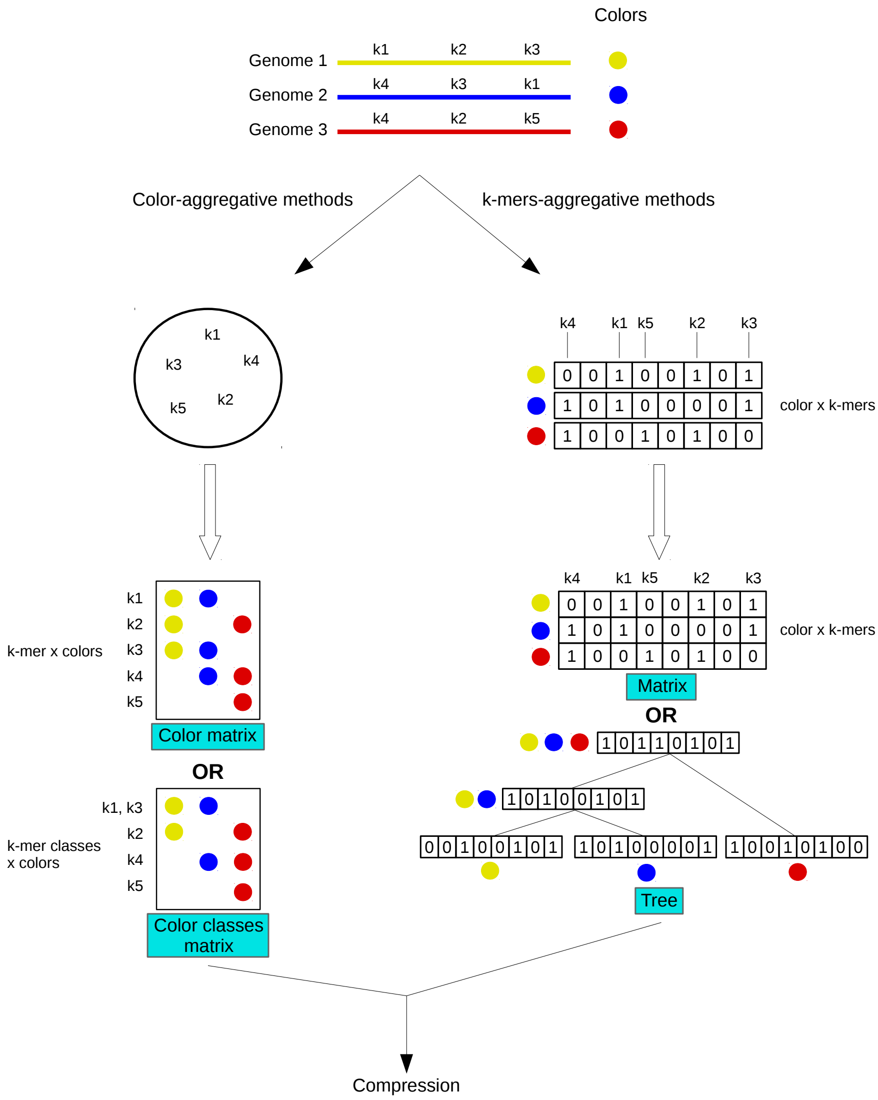


Figure 2.6: *Diagram of the differences between color-aggregative and k-mers aggregative methods.* Color-aggregative methods first group the k-mers of the different sequences ($k1$ to $k5$) before indexing and aggregate while k-mers aggregative methods first index all k-mers sets separately then aggregate them.

to each other allows to have a faster query, indeed it avoids to have too many k-mers distributed in very distant leaves of the tree and thus avoids to go down too often to the leaf. HowDeSBT also uses two bit arrays per node: B_{det} and B_{how}. B_{det} allows to know if for a node u, all its children have the same bit value at position i ($BFs[i]$). In the case where $B_{det}(u) = 1$ the bit value will be given by B_{how}. If $B_{det}(u)[i] = 1$ and

$B_{how}(u)[i] = 1$ then all $BFs[i] = 1$ in the subtree and conversely if all $BFs[i] = 0$ in the subtree then $B_{how}(u)[i] = 0$. For the active bits of a node u, B_{det} and B_{how} are defined as:

$$B_{det}(u) \triangleq B_{\cap}(u) \cup \overline{B_{\cup}(u)} \tag{B}$$

$$B_{how}(u) \triangleq B_{\cap}(u) \tag{C}$$

A basic uncompressed representation of the HowDeSBT tree structure is given in figure 2.7.

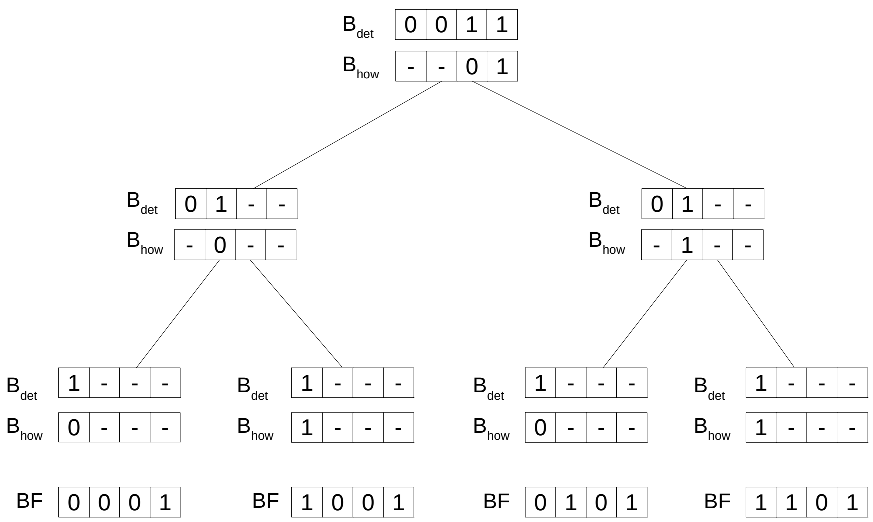


Figure 2.7: *Basic uncompressed tree structure of a HowDeSBT index.* The Bloom Filters at the leaves of the tree represent the sequences to be indexed, in our case the different bacterial strain genomes. B_{how} represents the nodes of the tree containing the union of the leaves below. B_{det} allows to know if the value of all child filters have the same bit value at a certain position. Positions containing a $-$ are positions that are considered inactive because they will never be looked at during a query. This can append either if their value is already known thanks to a parent node or if there are child nodes that do not contain the same value.

In HowDeSBT, a query is performed by recursively descending the tree. The search for the presence of the sequence is done using the set of its k-mers. Each k-mer is hashed which gives positions allowing the representation of the query as a Bloom filter that we will call *query BF* (see figure 2.8). Going down the tree, a recognition threshold θ is used ($\theta \in [\![0, 1]\!]$). It defines the minimal fraction of k-mers that a genome must contain to be considered as containing the queried sequence. This threshold is set to 0.9 in [Harris and Medvedev 2020] and is often placed between 0.7 and 0.9 [Marchet, Boucher, *et al.* 2020]. However, because of the number of errors in the long reads from nanopore sequencing we have set this threshold to 0.5 in the context of the thesis. During the descent in the tree two counters are incremented, the *presence* counter that counts the presence (positions at 1) of the query BF k-mers in the subtree and the *absence* counter that counts the absence (positions at 0) of the query BF k-mers in the subtree. The presence counter allows to stop the search at a node v as soon as the threshold θ is reached (see inequation D); the genomes of the subtree starting at v are then returned. Conversely the absence counter allows to stop the search when we know that θ will never be reached (see inequation E).

$$\theta \leq \frac{presence}{|Q|} \tag{D}$$

$$1 - \theta \leq \frac{absence}{|Q|} \tag{E}$$

where $|Q|$ is the size of the query BF. In the worst case which consists in having a read containing only k-mers forcing to go through all the tree, the complexity of the query will be $O(Q \times n)$ with Q the number of k-mers of the read and n the number of initial genomes.

Some bits of B_{det} and B_{how} will never be used during a query, these bits are called *inactive* $(-)$ (introduced by [Solomon and Kingsford 2018] in SSBT) which is the opposite of an *active* bit.

There are two possibilities for a bit at the position i to be inactive:

- $B_{how}(u)[i]$ is inactive if it cannot be determined at this node ($B_{det}(u)[i] = 0$). This means that there are descendant nodes with different values at the position i.

- $B_{det}(u)[i]$ and $B_{how}(u)[i]$ are inactives if there are already determined by an ancestor node ($B_{det}(ancestor(u))[i] = 1$).

These inactive (non-informative) bits are not saved in order to save space. This removal changes the indices into the filters, so the query is modified accordingly by using rank and select algorithm [Mäkinen and Navarro 2007]. The tree structure without the inactive bits is then compressed by using the succinct indexable dictionary from the RRR compression [R. Raman, V. Raman, and Satti 2007] method that allows membership, rank and select queries in O(1).

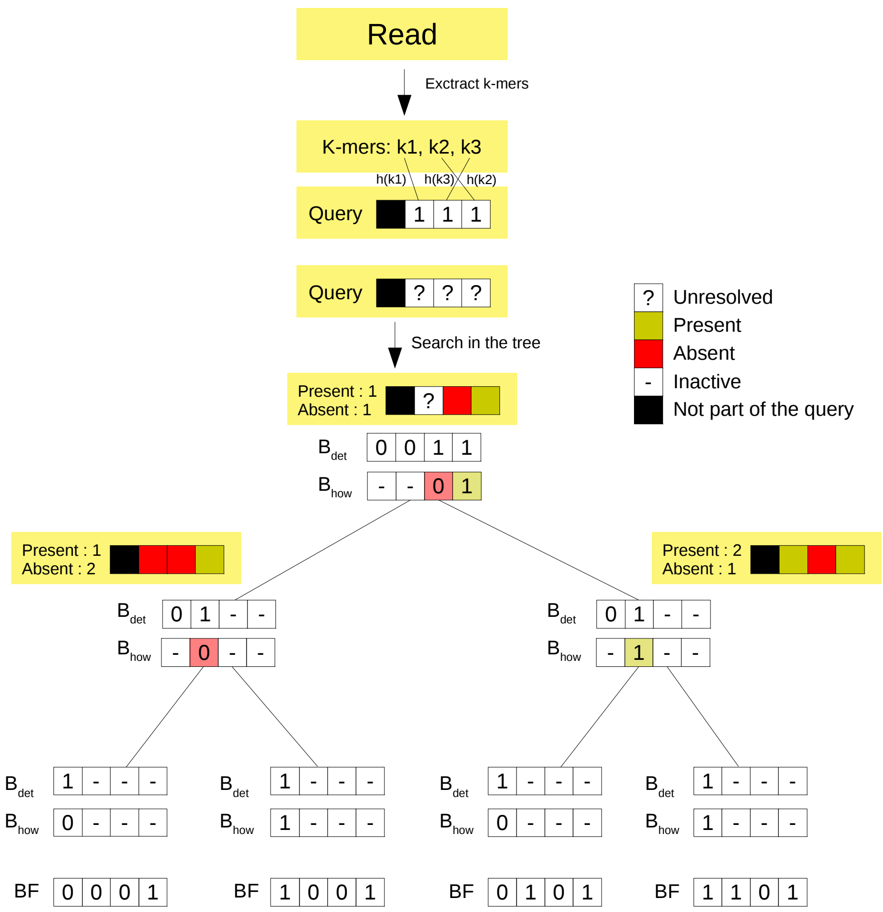


Figure 2.8: *Query of uncompressed HowDeSBT tree structure with a read.* The read is decomposed into k-mers which are hashed and inserted into a query Bloom filter of the same size as the index Bloom filters. In this example, the read contain three k-mers. The query is done as follow: the k-mers of the query BF are searched by going down the tree from the root. The presence and absence counters of the k-mers are incremented using nodes information and as soon as a node v has a ratio of presence to total number of k-mers of the query BF above a threshold *theta*, the genomes at the leaves are returned. In the figure, the recognition threshold is set to 0.5, which means that the read is considered present if there are more than 50% of the queried k-mers in the index. In this example, the read is found in the right part of the tree but not in the left part.

HowDeSBT's index structure has two major advantages over the conventional SBT structure:

- The acceleration and optimization of queries by using the B_{how} and B_{det} bit arrays as well as efficient clustering of initial genomes.

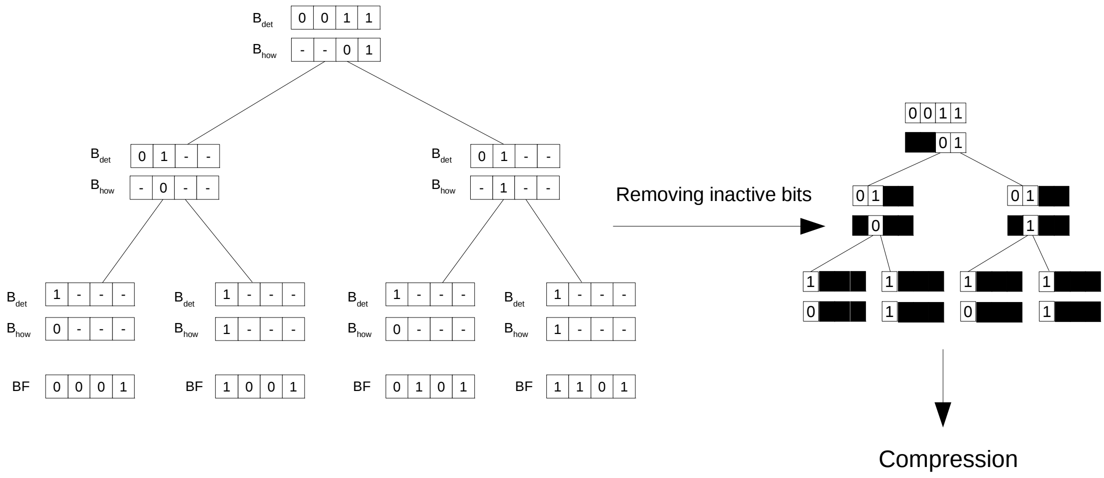


Figure 2.9: *Suppression of inactive bits from the HowDeSBT index.*

- The reduction of the space required for index storage by removing inactive bits and compressing BFs.

Finally, as we have just seen, all these indexing methods allow the representation of sets of genomes in a very time and space efficient way. The next step in genome identification, is the assignment of reads to genomes from the query results of these indexes. Table 2.5, at the end of this section, shows the main methods of bacterial identification using a k-mer based approach.

Identification softwares based on k-mers

The current reference in k-mer-based bacterial identification approach is Kraken [Wood and Salzberg 2014] and its second version Kraken 2 [Wood, Lu, and Langmead 2019]. Kraken was the first method to allow fast identification of reads in a metagenomic sample. Kraken uses a data structure related to the color-aggregation approach. Kraken uses the Jellyfish [Marçais and Kingsford 2011] k-mer counter which cuts the genomes into k-mers and makes a matrix with each k-mer and its count. However, for taxonomic assignment Kraken will save the LCA (Lowest Common Ancestor) corresponding to a k-mer instead of the k-mer count. This LCA corresponds to the lowest taxonomic assignment taking into account all organisms whose genomes contain the k-mer.

Like most tools working with k-mer from read, Kraken and Kraken 2 use the notion of canonical k-mer. Indeed, as dna has two strands, when searching for reads in a

Table 2.4: *Overview of existing color-aggregative and k-mer aggregative methods modified from [Marchet, Boucher,* et al. *2020].* Most of this methods are recent and new methods improving query or indexing time or space, or even the index size are developed every year.

Method	Software	K-mer set data structure	Set of sets data structure	Allowed compression	Reference
Color aggregative	SeqOthello	Othello hashing	1 or several color matrices	yes	[Yu *et al.* 2018]
	Bifrost	hash table			[Holley and Melsted 2020]
	Metannot				[Mustafa *et al.* 2017]
	Multi-BRWT				[Karasikov *et al.* 2019]
	Pufferfish			no	[Almodaresi, Sarkar, *et al.* 2018]
	BLight				[Marchet, Kerbiriou, and Limasset 2021]
	REINDEER			yes	[Marchet, Iqbal, *et al.* 2020]
	Vari(-merge)	BOSS			[Muggli, Bowe, *et al.* 2017; Muggli, Alipanahi, and Boucher 2019]
	Rainbowfish				[Almodaresi, Pandey, and Patro 2017]
	Mantis(+MST)	CQF			[Pandey, Almodaresi, *et al.* 2018] [Almodaresi, Pandey, Ferdman, *et al.* 2020]
	BFT	Bloom filter trie			[Holley, Wittler, and Stoye 2016]
Kmer aggregative	SBT, SSBT, AllSomeSBT HowDeSBT	Bloom filter	search tree/forest	yes	[Solomon and Kingsford 2016; Solomon and Kingsford 2018] [Sun *et al.* 2017] [Harris and Medvedev 2020]
	BIGSI, COBS, RAMBO		BF matrix/matrices		[Bradley *et al.* 2019; Bingmann *et al.* 2019; Gupta *et al.* 2020]
	DREAM-YARA		interleaving BF		[Dadi *et al.* 2018]
	BioBloom Tools		multiindex BF	no	[Chu, Sadeghi, *et al.* 2014; Chu, Mohamadi, *et al.* 2020]

genome, we do not know from which strand the read comes. To overcome this issue, it is possible to use the canonical version of each k-mer. To do that we need to define the reverse complement rc by $rc(A) = T$, $rc(C) = G$, $rc(G) = C$, $rc(T) = A$, and $rc(x.w) = rc(w).rc(x)$, where x is a nucleotide and w a DNA sequence on the alphabet A, C, G, T. The simplified canonical representation $can(t)$ of a k-mer k with $can(k) = min(k, rc(k))$, where the minimization is performed with respect to the lexicographic order. For example, $can(AGTCAC) = can(GTGACT) = AGTCAC$.
To speed up the search, in addition to storing k-mers, Kraken will look at the minimizer (default size of 15) of the k-mers. For a k-mer K of size k, the minimizer M of size m is the canonical M-mers of this k-mer that have the smallest lexicographic value. It can be calculated using:

$$M(K) = min(\sum_{i=1}^{k-m+1} can(K[i : i + m - 1])) \tag{F}$$

where $m < k$. Sometime, a minimum hash value is used instead of the minimum lexicographic value but this does not change the general principle.
The k-mers with the same minimizer are then stored consecutively and sorted alphabetically, which allows to group close k-mers. An index of the minimizers is saved which allows to load the corresponding k-mers matrix part into memory. As adjacent k-mers often have the same minimizer, the part of the k-mer matrix does not need to be reloaded

in memory which allows quick access to the k-mers LCA. To look for a read in the index, the read is split into k-mers whose minimizers are used to quickly find the LCAs. Then a subtree of the species tree is constructed according to these LCAs. Each RTL (Root To Leaf) which are paths from the root to the leaves of the subtree are then traversed. The path selected for a read is the leaf with the RTL of maximum weight according to the LCA count for each node.

Kraken version 2 (Kraken 2) is more efficient than Kraken in both space and time and give similar identification results. The minimizers of the genomes k-mers are directly stored in the Kraken 2 index instead of using a minimizer index allowing to find exact k-mers like in Kraken. In addition, a spaced seed which represents a mask that removes certain positions of a sequence is applied on the stored minimizers (see further section 3.1.2 for more details about spaced seed). We will then refer to this minimizers as spaced minimizers. The hash value of the spaced minimizer is then calculated and combined to the LCA corresponding to the k-mer. In case several k-mers with different LCAs have the same spaced minimizer it is associated to all LCAs.
Searching for a read in the Kraken 2 index is performed in the same way as for Kraken except that Kraken 2 looks at the LCAs corresponding to spaced minimizer of k-mer. This use of spaced minimizer makes the bacterial identification performed by Kraken 2 more efficient in time and decreases the space used by the index but also makes the bacterial identification harder. Indeed, a minimizer may be identical for different k-mers, which can affiliate this minimizer to more than one LCA and thus make the taxonomic assignment less accurate. Therefore, this approach makes it difficult to assign reads to taxonomically related individuals such as bacterial strains.

Many taxonomic assignment software have used LCA based approaches after the publication of Kraken. For example the recent metagenomic classifier ganon [Piro, Dadi, *et al.* 2020] uses the LCA to resolve uncertain read assignations. Ganon uses as index the data structures implemented in DREAM-YARA based on a k-mer aggregative method using interleaved Bloom filters (IBF). Then it uses k-mer counting lemma to assigns reads to a genome. The k-mer counting lemma considers all overlapping k-mers of a read and gives a lower bound corresponding to the number of k-mers shared between a read containing e error match and its reference (see section 3.1.1 to further information). However, some classifiers such as CLARK [Ounit, Wanamaker, *et al.* 2015] have developed approaches that dispense the use of the LCA. CLARK developed an approach based on the use of taxonomy-level specific k-mers stored in a structure which can be related to a color matrix used to classify reads without using an LCA algorithm. Indeed, it uses the taxonomy-level specific k-mers count to assign the reads to the taxonomy with the highest count. Furthermore, to avoid erroneous k-mer noise CLARK does not count reads k-mers with a low occurrence.

In 2018, a version of Kraken developed to give more accurate identification results, called KrakenUniq [Breitwieser, Baker, and Salzberg 2018], was released. This version allows a better recall and precision than Kraken on bacterial identification by using the coverage of unique k-mers present in each genomes to refine the taxonomic assignment of reads.

Table 2.5: *Non-exhaustive table of short read identification software based on the use of k-mers.*

Identification	**Index type/methods**	**Taxonomic assignation**	**Reference**
Kraken(-2|-Uniq)	Method similar to color-aggregative methods	LCA	[Wood and Salzberg 2014; Wood, Lu, and Langmead 2019] [Breitwieser, Baker, and Salzberg 2018]
LMAT			[Ames *et al.* 2013]
CLARK(-S)		Target-specific k-mers	[Ounit, Wanamaker, *et al.* 2015; Ounit and Lonardi 2016]
StrainSeeker	Guide tree containing kmers specific to nodes: similar to kmers-aggegative methods	Assignment of isolates (strain) with placement in the guide tree	[Roosaare *et al.* 2017]
k-SLAM	Hybrid method using k-mers composition to find read/genome overlap followed by SW alignment and 'pseudo-assembly' to infer genes and variant	LCA	[Ainsworth *et al.* 2017]

Despite the large number of bacterial identification software based on fragments indexing, most of them do not take into account the strain level. This lack of accuracy can be explained by the relatively small amount of information that short reads contain which makes it difficult to identify one bacterial strain from another. For some methods, a second explanation to this lack of accuracy is the loss of information during the indexing of the initial genomes. We end this section by one of the rare short read identification software that goes down to the strain level, StrainSeeker [Roosaare *et al.* 2017], a bacterial identification program based on a k-mer approach. Unlike other taxonomic assignment software such as Kraken for example, StrainSeeker will not attempt to assign reads to a taxonomic rank or a reference genome. Instead, StrainSeeker takes as input the taxonomic tree of the strains (as a guide tree) and the corresponding genomes. The goal is to build a tree index where each node contains a list of k-mers specific to that node. This index is built from the leaves (strains) to the root: StrainSeeker retrieves the list of k-mers of the strains and, following the guide tree, moves the k-mers common to a set of strains up into the corresponding nodes. Subsequently, non-strain specific k-mer are removed from the leaves k-mer sets. Unlike other methods, StrainSeeker does not search for reads in the indexed genomes but does the opposite. Therefore, the query is done by searching the k-mers from each node of the index tree in the read k-mers set. At each node a ratio k-mers observed/k-mers expected is calculated and allows to go down in the tree. This method allows StrainSeeker to assign reads to a clade. However, because of its index (not scalable to very large number of genomes) and its way of performing the query, StrainSeeker is developed to identify bacterial isolates (i.e. single strains, not mixtures) but can actually give good results on a small mixture of strains (See section 5.1).

2.1.3 Approaches based on variation graphs

In order to have a better representation of the variations between strains, graph-based approaches, called variation graphs have been developed. Storing them is more space-consuming than indexes based on k-mers composition but they are more informative as they represent several sequences with their variations. Variation graphs are bidirectional graphs of DNA sequences where the nodes represent subsequences of the input sequences and the edges represent the links between these subsequences existing in the original sequences (see figure 2.10). By associating each initial sequence to a color, it is possible to associate each color to a path of the graph representing the variations observed in the initial sequence ; these paths are then called *colored paths*. By following the different colored paths it is then possible to retrieve the initial sequences. An implementation of this variation graphs can be found in the vg toolkit [Garrison *et al.* 2018].

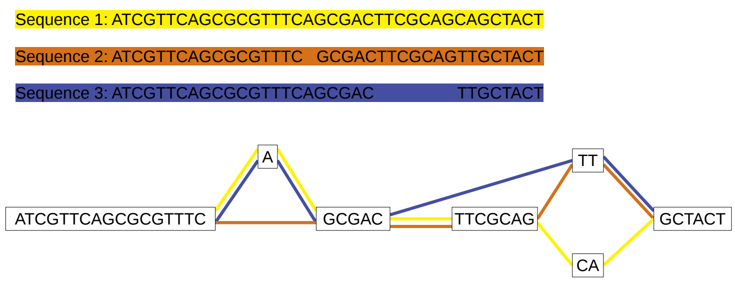


Figure 2.10: *Example of a variation graph built from three sequences.* Each colored path in the graph corresponds to one of the initial sequence.

These variation graphs allow an accurate representation of the variations between closes sequences, such as individuals of the same species or alleles of the same gene.

Although of great interest in the context of bacterial strain identification, these variation graphs are still under development. A recent bacterial strain identification and quantification software, StrainFLAIR [Da Silva *et al.* 2021], has been developed to demonstrate the feasibility of using variation graphs to store and query highly similar gene sequences of strains. By aligning the reads against a variation graphs one can measure the abundance of the different colored paths corresponding to the strain-specific genes and thus determine which strains are present in the starting sample by exploiting this abundances.

Conclusion on strains identification using short reads

The vast majority of bacterial strain identification software was developed during the rise of second generation sequencing (NGS) technologies. However, as these programs have adapted their approaches to short sequencing reads containing relatively few errors, most of them cannot work properly with very erroneous long reads. In the next section we discuss of software that have been able to cope with both short and long reads.

2.2 The specificity of identification of bacterial strains using long reads

Since the advent of long reads in the 2000s, bacterial identification software has evolved to adapt to these new reads. Despite the advantages of their size the difficulty in performing identification using such reads is their high error rate which makes alignments and k-mer index queries more complex. The problem becomes even more complicated if we consider the high similarity of reference genomes in the case of bacterial strain identification.

One of the first ideas that emerged to deal with this kind of reads was either (1) correcting these long reads with non-erroneous short reads (hybrid correction) or (2) using the sequencing depth [Fu, Wang, and Au 2019; H. Zhang, C. Jain, and Aluru 2020].

- Hybrid correction methods are composed of two categories: (1) correction methods that assemble the short reads by making a De Bruijn graph and align the long reads to this assembly for correction (e.g. LoRDEC [Salmela and Rivals 2014], ECTools [H. Lee *et al.* 2014], HG-CoLor [Morisse, Lecroq, and Lefebvre 2018]), and (2) correction methods that directly align the short reads to the long reads (e.g. Hercules [Firtina *et al.* 2018], CoLoRMap [Haghshenas *et al.* 2016], proovread [Hackl *et al.* 2014], LSC [Au *et al.* 2012]).

- The non-hybrid methods align the long reads against themselves in order to correct them using the sequencing depth (e.g. LoRMA [Salmela, Walve, *et al.* 2017], FLAS [Bao *et al.* 2019], CONSENT [Morisse, Marchet, *et al.* 2021] or Canu [Koren *et al.* 2017] which is a non-hybrid assembly software containing a read correction module).

However, these correction methods are very time-consuming which is a major issue in the context of this thesis as we aim to do a quick identification of bacterial strains. However, even without correcting the reads, some tools manage to realise bacterial identification from long reads.

2.2.1 Software developed to handle short reads, compatible with long reads

Some software originally designed to work with short reads can work with long reads. For example the k-mer based methods implemented in Kraken 2 and StrainSeeker presented in section 2.1.2 give correct results even when used with long reads. For methods based on alignment of reads with reference genomes, it is possible to adapt the approaches by simply changing the aligner used by an aligner specific to long reads such as Minimap2 [H. Li 2018] or Graphmap [Sović *et al.* 2016]. Certain tools based on sequence alignment work directly with long reads such as Centrifuge or Kaiju, both based on the FM index. Centrifuge is the software used by the ONT's EPI2ME platform, which is a data analysis platform that allows, via the use of What's In My Pot (WIMP), real-time quantitative identification of species from metagenomic samples. However, the few experiments we have done with WIMP have shown that it is often limited to a classification at the species or genus level.
Finally some metagenomic classifiers such the MEGAN software have modified their method to allow the use of long erroneous reads (e.g. MEGAN-LR [Huson, Albrecht, *et al.* 2018]). MEGAN-LR uses an interval-union LCA algorithm which consists in cutting the long reads into parts of a certain size and assigning an LCA to each part. The assignment consists in using the most present taxon among the LCAs of each part of the reads.

However, as for short read identification, only few of these software allow to go down to the level of the strain. Among these software working with bacterial strains, we can cite StrainSeeker and Centrifuge.

2.2.2 Native long read identification software

There are still relatively few strain identification software packages developed specifically for long reads. The first of these is MetaMaps [A. T. Dilthey *et al.* 2019] a software based on the alignment of long reads against reference genomes. In addition to simultaneously carrying out read assignment and quantification, it provides information on read mapping positions and read quality. However, it is not based on an existing long read alignment software but on a fast approximate alignment using MashMap [C. Jain, A. Dilthey, *et al.* 2017]. In short, to speed up the alignment of reads, MashMap uses a combination of a fast approximate read alignment algorithm using minimizers (Winnowing algorithm) combined with a Jaccard index calculation using the MinHash technique (see further section 3.2.1). In addition to this fast alignment a probabilistic mapping quality model is used and incorporated into an EM approach to estimate the sample composition and location of the mapped reads. However, MetaMaps is still a fairly new software and the current version remains complicated to install and we were not able to install it properly with the few installation attempts we tried.

Another software to be mentioned is the NanoMAP software [Hall, Speed, and Woodruff 2020], which is a very recent software allowing the characterisation at strain level of samples using long reads. NanoMAP is based on the simple assumption that rather than focusing on how to remedy the problem of long reads error rate, it is sufficient to exploit the structural information gained from the length of the reads. The goal is to make the most of the mapping quality scores (MAPQ) to get down to the bacterial strain. NanoMap algorithm works as follow: First, for a species, NanoMAP concatenates the genomes of the strains. Then, the reads are aligned to these genomes and the MAPQ of each read is calculated. This score is 0 if a read optimally aligns to more than one strain genome, and reads with a single optimal alignment in a genome will have a high MAPQ score. Concatenation of strain genomes into a single sequence allows only the most informative reads to be used because reads from the core genome of a species will have a MAPQ score near 0. Clustering of nearby genomes is performed using reads aligning to multiple genomes. The clusters formed give a first idea of the bacteria present in the sample. Then, within each cluster, the reads with a high MAPQ score (reads with a good alignment on an unique genome) are used to identify the strain genomes actually present in the sample. Finally, a calculation of the abundance of the strains is performed by an EM algorithm.

We are not aware of any software based on a k-mer-based method specific to long reads.

Conclusion on strains identification using long reads

More generally, there is a lack of strain identification software working with long reads. The functionalities of the ONT technology are of interest for rapid identification of bacterial strains. Indeed, ONT technology could allow a real time identification of strains which actually takes several weeks through sequencing platform. It would also be possible to stop the sequencing quickly and wash the flowcell in order to reuse it afterwards and even use the "Read Until" technology to artificially amplify the genome of subdominant strains that are present in the original sample by preventing the read of the dominant strains.

The recent increase of complete bacterial genomes in databases enable the identification at the strain level. However, even if there are many bacterial identification softwares, very few of them work at the strain level and many stop at the species level or higher. The fact that they cannot go down to the strain level is partly explainable by the lack of information that is contained in a short read. Long reads technologies offer more information but is still not mature, so only a few identification software exist to work with them.

This thesis proposes a new k-mer-based approach to identify, on a single laptop, bacterial strains in mixtures or isolates via the use of long reads from ONT sequencing.

Chapter 3

Strain identification from long reads with spaced seeds

Contents

Preamble: In this chapter we will look at a way of assigning long nanopore reads from a sample, such as bacterial isolates or mixtures of few strains, to their potential bacterial strains.

In the first chapter (see 1.2.3), we saw that long reads, and more specifically Oxford Nanopore long reads, still have a high error rate, even if it has decreased rapidly as technology and basecallers improved. Our goal is to manage this high sequencing error rate without correcting the reads since correction tends to discard small variations in the sequences that are crucial for bacterial strain identification.

3.1 Assignment of Nanopore reads to strains

In this section, we focus on the problem of assigning newly sequenced reads to the most likely reference genomes from which they may originate. This problem can be reduced to the estimation of a similarity measure between a read and a genome. This measure will be mostly derived from the best alignment between the read and the genome. The thesis started from the strong premise, similar to the NanoMAP assumption (see 2.2.2), that most of the reads from the Oxford Nanopore technology are long enough to contain the information to separate bacterial strains from each other, or at least separate small groups of strains from each other. In addition, we chose to use a k-mer-based approach to assign reads to strains as these methods are faster and take less space than those based on sequence alignment.

3.1.1 Assignment using k-mers

Worst case analysis

To assign a read to a reference genome, we can use the k-mer counting lemma [Reinert *et al.* 2015] as it is the case in ganon [Piro, Dadi, *et al.* 2020] for example.

Counting lemma: Let G be a reference sequence, R a read of size l_r included in G, e the maximum number of erroneous positions in R. Consider the multisets of k-mer occurrences KG and KR then the cardinal of $KG \cap KR$ is at least $hit_R = |KR| - k \times e$, where $|KR| = l_r - k + 1$.

The lemma define the worst case scenario where hit_R is minimum because errors are equidistantly distributed along a read. For example, a read R of size $l_R = 10$ contains $nk_{mers\,R} = 8$ kmers of size $k = 3$. If there are at most 2 errors ($e = 2$) in R, R and the reference must share at least $8 - 6 = 2$ k-mers.
However, in practice this equation is not restrictive enough to be useful. Assume an error rate of 5%, k-mers of size 31 and a read of size 1000 are reasonable parameters when working with Nanopore reads. In this case $hit_R = 970 - (31 * 50) < 0$ and is therefore of no use. Most of the time the counting lemma does not bring any exploitable constraint. In practice, in a certain number of software, references are filtered by requiring that at least 70 to 90% of the read k-mers are present in the reference. This threshold works well with sequences with few errors such as short reads. In our case, as we are working with long reads containing many errors, this threshold has been set at 50% (see section

2.1.2). It is therefore required that 50% of the read k-mers are found in the reference ($hit_R = \frac{nk_{mers R}}{2}$).
By reversing the equation of the counting lemma, one deduces the maximum error rate allowed to find at least half of the read k-mers in the reference k-mer set. The equation to find the number of errors is:

$$e = \frac{nk_{mers R}}{2k} \tag{A}$$

For a read R of size $l_R = 2000$ and a k-mer size $k = 31$ (default size in the Kraken software), there are $nk_{mers R} = 1970$ k-mers in R and we need to find at least half of these k-mers ($hit_R = 985$). The maximum number of errors allowed to find at least 985 k-mers of the reads in the reference would be $e = \frac{985}{62} \approx 15.9$ which represent a $\approx 0.8\%$ maximum error rate, which would be too low for long reads.

Average analysis

For this reason, most people prefer to fix the threshold with an average analysis, using a simple binomial model [Singh 2020] and an assumption of a uniformity of errors with error rate p. Under this assumption, finding the number of common k-mers corresponds to finding the number of non-erroneous k-mers, which is the total number of k-mers multiplied by the probability of having no error in a k-mer of size k. If X is a random variable denoting the number of errors, we get:

$$hit_R = nk_{mers R} \times P(X = 0) \tag{B}$$

According to the probabilities of the binomial distribution, for all x in $[0, k]$ we have:

$$P(X = x) = C_k^x \times p^x \times (1-p)^{k-x} \tag{C}$$

For $x = 0$, the formula becomes $P(X = 0) = (1-p)^k$ and we fall back on a uniform distribution.

If we take as example a read R of size $l_R = 2000$ and a 5% error rate, the average number of common k-mers of size 31 between the read and the reference will be about 402 k-mers (see equation B).

In a more general way, if we require that $\alpha\%$ of k-mers are without error on average we have:

$$\alpha \times nk_{mers R} = nk_{mers R} \times (1 - p)^k \tag{D}$$

Then, we can deduce (if $nk_{mers R} \neq 0$):

$$\begin{aligned} \alpha &= (1 - p)^k \\ p &= 1 - \sqrt[k]{\alpha} \\ k &= \frac{log(\alpha)}{log(1 - p)} \end{aligned} \tag{E}$$

with p the error rate, α the percent of k-mers without error and k the k-mer size. It is then possible to calculate one parameter (α, p or k) by fixing the two others.

Now, to have 50% of the k-mers in common between a read and a reference we would need, on average, 2% error for a read of size 2000 and a k-mer size of 31 and about 4.5% error with a k-mer size of 15 (see equations E).

An average error rate of 5% can be achieved with nanopore sequencing, for example *S. thermophilus* reads used in experiments described in section 5.1 have the following error rate (see table 3.1).

Table 3.1: *Average percent error rate in the S. thermophilus strains sequences.* The filters retain only the sequences with a quality greater than 9 and a size greater than 2000 bp. Reads have been basecalled using the version v4.4.1 of the Guppy ONT basecaller. Table from [Siekaniec *et al.* 2021]

Errors	**Mismatchs**	**Deletions**	**Insertions**	**Total**
All sequences	1.50%	2.16%	1.40%	5.06%
With filters	1.44%	2.08%	1.37%	4.89%

However a certain proportion of reads will have an error rate over 5%.
Using the equation E we can get the maximum size of k-mers necessary to have on average half of occurrences without error, as a function of error rate. With a 15% error rate, the size of k-mers needed to have on average half of the non-erroneous k-mers will be $k = 4$. This is clearly too small for a recognition at strain level. Tretramer composition is used in metagenomics to predict the genus or the species level [Sandberg *et al.* 2001] using sequences with very few errors and remains inadequate for the identification of bacterial strains. If we decrease the error rate to 5% the needed size of the k-mers is 13 which is actually more usable.

We present in the top part of the table 3.2 an experiment on the distribution of assignment using HowDeSBT, for various values of k. It has been computed on 4 000 reads sequenced from a *S. thermophilus* JIM8232 strain. The sequencing error rate is estimated to 5%. Decreasing the size of the k-mers allows more reads to be assigned but they will be assigned to more reference genomes, thus increasing the ambiguity of the results. The challenge is then to find a compromise for the k-mer size, large enough to assign a maximum number of reads in a specific way and small enough to be able to assign the read to at least one reference genome.

Table 3.2: *Effect of k-mer size reduction on the assignment of 4000 reads of S. thermophilus JIM8232 on a database of 77 S. thermophilus strains + S. macedonicus ACA-DC 198 + L. delbrueckii subsp. bulgaricus ATCC 11842 using HowDeSBT.* The effect of using spaced seeds of size 17, 15 and 13 with a weight of respectively 15, 13 and 11 is also presented for comparison with k-mers of size 15 and 13. Decreasing the size of k-mers allows to assign more reads but less specifically. The use of spaced seeds improves the trade-off between the number of reads assigned and the specificity of assignment compared to the use of k-mers of the same size. The assignment results presented in the table are percentages of assigned/unassigned reads.

K-mer size	Unassigned reads (%)	Reads assigned to a maximum of 5 strains (%)	Reads assigned to more than 5 strains (%)	Reads assigned to a single strain (%)
31	23.75	17.25	59	10.13
15	0.83	16.50	82.67	3.60
13	0	4.68	95.32	1.83
11	0	0	100	0
Spaced seed (size/weight)	**Unassigned reads (%)**	**Reads assigned to a maximum of 5 strains (%)**	**Reads assigned to more than 5 strains (%)**	**Reads assigned to a single strain (%)**
33/31	12.53	12.95	74.52	6.58
17/15	2.30	8.58	89.12	4.10
15/13	0.05	5.10	94.85	2.15
13/11	0	0	100	0

Overall, the error rate of current sequencers hardly allows to find a good combination of the parameters p, k and α. Moreover, an exact assignation is not always possible since we do not necessarily have in the databases the sequence of the correct reference genome. In this case, we want to find the closest known genome, which will induce an additional error rate due to the differences between the sequenced strain and the reference one, rather than linked to the sequencing technology.

One of the particularities of Nanopore reads is the high error rate in homopolymer sequences which increases very rapidly with the homopolymer size [Delahaye and Nicolas 2021]. In order to be less sensitive to sequencing errors, we started this thesis by compressing these homopolymer sequences to one letter to avoid dealing with this kind of information. This technique has been used in the software LSC corrector [Au *et al.* 2012], which compresses the homopolymers in both short and long reads to improve the alignment of one against the other. It is also possible to keep homopolymers of size less then four where the error rate can still be considered acceptable [Delahaye and Nicolas 2021] and to reduce other homopolymers to a size of 4. Of course, a certain amount of sequence information is lost after this compression. In fact, our preliminary experiments showed that the loss was too high to discriminate the generic sequences of some close strains of the same species. Therefore, this idea was later abandoned for this thesis.

3.1.2 Assignment using spaced seeds

Instead of decreasing the k-mers size to be less sensitive to sequencing errors, it is possible to use an extended notion of k-mer explicitly including a tolerance with respect to errors. This is the goal of spaced seeds (see figure 3.1). A spaced seed is a binary pattern composed of 1 and 0 (sometimes noted # and −) representing respectively the notion of match and joker (or don't care) position. A further constraint is that a spaced seed must start and end with a 1 character. The seed is used as a mask when comparing two sequences. For example the seed 10011 represents one match followed by two don't care positions and then two matches. A spaced seed of size s, has a weight $w \leq s$, which is its number of 1 (matching position). The sequence of letters at matching positions is called a *q-gram* (or spaced k-mer in some studies). For example, the comparison of the sequences $ATGC$ and $ATTC$ using the spaced seed 1101 of size four and weight three produces a match, with q-gram (trigram) ATC. The effect of using spaced seeds ($s = 13, 15, 17$ and $w = 11, 13, 15$) on a *S. thermophilus* JIM8232 read assignment with an error rate of about 5% is presented in the bottom part of the table 3.2. Compared to a k-mer of the same size, using a spaced seed improves the trade-off between the number of assigned reads and their specific assignment to one or a few reference genomes. For example, using the spaced seed size 15 and weight 13 allows more reads to be assigned than a 13-mer while missing fewer reads than a 15-mer.

The first use of spaced seeds was sequence comparison in seed-and-extend algorithms such as PatternHunter [B. Ma, Tromp, and M. Li 2002]. A q-gram match corresponds to a potential alignment position that will be subsequently confirmed or denied by the extend part of the algorithm. The spaced seed provided the starting point of the search.

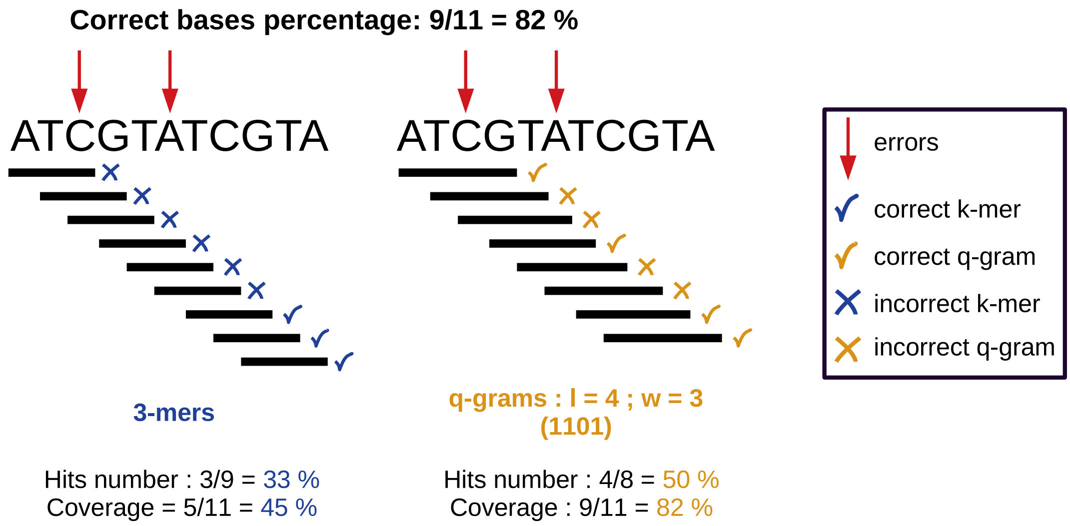


Figure 3.1: *Comparison of the behaviour of a k-mer of length 3 and the spaced seed* 1101 *of length 4 and weight 3 on a sequence of size 11 containing 2 errors.*

This paper showed empirically that spaced seeds improved the sensitivity/specificity tradeoff in homology search by alignment. It was subsequently analysed more formally in other works such as [Kucherov, Noé, and Roytberg 2006].
Spaced seeds were also used with k-mer based approaches for approximated sequence alignment [Burkhardt and Kärkkäinen 2003]. The use of spaced seeds has also been shown to improve alignment-free methods. In 2014 [Leimeister *et al.* 2014] showed that using q-grams instead of k-mers allowed to obtain a more accurate phylogenetic tree reconstruction. Then [Morgenstern *et al.* 2015] proved that matching q-grams between sequences provides an even better phylogenetic distance estimator. The same year, an extended implementation of Kraken call Seed-Kraken [Břinda, Sykulski, and Kucherov 2015] has shown that spaced seeds allow a significant improvement in the accuracy of read classification compared to traditional contiguous k-mers. This improvement in accuracy was subsequently also demonstrated by CLARK-S [Ounit and Lonardi 2016], a version of the metagenomic reads classifier CLARK [Ounit, Wanamaker, *et al.* 2015] with spaced seeds. The use of spaced seeds is also present in the most recent version of Kraken: Kranken 2 against which we compared our method (see section 5.1). A simple example showing how the use of a spaced seed may provide better results than k-mers is proposed in figure 3.1.

Some authors have introduced algorithms working on several seeds simultaneously (multiple seeds). This provides an advantage either in seed-and-extend sequence alignment or in alignment-free comparison methods [Noé and Martin 2014; Leimeister *et al.* 2014]. However, in the latter case, each k-mer of a reference genome will be represented by all q-grams related to the use of the different spaced seeds. For example a set of three different spaced seeds requires for each k-mer of the genome three q-grams to save. There are two possibilities for sequence indexing: (1) each spaced seed leads to the construction of a new index for the reference sequences which makes the use of the index more complex because it is necessary to solve three separate queries and merge the result afterwards, (2) all q-grams are inserted into a single index, which may for some type of index such as those based on Bloom filters increase the false positive rate. The latter option is the one used by the miBF data structure [Chu, Mohamadi, *et al.* 2020]. This structure replaces the different hash functions of the Bloom filter by using different spaced seeds applied to the k-mers which are then hashed with a single hash function. This technique allows to combine high sensitivity with low memory usage. In our case, after preliminary experiments, we decided to work with a single spaced seed because the use of several seeds requires to increase the size of the Bloom filters to have the same false positive rate and increases the time of creation and query of the index.

Actually, the use of spaced seeds, as opposed to k-mers, has led to improve performance in terms of sensitivity and specificity in many applications at the expense of higher computational cost. To overcome this problem, fast approaches for spaced seeds indexing have recently been developed [Girotto, Comin, and Pizzi 2018b; Girotto, Comin, and Pizzi 2018a; Petrucci *et al.* 2020]. However, these approaches were not used during the thesis because the creation time of the index was not the limiting point of our method. However, it would be interesting to integrate one of them in our tool (see section 4.2) in order to accelerate the index construction.

In the case of alignments, spaced seeds can be compared by their sensitivity [Herms and Rahmann 2008], which represents their probability of matching on an alignment generated by a specific probabilistic alignment model. During the thesis, we used spaced seeds in the context of alignment-free sequence comparison. The similarity of two sequences is measured by the number of matching q-grams between them. We then defined a genome as the set of all q-grams that compose it. It is then possible to index these q-grams efficiently as with k-mers, in a Bloom filter. We indexed the reference genomes in this way. By comparing the sets of q-grams of two sequences it is then possible to estimate the similarity of the genomes. In doing so, as with k-mers, the positions of q-grams in the genome is lost. The hit number is associated with a second estimator of the similarity, which is the coverage of the sequence. The coverage [Noé and Martin 2014] can be defined as the number of bases covered by the matching q-grams. Obviously coverage and hit number depend on the error distribution but also on the shape of the spaced seed used.

The optimal spaced seeds for alignment will not be the same as the optimal one for alignment-free comparisons [Noé and Martin 2014]. Studies have then been carried out on the search for the best spaced seeds pattern according to their use. In 2017, it led to the creation of iedera [Noé 2017], a tool for selecting and designing spaced seeds. Recent studies further improve the seed sensitivity like the rasbhari algorithm [L. Hahn *et al.* 2016] and the ALeS software [Mallik and Ilie 2021] developed to generate spaced seeds specific to alignment or alignment-free comparison. Moreover, the VSTseed software [V. Titarenko and S. Titarenko 2021] demonstrated the utility of using periodic spaced seeds (seed with a periodic pattern) to accurately determine the position of short reads in a reference. It is therefore possible to obtain very sensitive task-specific seed patterns.

In our case, spaced seeds were used with an alignment-free comparison method and allows us to be less sensitive to the ONT sequencer error rate. We propose the following procedure to mix Bloom filters with spaced seeds: (1) the studied strain genomes (mainly *S. thermophilus* genomes) are split into k-mers. (2) A spaced seeds is applied on these k-mers to obtain the q-grams of these genomes. (3) These q-grams are inserted into a Bloom filter based data structure (see further section 4.1). The k-mers size we used is 15 with a weight of 13 and the seed pattern is 111111001111111.
This seed pattern was selected between all spaced seed pattern of size in a range from 9 to 21 using the iedera software for finding the best pattern (see figure 3.2). This selection is based on identification results from 4000 reads originating from the *Streptococcus thermophilus* JIM8232 strain using our method ORI, which will be presented in section 4.2.

3.1.3 Perspectives: the use of indel seeds

Globally, the concept of spaced seeds allows to be less sensitive to mismatch errors between two sequences. Restrictions exist for this concept. For example, the local alignment software YASS [Noé and Kucherov 2005] implements spaced seeds called *transition constrained seeds* using "wildcard positions" that accept only transition type mismatches (A to G or T to C) and no transversions (A or G to T or C).

For insertion/deletion errors (indel) an extended type of spaced seeds can be used ;

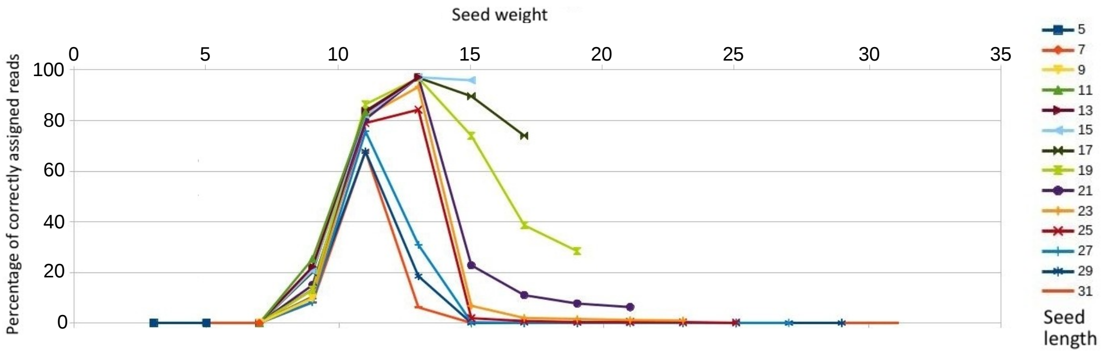


Figure 3.2: *Identification results on 4000 reads of the Streptococcus thermophilus JIM8232 strain using different seed lengths and weights.* The seed patterns were calculated using the iedera software and the best identification result is obtained using a seed of size 15 and weight 13 (light blue).

the *indel seeds*. An indel seed is a sequence on alphabet $\{0, 1, X\}$, where 1 represents a match, 0 represents a wildcard position (match or mismatch) and X represents a joker position that can be any event: a match, a mismatch, an insertion in the reference or an insertion in the query. As for spaced seeds, indel seeds must start and end with a 1 character. However, unlike for classic spaced seeds, the number of possible q-grams N_q associated to a given indel seed is greater than one. N_q depends on the number of X runs (N_{range}) as well as their length (L_{range}) in the indel seed (see figure 3.3). Formally:

$$N_q = \prod_{i=1}^{N_{range}} (1 + L_{range}[i]) \tag{F}$$

The usefulness of using this type of seed in homology search by sequence alignment was demonstrated by [Mak, Gelfand, and Benson 2006]. Works have also been done to find the most sensitive indel seeds in an efficient way like in [K. Chen *et al.* 2009]. Subsequently, these kind of seeds has been used by GraphMap [Sović *et al.* 2016], a sequence mapping algorithm that showed a gain in accuracy when using indel seeds with nanopore sequences. This algorithm has been used especially for the alignment of long reads. Indeed the use of indel seeds is particularly relevant in the case of long reads from Nanopore sequencing because of their high insertion/deletion rate (see section 1.2.3).

We are not aware of any work actually using indel seeds in the context of alignment-free sequence comparison. This is probably due to the rapid increase of the number of

```
1X1X1

10101
1011
1101
111
```

$N_{range} = 2$
$L_{range} = \{\mathbf{1},\mathbf{1}\}$
$N_q = (1+\mathbf{1}) \times (1+\mathbf{1}) = 4$

```
1X1XX1

101001
10101
1011
11001
1101
111
```

$N_{range} = 2$
$L_{range} = \{\mathbf{1},\mathbf{2}\}$
$N_q = (1+\mathbf{1}) \times (1+\mathbf{2}) = 6$

```
1XX1XX1

1001001
100101
10011
101001
10101
1011
11001
1101
111
```

$N_{range} = 2$
$L_{range} = \{\mathbf{2},\mathbf{2}\}$
$N_q = (1+\mathbf{2}) \times (1+\mathbf{2}) = 9$

Figure 3.3: *The number of q-grams to be considered as a function of the number of X runs and their size in the indel seed.*

q-grams with respect to the number of X runs in the seed. Indeed, the issue is similar to the use of multiple seeds but with a supplementary multiplicative factor that severely impact the indexing space and the query time. Because of this issues, indel seeds have not been used in our work. However the creation of an index able to manage this kind of seed would be very useful for further developments in the context of Nanopore data.

3.1.4 Non-assignment of reads

The fact that some reads cannot be assigned to a reference genome, is a desirable behaviour. In case of contamination of a sample by a non-bacterial genome for example, the correct behavior is not to assign reads from this genome to any other genome in the reference genome database. Recently [Marcelino, Holmes, and Sorrell 2020] have pointed out that the use of taxon-specific reference databases for metagenome classification can pose the problem of false species identification. This misclassification can be due to reads containing sequences that are conserved across all living organisms. This problem is particularly true for software working with short reads containing little information but can also be a problem with long reads when the contaminant is a species close to one of the reference genomes used for identification. This problem directly concerns our method and has been taken into account. It will be discussed in the future section 4.2.2 and in the analysis of results in section 5.3.

3.2 Strain identification

In this section we will discuss about the reads assigned to one or more genomes and their use in identifying the bacterial strains actually present in the sample.

3.2.1 Measuring the proximity between strains

It is clear that a read can be considered to be present in several genomes. In every genome, there exists a core part that is common to many species. When working with several strains of a same species, this common portion may even become predominant.

In order to better understand the global landscape of proximity between strains and therefore the difficulties of the identification of some of them, we started by a comparative study of genomes themselves. In general, genomes are compared either on the basis of their *gene content* or by using their *whole genome sequence*.

In the case of comparison of strain genomes based on the study of gene content, the set of genes present in a species is called its pangenome. It is composed of the *core* genome comprising all the genes present in all the individuals of the species and the *variable* genome containing the other genes. More precisely, the pangenome can be separated into three classes (see figure 3.4): (1) the *persistent* genome containing genes present in all (core genome) or a large part of the strains, (2) the *shell* genome containing genes present at a medium frequency in the species and (3) the *cloud* genome containing genes present at a low frequency in the species (such as genes specific to a single strain) [Koonin and Wolf 2008; Collins and Higgs 2012; Gautreau *et al.* 2020].

In case of a comparison based on whole genome sequences, there are many ways to estimate their proximity. The main one is to calculate the average nucleotide identity (ANI) distance [Goris *et al.* n.d.] between the two sequences (see section 1.1.2). The ANI can be defined as the mean nucleotide identity of orthologous shared gene pairs. It can be calculated using an alignment of sequences (e.g. with BLAST) or without alignment.

Other measures exist to calculate the proximity of genomes. Among these measures, there are those based on the k-mer composition of genomes allowing the calculation of distances such as the Jaccard distance:

$$J(G1, G2) = \frac{|G1 \cup G2| - |G1 \cap G2|}{|G1 \cup G2|} \tag{G}$$

where G1 is the set of k-mers (or q-grams) from the first genome and G2 is the set of k-mers (or q-grams) from the second genome. This Jaccard distance can also be estimated

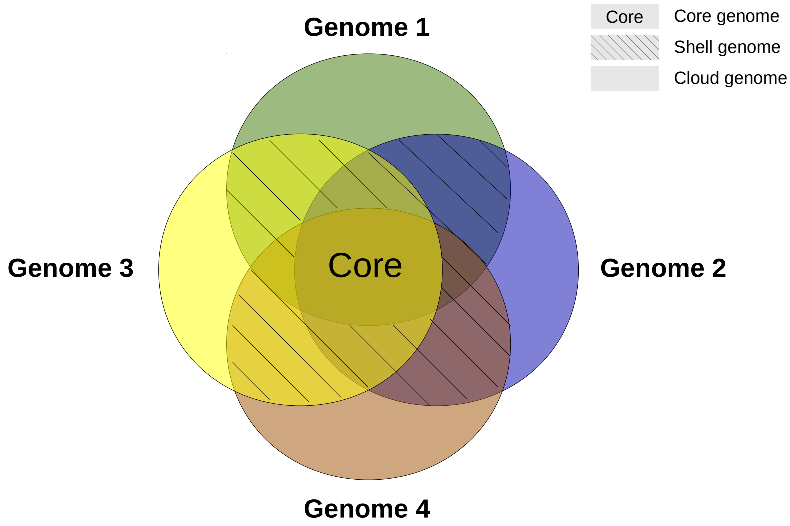


Figure 3.4: *Venn diagram representing the pangenome of 4 genomes.* As there are few genomes, the persistent genome can be considered equal to the core genome and the cloud genome contains only the genes specific to one genome.

very efficiently using MinHash, a technique invented by Andrei Broder in 1997 [Broder 1997]. The MinHash technique works as follow (see figure 3.5): the k-mers of two genomes are retrieved and each k-mer is passed through a hash function h to obtain a hash value. The resulting hash sets, $G1$ and $G2$ containing $|G1|$ and $|G2|$ distinct hashes are then used to approximate the *Jaccard index* ($J(G1, G2)$). It corresponds to the intersection between $G1$ and $G2$ divided by their union and can be approximated by considering a much smaller random sample of $G1$ and $G2$. This random sample is built by recovering only the n smallest hash values for $G1$, $G2$ and $G1 \cup G2$ called respectively $S(G1)$, $S(G2)$ and $S(G1 \cup G2)$. So, $S(G1 \cup G2) \neq S(G1) \cup S(G2)$ and $S(G1 \cap G2)$ corresponds to $S(G1 \cup G2) \cap S(G1) \cap S(G2)$. As $S(G1 \cup G2)$ is obtained from a random sample of $G1 \cup G2$, the fraction of $S(G1 \cup G2)$ shared by $S(G1)$ and $S(G2)$ can be used to calculate an unbiased estimate of $J(G1, G2)$. Finally, the Jaccard distance ($J_{distance}$) can be recalculated from the Jaccard index by doing $J_{distance}(G1, G2) = 1 - J(G1, G2)$. This technique is used by Mash [Ondov *et al.* 2016] and sourmash [Pierce *et al.* 2019] to perform sequence similarity comparisons by estimating the Jaccard distance.

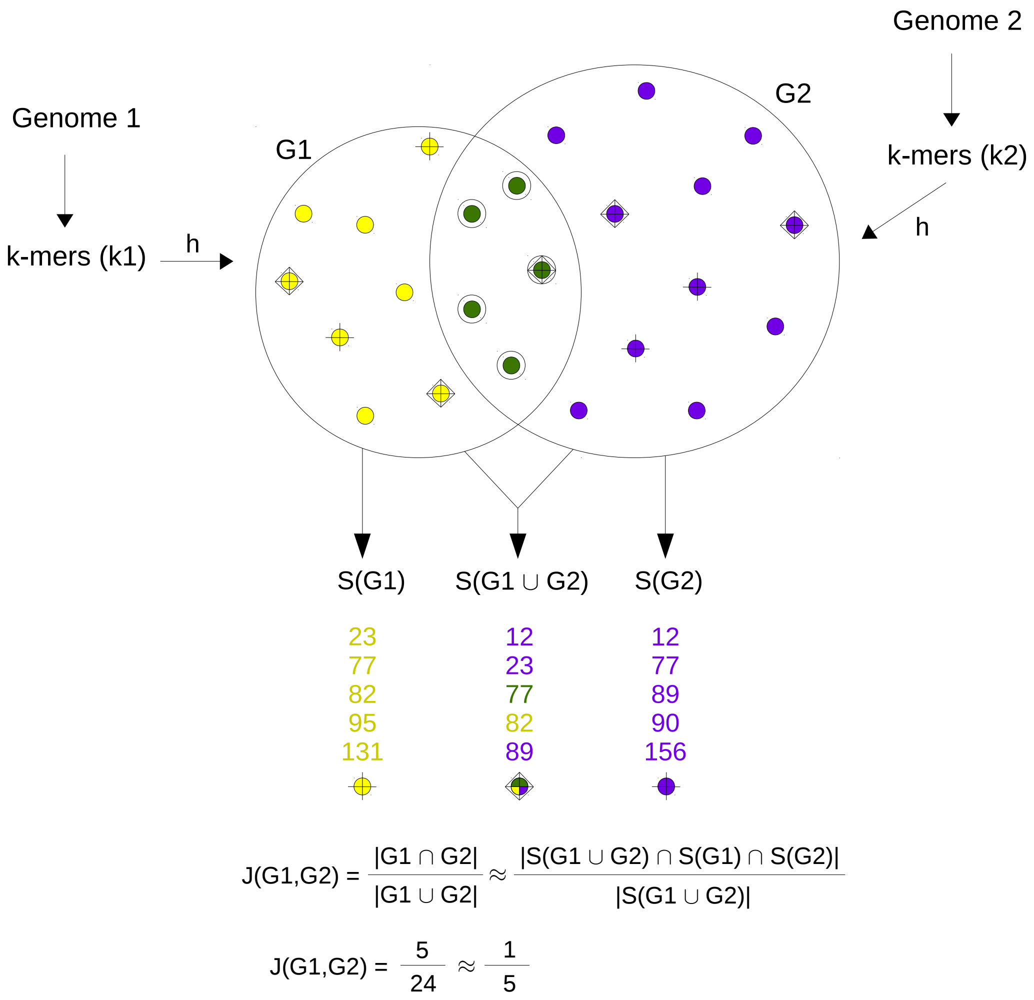


Figure 3.5: *Overview of the MinHash technique for estimating the Jaccard index ($J(G1, G2)$).* The two hash sets $G1$ and $G2$, contains $|G1|$ and $|G2|$ distinct hashes (yellow circles $G1 \setminus (G1 \cup G2)$ and blue circles $G2 \setminus (G1 \cup G2)$). Double green circles represent the common hash value between $G1$ and $G2$ ($G1 \cap G2$). The Jaccard index estimation is done by recovering only the five smallest hash values for $G1$, $G2$ (circles with crosses) and $S(G1) \cup S(G2)$ (circles with crosses and diamond). Redrawn from [Ondov *et al.* 2016].

Proximity of *Streptococcus thermophilus* strains

We have studied the proximity between strains of *S. thermophilus* both at the gene level and at the finer genomic content level.

Gene level

At the gene level, we have used the MicroScope [Vallenet *et al.* 2020] platform (https://mage.genoscope.cns.fr ; access date May 2020, v3.14.0) to annotate the genome of 77 *S. thermophilus* strains and calculate their pangenome. MicroScope uses the AMI-Gene software [Bocs *et al.* 2003] to predict protein-coding genes and the results of more than 20 methods are then used to assign a molecular function to these genes [Vallenet *et al.* 2020]. MicroScope calculates the pangenome, which is separated into three sets of genes corresponding to the persistent genome, shell genome and cloud genome. The pangenome is based on the MicroScope gene families (MICFAM) computed with the SiLiX software [Miele, Penel, and Duret 2011], which uses a single linkage clustering algorithm on homologous genes sharing an amino acid alignment coverage and identity greater than 80%. We then manually standardised the gene names of the pangenome families for all strains and created a $Strains \times Genes$ matrix (see figure 3.6).
Using the information from this matrix we produced a biclustering of strains and genes using formal concept analysis (FCA) [Ignatov 2015] computed with Answer Set Programming (ASP), a declarative programming language implementing constraint programming [Gebser *et al.* 2012]. In FCA a concept can be defined by its intent and its extent: the extent is the set of objects that belong to the concept (in our case the strains) while the intent is the set of attributes shared by these objects (in our case the genes of these strains). The idea is to search for maximal (S, G) biclusters (concepts), where S is a subset of strains and G a subset of genes present in all strains of S. A maximal bicluster is such that adding an element to one set cause the deletion of some elements in the other set. For example in figure 3.6, the bicluster $(\{Strain1, Strain2, Strain3\}, \{Gene1, Gene3, Gene4\})$ is maximal because the addition of $Strain4$ will cause $Gene3$ and $Gene4$ to be lost. Reciprocally, adding $\{Gene5\}$ to the bicluster need for instance to delete $\{Strain1, Strain2\}$. Among the maximal biclusters is the association of the complete set of strains within the core genome genes (e.g. $Gene1$ in figure 3.6) and the association of one strain with its strain-specific genes (e.g. $Gene2$ in figure 3.6). These genes are directly provided by MicroScope after the computation of the pangenome. Therefore, the maximal biclusters interesting to compute are those containing a number of strains greater than 1 and less than n, with n the number of strains studied. An example of these biclusters are represented on the right lattice of figure 3.7. They correspond to the concepts that do not contain all the intents, with a number of intents greater than 1 from which we remove the non-specific extents.

	Strain1	Strain2	Strain3	Strain4
Gene1	X	X	X	X
Gene2		X		
Gene3	X	X	X	
Gene4	X	X	X	
Gene5			X	X

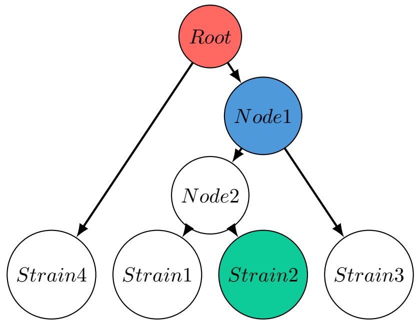


Figure 3.6: *Biclusters in a Strains* $\times$ *Genes matrix and associated labeling of nodes in a classification tree.*

At the genomic content level, we have also generated a strain classification tree with Microscope, which proposes a neighbour-joining algorithm with pairwise genomic distances obtained from the Mash software [Ondov *et al.* 2016] (see figure 3.13 at the end of this chapter). We used the knowledge of maximal biclusters to label some of the nodes and explain the proximity of *S. thermophilus* strains in terms of gene context. For this purpose, a node is represented by the set of its child leaves (strains) and, among the maximal biclusters, the genes (extent) corresponding to this set of strains (intent) are associated to this node. Therefore the leaves of the tree contain the strain-specific genes and the root of the tree contain the genes of the core genome. However, the lattice structure of maximal biclusters and the tree structure of strain clusters are not necessary compatible. Some nodes may contain no genes at all because, among the maximal biclusters there is no gene set corresponding to the set of strains in that node. In figure 3.6, this is the case for node2 because any characterisation of it would also cover $Strain3$. Reversely, there are maximal biclusters that cannot be represented by a node of the tree because they follow a different topology. For example $\{Strain3, Strain4\}$ is not present in the tree despite the fact that it can be uniquely characterised by the presence of $Gene1$ and $Gene5$. We chose to just display the compatible clusters of strains, which correspond to clusters with a strong support since they can be fully characterized both by their gene and their k-mer content. This is achieved through a dedicated interface, the iTOL tool [Letunic and Bork 2019]. Interactive Tree Of Life (iTOL) is an online software for displaying, annotating and managing phylogenetic or clustering trees. The maximum biclusters corresponding to the nodes are displayed interactively via popups associated with each node. The iTOL tree containing the 77 *S. thermophilus* strains is available here `https://itol.embl.de/tree/1312541346713115979255855`.

The complete list of concepts can be found here: `https://raw.githubusercontent.`

com/gsiekaniec/GeneTree/master/Streptococcus_thermophilus/Maximal_bicluster.txt and https://raw.githubusercontent.com/gsiekaniec/GeneTree/master/Streptococcus_thermophilus/CoreAndSpeGenomeBicluster.txt.

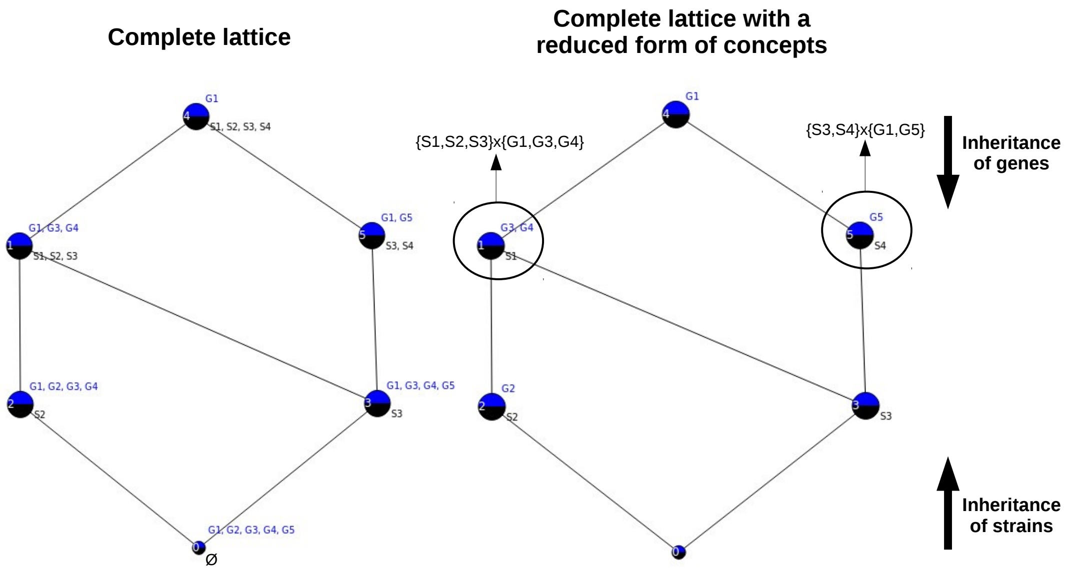


Figure 3.7: *Concept lattice based on the example presented in figure 3.6.* $\{G_i\}_{i=1..5}$ represent the different genes and $\{S_i\}_{i=1..4}$ represent the different strains. In the reduced form, each element appears just once in the lattice and one can retrieve concepts by propagating these elements in the direction of inheritance.

Whole genome level

At the genomic fragment level, several metrics were used to estimate the similarity between two genomes.

The first is the average nucleotide identity (ANI) in their shared coding regions via the use of FastANI (ManyToMany mode). FastANI [C. Jain, Rodriguez-R, *et al.* 2018] allows whole-genome alignment-free calculation of average nucleotide identity (ANI) and avoids costly sequence alignments by using Mashmap [C. Jain, A. Dilthey, *et al.* 2017], a

fast local aligner based on MinHash (see figure 3.5), to efficiently calculate orthologous mappings and alignment identity estimates.
In addition to the ANI distance calculation, two other genome-wide distances were used, the Jaccard (J) and Hamming (H) distances. In our case we have estimated, distances J and H on the set of q-grams of the genomes. The Jaccard distances on the set of 77 *S. thermophilus* strains are shown as a heatmap of a Jaccard distance in figure 3.8. Two genomes, one from *Streptococcus macedonicus* (different species) and the other from *Lactobacillus delbrueckii* subsp. *bulgaricus* (different family), were added as an outgroup. The diagonal represents the Jaccard distance of a strain against itself, which is a distance of 0 (very dark blue). We made three main observations on this heatmap. (1) The outgroup containing a different species and family is easily distinguished from the *S. thermophilus* strains with the Jaccard distance (red rows and columns on the edge of the heatmap). (2) Large dark blue squares appearing around the diagonal show clusters of strains that are close to each other. The white squares with a number represent clusters of particularly close strains that will be grouped together and considered as a single strain (see further section 3.2.2). (3) A group of strains composed of 6 CIRM-BIA strains (white cross on the heatmap marked with a star) is separated from the other strains, forming a new subgroup of strains isolated mostly from traditional Italian dairy products.

So far, we have considered three distances, the Jaccard and ANI distances and the Mash distance used in the MicroScope classification procedure. Are these different measures coherent and do they have the same resolution power ? To compare them, we have first extracted from the tree (Newick format) the corresponding distance matrix, using the dendropy python package [Sukumaran and Holder 2010]. The observed Pearson correlation between this matrix and the ANI/Jaccard distance matrices are very good ($r = 0.967$, $p - value = 1e - 04$ and $r = 0.987$, $p - value = 1e - 04$ respectively for the ANI and Jaccard distances). Overall, the three measures are close enough that any of them can be used without affecting the results of bacterial strain classification.

Thereafter, as the genomes are indexed in Bloom filters, we decided to test a distance that is tailored to this data structure for a more direct and thus more efficient computation of its value. We chose to measure the Hamming distance between the filters. The Hamming distance H is calculated by dividing the number of different positions between the Bloom filters of two genomes by the total number of positions (see equation H).

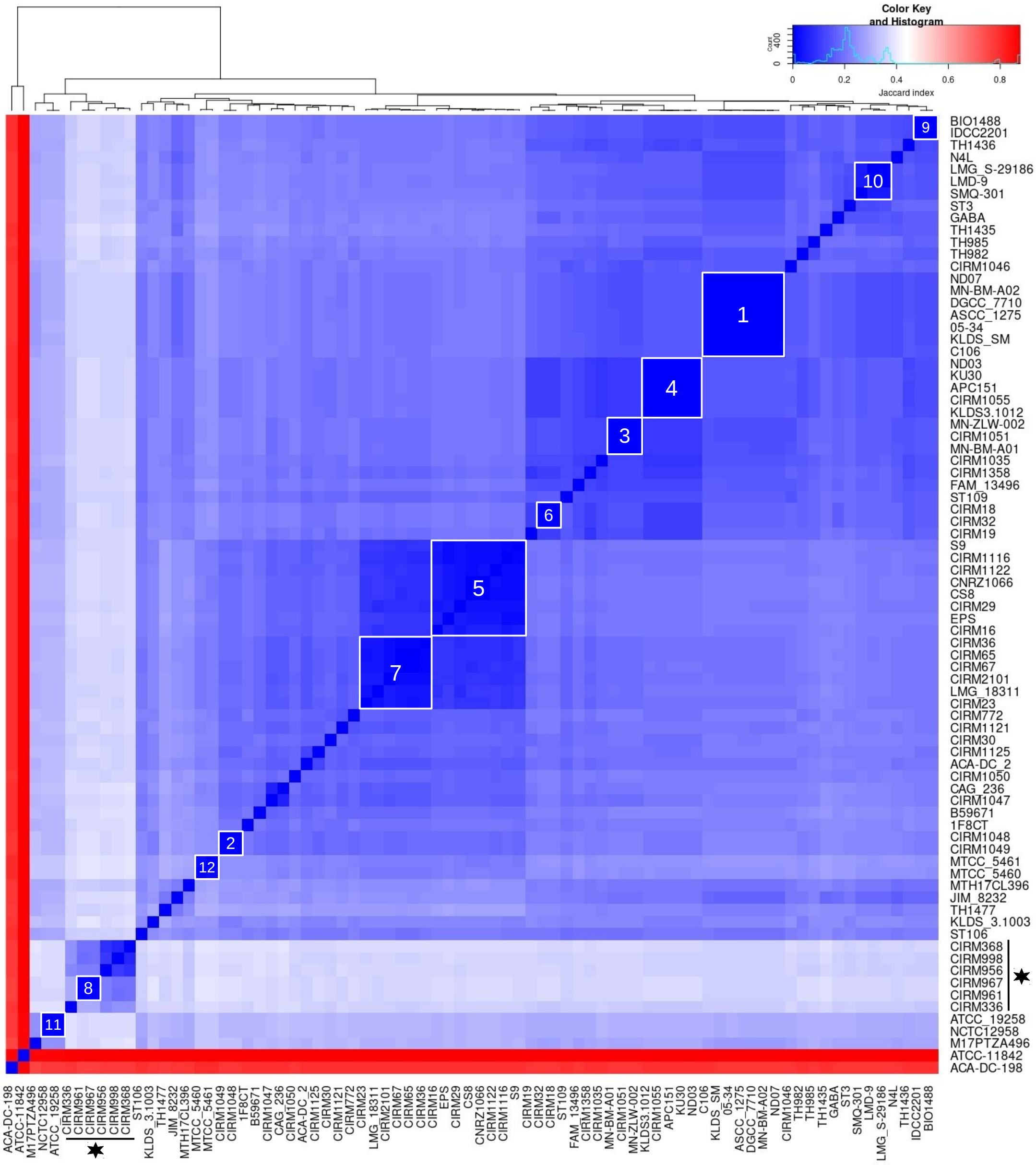


Figure 3.8: *Heatmap of the Jaccard distance for 77 S. thermophilus strains + S. macedonicus ACA-DC 198 + L. delbrueckii subsp. bulgaricus ATCC 11842. The clusters of table 3.3 are represented by white squares numbered from 1 to 12.*

$$H(BF1, BF2) = \frac{\text{Number of positions where BF1 and BF2 are different}}{\text{Total number of positions}} \tag{H}$$

where $BF1$ and $BF2$ correspond to two Bloom filters of the same size representing two genomes of bacterial strains.

As for the Mash distance, we evaluated the correspondence between the Hamming distance and the Jaccard distance for strain comparison. The correlation between the two distances is fine ($r = 0.99$, $p{-}value = 1e{-}04$). It is also possible by quadratic regression to determine the equation $H = 3.81e^{-03}(J^2 + J)$ between the Hamming distance H and the Jaccard distance J (see figure 3.9). For small values ($J < 0.15$), a linear relationship $H = 4.26e^{-03}J$ also fits very well. However, these equations are only correct for the BFs of size 5.10^8 bits that we used. Indeed, as the ratio is on the total number of positions, the Hamming distance depends on the size of the BFs. Theoretically, if the number of different positions remains the same, the larger the filter, the smaller the value of the Hamming distance. However, as we work with BFs, it is not that simple because the number of different positions is dependent on the false positive rate of the filter, which is itself dependent on the size of the filter. Generally speaking, the distances calculated from BFs (H or J) depend on the false positive rate and therefore on the size of the filters if the genomes compared do not change. The smaller the filter the more the distance obtained will be biased by the false positive rate.

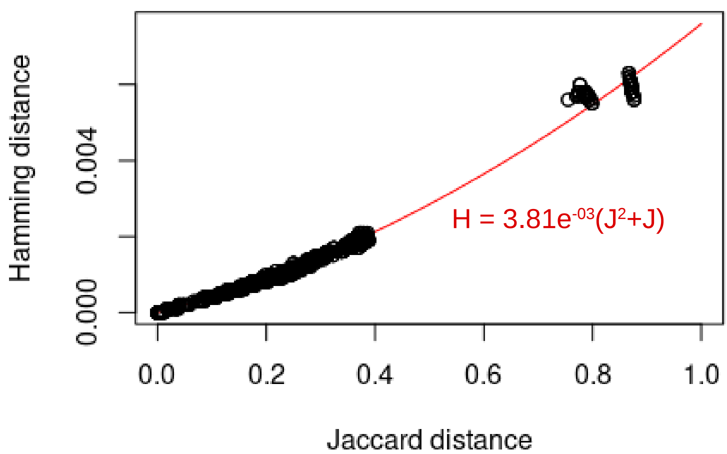


Figure 3.9: *Correlation between Jaccard and Hamming distances with a quadratic regression curve.*

3.2.2 Strain identification as an optimization problem

From a set of reference bacterial strains and the reads of a sample, the goal of identification is to find a subset of strains that best explain the reads. To do this, reads are assigned to their potential strains then identification is realised by using the $Reads \times Strains$ matrix obtained from the previous assignment.

Preprocessing the reads

As we work with long nanopore reads that can have a high error rate, we first select high quality reads by keeping only those:

- with a length greater than 2000 bp and an average Phred score greater than 9. The Phred score Q is a basecaling quality score that simply reflects the probability of error P in the sequence: $Q = -10log_{10}(P)$.
- in which at least 50% of the q-grams are found in one at least on strain.

This first preprocessing of the reads allows to keep the reads of better quality and to avoid assigning them to genomes that are too distant (e.g. other species/genus).
The probability of error can be calculated from a quality score given by:

$$P = 10^{\frac{-Q}{10}} \tag{I}$$

A Phred score of 9 is therefore associated with an error rate of $\sim$ 12.6%. However it has been proven that this score is in practice different from the real error rate [Delahaye and Nicolas 2021]. If we use the equation $E = 0.042Q^2 - 2.68Q + 43.92$ provided by [Delahaye and Nicolas 2021] we rather have a filter of reads with an error rate higher than $\sim$ 23.2%.

In order to exclude uninformative reads and thus reduce the complexity of the problem by reducing the space of possible solutions, a second quick pre-processing step was applied to filter out reads and strains that provided little information. Thus, reads found in too many strains were not taken into account during the identification (by default, if a read can be affiliated to 18 or more strains, they are considered ubiquitous and not useful for identification). Furthermore, only the n best strains (12 by default) are considered for each remaining read. The best strains are recovered according to the proportion of their q-grams in the read. They are then ranked for each read according to this q-gram proportion.

The identification problem

We then present the identification problem as an optimization problem seeking to account for the data by retaining the minimum number of necessary strains. It is thus a question of exploiting a hypothesis of parsimony.
From a theoretical point of view, this identification issue can be seen as a set cover optimisation problem whose aim is to find a minimal subset of strains to explain the reads completely. This is one of Karp's NP-complete problems and people usually limit themselves to an approximate solution via a statistical approach (e.g. [Yang *et al.* 2019]). But it is possible to search for an exact solution of moderate size (e.g. a dozen of strains in the sample) using modern combinatorial solvers. During the thesis, the Answer Set Programming (ASP) declarative problem solving approach was used to resolve this issue. During the identification of the strains, a pre-processing is done to select the candidates strains and to discard the marginal strains. A strain is considered as marginal if:

- less than 1% of reads are assigned to it.
- its weight (sum of its ranks) is less than 2% of the total weight (sum of the ranks of all the strains).

This preprocessing allows to ignore strains that are not present enough in the assignment results or often assigned but not as well as another strain.
Then the strains present in the sample are selected according to the following constraints:

- each read must be assigned to a selected strain.
- a selected strain must have at least one read specific to it if we consider only the selected strains.

The identification is done by (1) minimizing the number of selected strains and (2) maximizing the number of assigned reads and reads specific to each selected strain.

3.2.3 Clustering strain genomes before identification

The proximity of the *S. thermophilus* genomes makes taxonomic assignment of the strains difficult, and the use of erroneous reads makes it even more difficult. Indeed, comparison of the 16S-23S genomic regions of *S. thermophilus* strains showed a maximum of 11 divergence mutations over a length of about 1420 bases for ST106 vs. CIRM-BIA967 strains and most have no difference. Since these strains have very similar core genomes, it is difficult if not impossible to differentiate them, even using MLST [Junjua *et al.* 2016]. If we look at the heatmap in figure 3.8 we can see that some strains are extremely close, even if taking into account the whole genome sequences. One may wonder whether these genomes should be considered as two different strains or not. We propose to delimit a practical threshold from the point of view of sequence-based identification. In this way, the biologist knows in advance the group of strains that would require other methods for their discrimination. Of course, these groups should be as small as possible. An additional advantage of defining a boundary is to identify isolates that potentially have an identical phenotype. We call these very similar strains *sibling strains*. The identification software we have developed (see section 4.2) allows to consider them as a single entity to improve the accuracy of the detection.

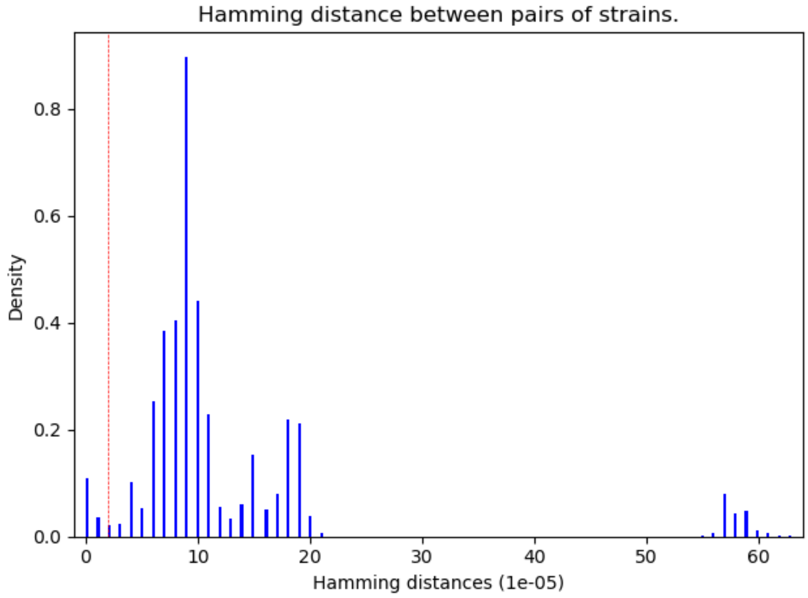


Figure 3.10: *Histogram representing the distribution of Hamming distance between pairs of S. thermophilus strains.* The distances observed on the right side of the histogram are due to the presence of a strain of *S. macedonicus* and a strain of *L. delbruecki* subsp. *bulgaricus* in the index. The vertical dotted red line represents the threshold for grouping the sibling strains

The sibling strains are defined with respect to a threshold θ for the maximum distance between their genomic sequences. As the choice of the type of distance used does not seem crucial (all distances are consistent with each other), we used the Hamming distance. When the Hamming distance threshold is set to 0 ($\theta = 0$), this means that all strains are considered as distinct isolates. By default, the threshold was set to $\theta = 2e^{-4}$

(0.005 for Jaccard) and used to group the sibling strains present in the 77 strains of *S. thermophilus* with a Bloom filter of size 5.10^8. This threshold was determined empirically by testing different thresholds and taking the minimum threshold that gave the lowest identification error rate. Subsequently in the identification software we have developed (see section 4.2), an histogram representation of the Hamming distance distribution between strains is provided to the user. This representation allows to determine a coherent threshold to group the sibling strains. It is shown in figure 3.10 for the *S. thermophilus* genomes.

Once this threshold is set, the groups of sibling strains are formed in the following way:

1. An undirected graph that we call the sibling strains graph is created with the strains as vertices and edges connecting the strains whose Hamming distance is lower than θ. The sibling strains graph created for the *S. thermophilus* genomes with $\theta = 2e^{-4}$ is presented in figure 3.11.

2. Finally, the Bloom filters of the genomes belonging to the same connected component of the graph are merged, thus recovering the pangenome of these related strains in a single filter. The different clusters obtained with the *S. thermophilus* strains are shown in table 3.3 and visible in the heatmap in figure 3.8 and in the figure 3.11. Almost all are dense cliques or pseudocliques, a strong indication that the chosen threshold provides a consistent view of equivalence classes across strains.

Table 3.3: *Clusters of S. thermophilus sibling strains.* Cluster numbers also appear in the heatmap figure 3.8.

Cluster id	Strains
1	05-34 ; ASCC_1275 ; DGCC_7710 ; KLDS_SM ; MN-BM-A02 ; ND07 ; C106
2	CIRM1048 ; CIRM1049
3	CIRM1051 ; MN-BM-A01 ; MN-ZLW-002
4	CIRM1055 ; APC151 ; KLDS3.1012 ; KU30 ; ND03
5	CIRM1116 ; CIRM1122 ; CIRM16 ; CIRM29 ; CNRZ1066 ; CS8 ; EPS ; S9
6	CIRM18 ; CIRM32
7	CIRM2101 ; CIRM23 ; CIRM36 ; CIRM65 ; CIRM67 ; LMG_18311
8	CIRM961 ; CIRM967
9	IDCC2201 ; BIO1488
10	LMD-9 ; LMG_S-29186 ; SMQ-301
11	ATCC_19258 ; NCTC12958
12	MTCC_5460 ; MTCC_5461

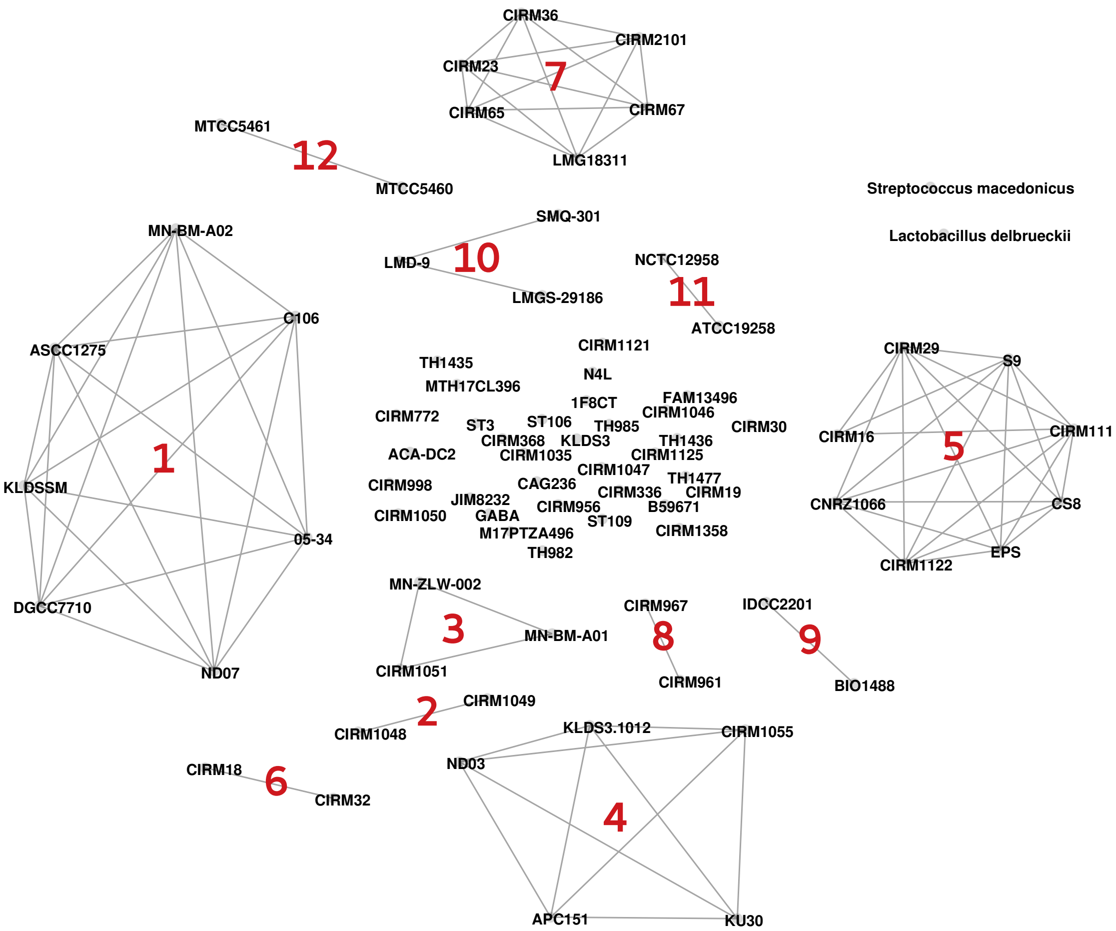


Figure 3.11: *sibling strains graph representing 77 S. thermophilus strains (plus one strain of S. macedonicus and one strain of L. delbruecki subsp. bulgaricus) created with* $\theta = 2e^{-4}$. Vertices represents strains and edges connect the strains whose Hamming distance is lower θ. The clusters created from this graph and present in the table 3.3 are represented by red numbers from 1 to 12

Genome inclusion, a special case in genome proximity

When merging closely related strains, we were quickly confronted with strains so close to each other that the question "are some strains included in others?" has arisen.

The calculation of the inclusion between two sequences is different from the calculation of the distance between these sequences. Since it does not necessarily mean a small distance between the two sequences.

The problem arising from this genome inclusion is that with our method, a genome

totally included in another one could not be identified because it contains no specific q-grams. To make sure that no *S. thermophilus* genome were totally included in another one, we compared the initial Bloom filters by computing the percentage of position 1 of one filter found in the other filter. If the value is equal to 100% then the q-grams of the first genome are entirely present in the q-grams set of the other genome. There are two possibilities for this to happen, either the genome is entirely included in the other genome, or it varies only by the number of occurrences of each q-gram (which is not taken into account in the Bloom filters). In both cases, a fine representation of genomes would be necessary such as variation graphs for example (see section 2.1.3).
In our case, some of the *S. thermophilus* genomes are more than 99% included in another one, i.e. the *S. thermophilus* strain CIRM67 is 99.9% included in *S. thermophilus* CIRM65 but no strain was totally included (100%) in another one. However, this means that, for example, to identify CIRM67 from CIRM65 we need to use the 0.1% differences between the two Bloom filters which makes the identification quite difficult.

Perspective to improve the sibling strains clustering

The clusters could be less coherent than for *S. thermophilus* strains. This is the case if the connected component is made of several overlapping cliques sharing very few edges. For example, case 5 in figure 3.12 has two cliques of size 6 and 7 overlapping on a single node. Since the two groups of sibling strains have only one common strain, it makes more sense to separate them than to group them together. For this purpose the minimum cut (min-cut) can be used. This min-cut problem consists in cutting a minimal number of edges to split the connected component.
To determine in which case to merge the overlapping cliques and in which case to separate them it is possible to set a threshold taking into account the level of overlap of the two cliques. For this purpose different measures can be used, one based on the number of nodes and another on the number of edges of the sub-graph composed of the two overlapping cliques. In any case, if more than two cliques overlap, cliques are merged (or separated) in an iterative way, starting with the largest cliques.

The first measure would be a simple ratio R of the number of nodes in the sub-graph divided by the number of nodes in the two separate cliques:

$$R = \frac{n}{n_i + n_j} \tag{J}$$

with n_i the number of nodes of the first clique, n_j the number of nodes of the second clique and n the number of nodes in the union of the nodes of the two cliques $N_i \cup N_j$. If R is greater than a threshold Θ, the two cliques are merged. Otherwise the cliques are separated using the minimum cut.

The second measure would be the *clustering coefficient* which allows to estimate the completeness of a graph. Examples of clustering coefficient calculation for different graphs are shown in figure K. In our case we considered the sub-graph composed of the two overlapping cliques. It is calculated by dividing the proportion of edges existing in this sub-graph by the number of edges of a complete graph containing the same number of vertex. As the number of edges of a complete graph with N vertices is $\frac{N \times N-1}{2}$, the clustering coefficient is computed as follows (see figure 3.12 and equation K):

$$Clco(N_i, N_j) = \frac{2 \times |E_{N_i} \cup E_{N_j}|}{|N_i \cup N_j| \times (|N_i \cup N_j| - 1)} \tag{K}$$

with N_i the nodes of the first clique, N_j the nodes of the second clique and $E_{N_i} \cup E_{N_j}$ the union of the edges of both cliques.
If this clustering coefficient is lower than a threshold β, the two cliques are merged. In the opposite case, the cliques are separated using the minimum cut.

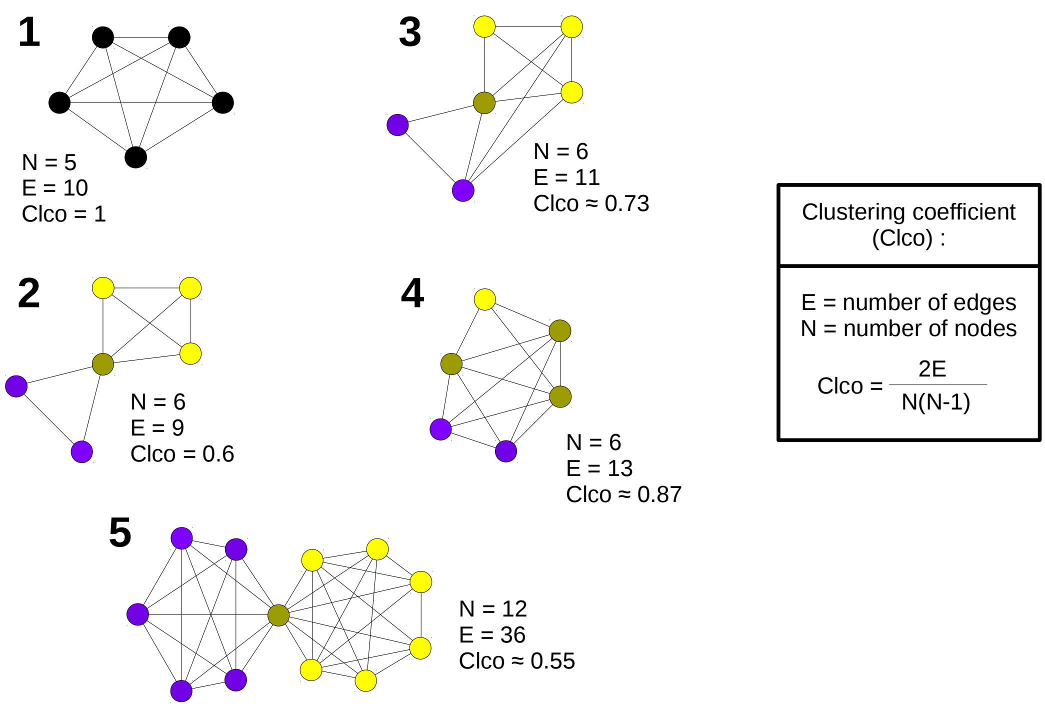


Figure 3.12: *Computation of the aggregated clustering coefficient (clco) for five graphs representing a clique overlap.* Graph 1 being complete, the clco is equal to 1. The equation use to compute the clustering coefficient is presented in equation K.

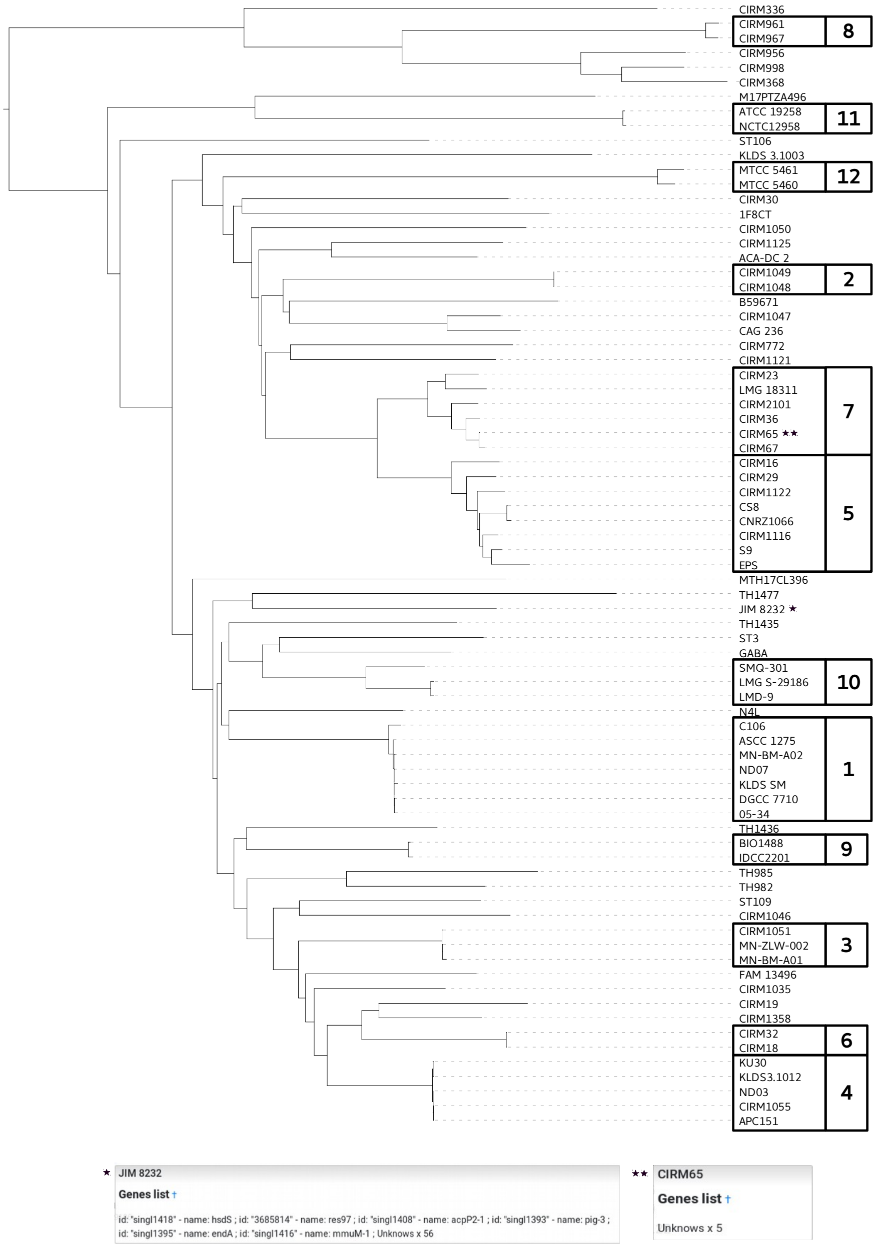


Figure 3.13: *Representation of the iTOL tree containing the 77 S. thermophilus strains available at* `https://itol.embl.de/tree/1312541346713115979255 85`. The list of genes present in *Streptococcus thermophilus* strains JIM8232 and CIRM65 are shown for information only, these lists appear interactively as a popup on the iTOL tree. The clusters of the table 3.3 are represented by black boxes from 1 to 12.

Chapter 4

ORI, a new software for strain identification from long reads

Contents

Preamble: This chapter presents the creation of a new tool to identify bacterial strains from long Oxford nanopore reads. The development of this tool is based on the spaced seeds, the strain identification method and the clustering of sibling strains presented in chapter 3. This chapter emphasizes the difficulty of indexing many bacterial genomes and the choices we made to move to a large genome bank while using a low memory footprint. Our approach is based on an extension of the HowDeSBT index.

4.1 The difficulty of indexing many bacterial genomes

In recent years, the number of genomes of strains in the databases became very large, the technology improvements speeding up the number of sequenced samples every year (see section 1.1.2 and figure 1.3). As an example there are currently (in September 2021) more than 25 000 genomes of *Escherichia coli* in the National Center for Biotechnology Information (NCBI) database.
This makes it possible to improve the precision of bacterial strain identification. However, it complicates the development of identification software that must scale up to a large data bank size. This has been the case for metagenomic data, but it is also becoming the case for simpler samples made of one or a few strains. In our work, we added the constraint of being able to run the program on a standard laptop. For example, to retrieve which strain is present in a sample of *E. coli*, it is necessary to store at least the 2063 known complete genomes, which requires a minimum of about 10 Gb for the sequences alone. If we consider the complete genomes of the order *Lactobacillales* (25469 genomes), the raw data represent more than 55 Gb. These data would not fit in a standard laptop with 8 Gb of RAM (Random Access Memory). Even a computer with 32 Gb of RAM would not be able to process all the *Lactobacillales* without specific data processing.

We start thus this thesis by looking for an index that would highly compress the data and use a reasonable amount of memory during the query. We reviewed in chapter 2, several methods to index the genomes in order to use as little memory as possible while allowing to efficiently query the data. We chose to start from the k-mer data structure based on Bloom filters implemented in HowDeSBT (see section 2.1.2) since it was the structure that used the least amount of space to store the data at the time of our review.

A version of HowDeSBT specific to strain identification

The general functioning of HowDeSBT has already been presented in section 2.1.2. We have adapted the method to the context of bacterial strains identification from long erroneous reads. The modifications made are the following:

- To be less sensitive to ONT read errors, a spaced seed is used to extract q-grams from the bacterial strain genomes to be indexed instead of k-mers. The indexing of q-grams is shown in 1.

Algorithm 1: Indexing q-grams from a genome

```
Data: genome G containing n k-mers of size k
Result: Bloom filter containing q-grams
for kmer in G do
    // Canonical representation of k-mers
    rev = reverse(kmer) // compute the complement of the k-mer
    canonical = min(mask(kmer), mask(rev))
    h = hash(canonical)
    insertInBF(h)
end
```

The complexity the q-gram indexing is globally the same as k-mer indexing, $O(n)$ with n the number of k-mer occurrences in the genome G. There is an extra cost in the number of operations to compute q-grams, for the masking using the spaced seed pattern. It increases the number of operations from $n(k+4)$ to $n(k+6)$, a minor constant factor.

In the same way, when querying the index, the q-grams are extracted from the reads instead of k-mers. Actually, during this query, the complete file containing all the reads is loaded in memory which allows to make the request faster by querying all the reads at the same time, going down the index tree. The spaced seed used during our experiment is the one presented at the end of the section 3.1.2.

- We added the computation of the Hamming distance matrix between strains using the Bloom filters of the index. These step requires a lot of time and memory.
  Indeed, all pairs of BF have to be compared. The computation of the Hamming distance between two filters requires to traverse the filter obtained through a logical operation (XOR) on them. The global time complexity is therefore $O(n^2m)$ with n the number of BFs and m the size of a filter. Regarding the memory, in order not to lose too much time loading and unloading the filters in RAM, they are all loaded at once, which implies a memory complexity in $O(nm)$.
  In order to reduce this memory usage it is possible to implement hybrid approaches where only part of the filters are loaded in memory before unloading them to load another one, shifting the memory/time trade-off to even longer processing.
  In order to scale up to the whole set of complete *E. coli* genomes (see section 5.3), we chose another approach. We assumed that the hash function used to fill the BFs was uniform which, means that the values are uniformly distributed over a

BF. It is then possible to estimate the Hamming distance between two BFs by using only a small part of them.

In order to check the quality of this estimation, we compared first the distances on *S. thermophilus* data (on which the complete computation of the distances was performed). The ordering of values was almost perfect when using 50% of the BFs. We then made the test on a larger dataset containing 100 *E. coli* genomes (on which the complete computation of distances was also available). It was possible to decrease the portion of the filters used to less than 1% without modifying the clusters of sibling strains formed by our method. More cautious reductions are already interesting and sufficient in most cases: using 50% of the filters reduces processing time and memory by two.

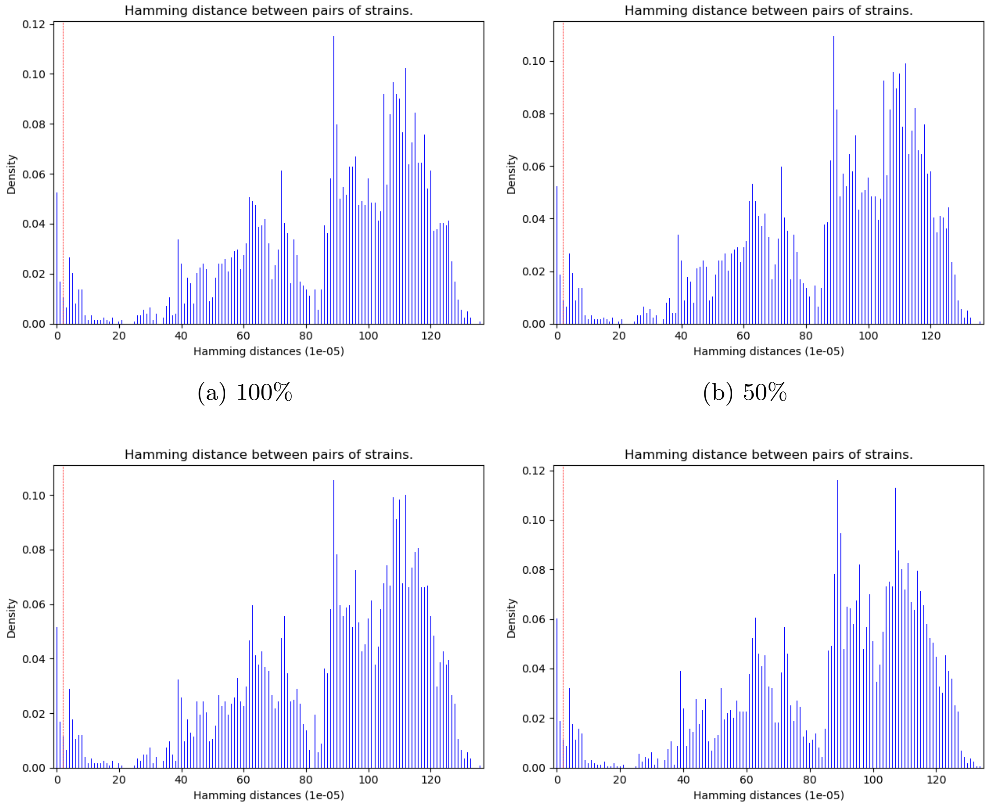
Overall, there is relatively few changes in the distances between 50 or 25% filter use and full use (100%; see the distributions of distances in figure 4.1). We have measured the differences between Hamming distance distributions with the Jensen-Shannon distance JS, a normalized symetric score base on entropy (Kullback-Leibler divergence). The JS distance ranges from 0 (identical) to 1 (maximally different). The results are the following: $JS(50||100) \approx 0.00119$, $JS(25||100) \approx 0.0015$ and $JS(1||100) \approx 0.0039$. As can be seen, the discrepancy remains very low and only starts become noticeable after a significant compression to 1%.

The sibling strains determination is done as explained in section 3.2.3 together with the merging step of the strain BFs.

4.1.1 Data compression using our version of HowDeSBT

We present in table 4.1 some statistics on the extended HowDeSBT indexes we have built on various genomes. The different datasets are presented in order of increasing size. Index 1 is the set of complete genomes of *Streptococcus thermophilus*, to which are added the complete genomes of *Lactobacillus delbruecki* for index 2. In index 3 the complete genomes of *Bifidobacterium* are added then those of *Leuconostoc*, *Lactococcus* and *Enterococcus* in index 4. The last index contains all complete genomes from *Escherichia coli*.

As expected, our data structure offers a high compression rate, higher than 85% in all cases and higher than 90% in 3 cases. In their article, [Solomon and Kingsford 2018] obtained similar compression rate on their RNA-seq dataset. The lower compression level in experiments 3 and 4 can be explained by the fact that these experiments contain more distant genomes than others. Indeed, the data compression in HowDeSBT depends on



(a) 100% (b) 50%

(c) 25% (d) 1%

Figure 4.1: *Distribution of Hamming distances between 100 E. coli strains obtained using different percentage (from 100% to 1%) of their Bloom filters.* The differences remain small and noticeable only for the smallest 1% and high distance values. The Jensen-Shannon distances (using a base-2 logarithm) between the distribution using full Bloom filters and the other distributions are as follows: $JS(100||100) = 0$, $JS(50||100) \approx 0.0012$, $JS(25||100) \approx 0.0015$ and $JS(1||100) \approx 0.0039$. The red lines correspond to the threshold set to group the sibling strains together.

the tree structure determined by agglomerative hierarchical clustering using Hamming distances between BFs (see section 2.1.2). Thus, the compression increasing with the proportion of common k-mers (or q-grams in our case) in the close indexed genomes. These common q-grams will be stored only once in nodes of the BF tree. On the contrary, a common q-gram between two distant strains will be stored twice. As a result, close genomes will be better compressed than more distant genomes. This index is therefore well adapted to the representation of many strains of the same species.

Table 4.1: *Indexation of different complete genome datasets using HowDeSBT.* The sequence size column represents the total size of the genomic sequences to be indexed. The BF size represents the size of a Bloom filters before compression and the index size is the size of the final index after compression. This BF size has been set taking into account the largest genome to be indexed and a requested false positive rate (less than 1% in this case). This size has however little repercussion on the final size of the index because the BFs are compressed to use as little space as possible. Note that the creation time is not a crucial parameter since it can be reached once for all applications made from the index.

ID	**Indexed genomes**	**Genome number**	**Sequence size**	**BF size**	**Index size**	**Compression rate**	**Max memory usage**	**Time (hh:mm:ss)**
1	*Streptococcus thermophilus*	95	171 Mb	0.21 Gb	1.3 Mb	99.24%	213 Mb	00:03:30
2	1 + *Lactobacillus delbruecki*	332	586 Mb	0.23 Gb	51 Mb	91.3%	260 Mb	00:12:50
3	2 + *Bifidobacterium*	1 026	2.1 Gb	0.33 Gb	299 Mb	85.8%	334 Mb	01:03:59
4	3 + *Leuconostoc* *Lactococcus* *Enterococcus*	1 298	2.7 Gb	0.33 Gb	404 Mb	85%	335 Mb	01:16:52
5	*Escherichia coli*	2 063	9.8 Gb	0.76 Gb	539 Mb	94.5%	941 Mb	04:12:24

4.1.2 Memory and time required to query an HowDeSBT index

Once the index created, it is important to evaluate the complexity of queries on this data structure. We have queried the different indexes created previously with three different sets of 1 000 reads from a species contained in the index (*S. thermophilus* for indexes 1,2,3 and 4 and *E. coli* for index 5). The average time and memory consumption for these queries are shown in table 4.2.

Table 4.2: *Query of our index on the complete genome datasets presented in table 4.1.* We used as queries three sets of 1 000 reads for each dataset. The results are the average time and memory used by the query. The three sets of 1 000 reads correspond to *S. thermophilus* JIM8232 (experiments 1,2,3 and 4) or *E. coli* (experiment 5).

ID	**Indexed genomes**	**Index size**	**Max memory usage**	**Query time (hh:mm:ss)**
1	*Streptococcus thermophilus*	1.3 Mb	141 Mb	00:08:18
2	1 + *Lactobacillus delbruecki*	51 Mb	146 Mb	00:09:28
3	2 + *Bifidobacterium*	299 Mb	163 Mb	00:12:55
4	3 + *Leuconostoc* *Lactococcus* *Enterococcus*	404 Mb	168 Mb	00:17:49
5	*Escherichia coli*	539 Mb	171 Mb	00:51:52

Memory usage

The maximum amount of memory used by our indexing process is slightly impacted by the index size. In fact, the memory used during the request depends on the size of the starting BFs and their compression rate because the index only loads one filter at a time into memory. Moreover, during the query, the largest part of the memory used is linked to the number of reads.

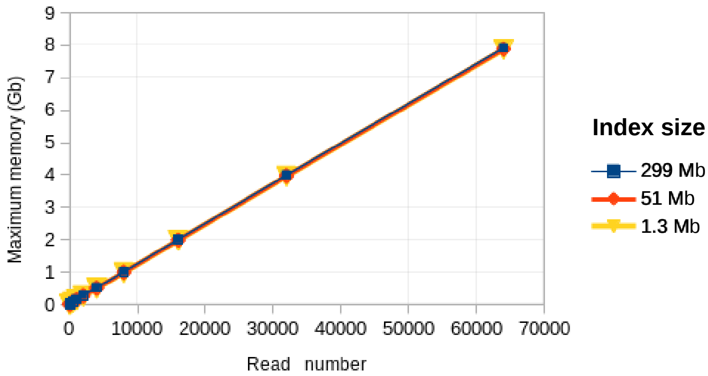


Figure 4.2: *Maximum memory usage for querying the first three indexes from table 4.1 of respectively 1.3, 51 and 299 Mb with an increasing number of reads from Streptococcus thermophilus JIM8232.* The maximum memory usage seems clearly to be a linear function of the reads number and independent of the index size.

Figure 4.2 shows that the maximum memory consumption during queries is a linear function of the number of reads ($f(x) \approx 0.0001x + 0.03$ in Gb). For queries that contain until 45 000 reads (corresponding to about 6 Gb of maximum memory usage), the maximum memory usage is not a problem even on a single laptop. In most practical contexts, it will be sufficient to query the content of a bacterial sample. As we will see in section 5.1 it is recommended in practice to use about 4 000 reads with our method because the noise brought by the high error rate of these reads makes the identification less accurate with too many reads. For the detection of rare events requiring the use of more reads (see section 5.1), the memory growth remains reasonable. However, as for other identification software, our identification method is not adapted to mixtures with a species present in very low abundance. Indeed, false positive identifications become very frequent in such a case due to the large core genome shared by many strains.

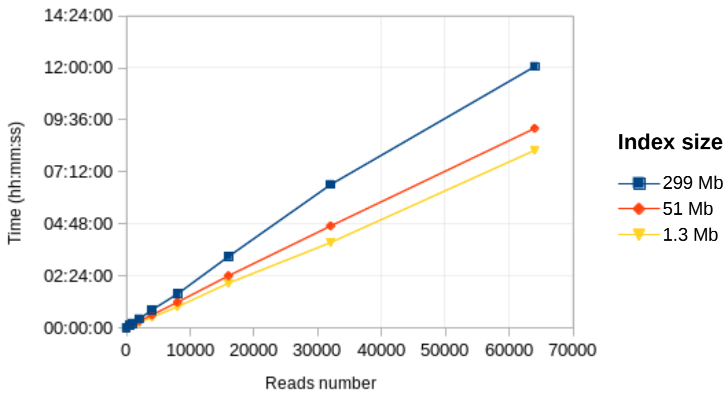


Figure 4.3: *Time usage for querying the first three HowDeSBT structures from table 4.1 of respectively 1.3, 51 and 299 Mb with an increasing number of reads from Streptococcus thermophilus JIM8232.* Time seems to be linear to the number of reads, as for maximum memory usage, but depends this time on the size of the index. The size of the reads might also slightly affect query time.

Time

The query time is impacted both by the size of the index (see table 4.2 and figure 4.3) and by the number of reads in the query. As we can see in figure 4.3, the computation time is a linear function of the number of reads in the request for a fixed size of index. A larger index increases the computation time because the larger the index the more likely the BF tree will be complex. In fact, an index containing the same 100 genomes would take up little space because all the q-grams will be common and present once in the root of the tree. In general, as the size of the index increases, the number of levels of tree increases and the query needs to seek at an increasing number of BF node.
To be complete, the computation time depends also on the requested reads themselves. As an example, a query with q-grams not present in the index will be processed very quickly because all the reads will be rejected at the root of the index tree.
By keeping the number of reads low, the processing time is acceptable. It takes for example less than one hour of computation to request up to 4 000 reads (our recommendation) with the index of 299 Mb containing more than 1 000 genomes (see figure 4.3). This query time remains however a limiting point of the method if one needs a fast identification of bacterial strains. We will see later in section 4.3 some ideas to speed up this query processing.

4.2 ORI: a new software for strain identification from long reads

We have integrated our ideas within a new bacterial strains identification software called Oxford nanopore Read Identification (ORI). Table 4.3 summarizes the main commands and parameters of ORI.

Table 4.3: *Main ORI's commands and parameters.*

Command name	**Command description**	**Parameter**	**Parameter description**
Index creation			
ORI length	Computes the size of the genomes and determines the size of the BFs to use	–false_positive_rate/-fpr	Maximum false positive percent wanted for the BF containing the largest genome
howdesbt makebfQ	Creates the BFs for each genome	–qgram	File containing the spaced seed pattern
		–bits	Size of the BFs
ORI threshold	Creates the histogram representing the distribution of distances between genomes	–threshold/-t	Threshold used to merge sibling strains
howdesbt distance	Computes Hamming distance between genomes and merge of BFs of sibling strains	–threshold	
		–merge	Merge the sibling strains cliques
Query part			
ORI suppr_bad_reads	Removes poor quality and small reads	–qualityMin	Minimum quality threshold to save a read
		–lengthMin	Minimum length threshold to save a read
howdesbt queryQ	Queries the index with the reads	–threshold	Fraction of the reads that must be present in a strain to be assigned to it
Identification part			
ORI identification	Identifies the strains present in the sample by using the read assignments	–nbchoices/-n	Number of best strains to consider for each reads + if a read is affiliated to 1.5n strains it is not used for identification

4.2.1 Creation of the index

The main component of ORI is an index of reference genomes stored in a compact HowDeSBT structure. The method can be divided into three steps: (1) the reference genome index creation, (2) the query part of the index and (3) the strain identification part (see figure 4.4). The ORI software is available on github at https://github.com/gsiekaniec/ORI and can be installed through a conda package [*Anaconda Software Distribution* 2020] using the following command:

```
conda install -c gsiekaniec -c conda-forge ori
```

As explained previously in section 4.1.2, the ORI index is created with an HowDeSBT version modified to use a spaced seed to extract q-grams from k-mers of reference genomes. In details, when indexing the q-gram of a genome, only one version of the two strands of the chromosome is saved. As the DNA molecule is made up of two complementary strands we used the canonical form of the q-grams (see previous section 2.1.2

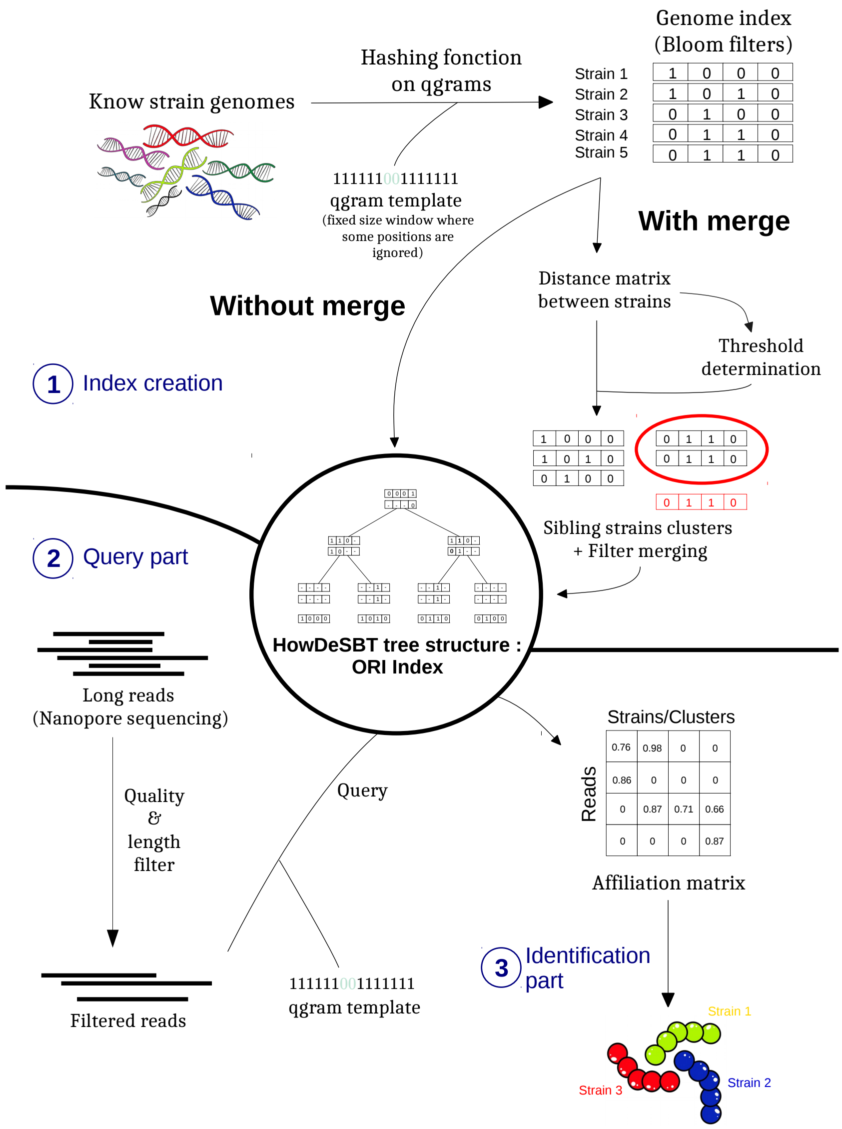


Figure 4.4: *ORI's pipeline divided into three parts: 1) index creation. 2) the query part and 3) the identification part.* In the workflow the user can trigger a merging step where s/he has to fix the value of a threshold parameter.

for further details on the canonical form of a sequence). To do that, the spaced seed is applied to the k-mer and its reverse complement before choosing the canonical q-gram.

Genomes selection

ORI can index complete genomes as well as draft genomes. However, the latter are not always of very good quality and can in turn reduce the quality of identification results. Indeed, genomes in the form of contigs are genomes whose assembly is not finished. These genomes may still be cut into many sequences, have a total size (number of base pairs) far from the real expected size, and contain chimeric contigs (sequences from two different locations of the genome merged into a contiguous one). Genomes in scaffold form are more elaborate. They contain less sequences and a total size closer to the expected genome size. However, these genomes will often contain a large range of N (indeterminate nucleotide). In our experiments the genomes used are complete genomes to which scaffolded and contiged genomes are sometimes added. In addition to the genomes, it is possible to recover the plasmid sequences present in the bacteria. As we worked on the identification of bacterial strains we decided not to take into account these plasmids. Indeed a plasmid can be transmitted from one bacterium to another and it seems quite hazardous to base a bacterial strain identification on the presence or absence of a plasmid. However, it is possible to make an ORI index specific to plasmids and to look for plasmids separately from the chromosome genomes. The creation of this index can be of particular interest in identifying specific functions such as antibiotic resistance (see section 1.1.2 for more details on plasmids).
The genomes were downloaded using a python script based on asynchronous programming. It is almost mandatory to efficiently download a large quantity of files such as all the *Lactobacillales* genomes for example, but it requires proper error handling to avoid stopping all downloads if one of them fails. We illustrate in figure 4.5 the advantages of asynchronous vs synchronous downloads.

On the determination of the Bloom filter size

The reference genomes are represented by their set of q-grams, using a spaced seed (section 3.1.2). These q-grams are hashed and then inserted into a Bloom filter of fixed size (5.10^8 bits for the experiments with *S. thermophilus* in section 5.1), one filter per reference genome. To fix the optimal size of the Bloom filters, ORI computes the size of the reference genomes and allows the user to fix the value of the desired false positive rate. By default, it is set to 1%. Then the BFs size is calculated via the formula A, applied on the total number of q-grams of the largest genome:

$$P_{false\_positive} \approx (1 - e^{\frac{-lm}{n}})^m \tag{A}$$

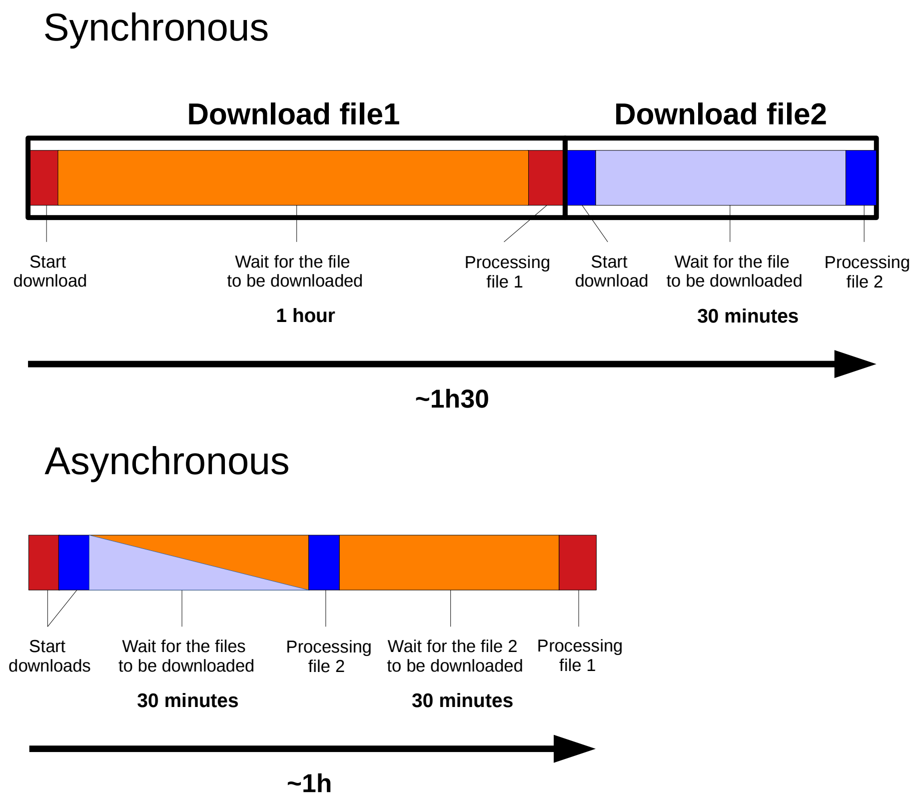


Figure 4.5: *Example of behaviour of synchronous vs asynchronous download strategies.* In this example, the time for launching the downloads and processing the downloaded files is considered to be insignificant. In asynchronous programming, the download time is dependent on the download time of the largest file and no longer on all the files.

where l is the number of inserted q-grams, n is the length of the BFs and m is the number of hash functions. In our case, $m = 1$. So the equation B to find n is

$$n \approx \frac{-l}{ln(1 - P_{false\_positive})} \tag{B}$$

Considering that each q-gram of the genome is distinct from the other one leads to overestimate the size of the BFs. This is however a minor issue since a BF that is less dense will be better compressed in the final index. Indeed, the basis of the compression algorithm (RRR) implemented in HowDeSBT is to encode the ranges of 0s in the most efficient possible way.

In the case where it is crucial to improve this BFs size estimation, we advise to use as a preprocessing step an algorithm called ntCard [Mohamadi, Khan, and Birol 2017], which allows to compute among other things the number of distinct k-mers of the genomes. It is then possible to no longer overestimate the size of the BFs. Note however that ntCard is based on k-mers and ORI works with q-grams which biases a bit the results. Moreover the estimation needs more processing time since it does not consider that all q-grams occurrences are different.

Merging the strains

Once the BF of each genome is created, the software proposes two possibilities, either the genomes of the sibling strains are grouped together before the construction of the index, or the genomes are considered separately. Merging is recommended to improve the identification accuracy. It consists in merging the very close strains into a single pangenome. The Hamming distance is calculated between all the BFs, which makes it possible to obtain a distance matrix between genomes. From this matrix, ORI creates a histogram that represents the distribution of distances between genomes. This histogram gives an idea of the genome proximity in a graphical way and helps to determine a threshold θ to group the strains considered as too close ($\theta = 2e^{-4}$ for the experiments with *S. thermophilus* of section 5.1). Once a threshold is set, the strains whose Hamming distance is lower than the threshold are grouped as explained in section 3.2.3 and their BFs are merged. The result of the identification process depends in part on the choice of this threshold, which must allow to group a limited number of strains with very similar genomes. The creation of the index tree structure is then performed in the same way as without merging (see section 2.1.2 for the index structure).

4.2.2 Index query and read assignment

The second step of ORI (query part in figure 4.4) takes as input the sequenced reads of a sample and assigns these reads to bacterial strains present in the index. It starts by filtering the reads, using their length and their quality to keep only the reads containing the most information with the least amount of errors. Then, a manual selection may occur. Following our preliminary experiments (see section 5.1 for details), the best strategy seems to keep only a reduced number of reads randomly chosen among those passing the filter. This reduces the computation time and improves the identification results: as we have already pointed out before (see section 4.1.2), too many reads brings too much noise related to sequencing errors which decreases the accuracy of our method (see section 5.1 for more details). Then the q-grams are extracted from each read and the index is queried with the read q-gram set. A minimum number of q-grams from the read (half of them, see –threshold parameter in table 4.3) must be found in a strain for the read to be retained. This threshold allows an efficient filtering of reads that are relevant to the species of interest. It takes advantage of the length of the reads and allows contaminants to be removed, which is more difficult with short reads [Marcelino, Holmes, and Sorrell 2020]. For example, we have highlighted by experiments presented in section 5.3 that reads from *Streptococcus macedonicus* would not be found in an index containing genomes of *Streptococcus thermophilus*. The result of a query is a list, for each read, of their assigned strains, weighted by their percentage of common q-grams. This list is then summarised in a $read \times strain$ matrix that contains for each pair $\{read, genome\}$ their percentage of common q-grams (see affiliation matrix in figure 4.4). It is this matrix that we use afterwards to identify the strains really present in the sample.

4.2.3 Identification of the bacterial strains

The final step (identification part in the figure 4.4) is the identification process itself, which takes the $read \times strain$ affiliation matrix as input and searches for a minimal set of strains that best explains the set of reads in the sample. This identification step of strains based on the reads assignment is achieved by optimization, as explained in the section 3.2.2.

4.3 Perspectives: Learning identification rules at a higher taxonomic level

ORI gives good identification results on isolates or mixture made of small number of strains. However, its current weakness is increase of response time when the index becomes very large. It is therefore harder to identify mixtures that are too diverse in terms of species, genus, family or order. However, an index containing all the lactic bacteria could be of interest in the context of an analysis of a dairy products for example. To compute efficiently this order-level indexing, the principle is to create several ORI indexes at the genus/family level for sparse genera or families or at the species level. We then need a fast technique to quickly assign the reads to the genus/family to which they pertain in order to query only the appropriate genus indexes with the corresponding reads.

The first idea is to use another identification software that stops at the species level such as Kraken 2 and run strain identifications from the cluster of reads of each identified species. We used this strategy to study *Escherichia coli* strains from a metagenomic sample of pig gut microbiota (see section 5.3). It worked well but is of course dependent on the accuracy of the software chosen. To avoid this, the idea is to stay higher in the taxonomic tree in order to obtain clearer, more robust delineations between bacterial groups.

We tested this idea on an ORI index (the order index) starting from the *Lactobacillales* order and stopping at the level of the 71 genera of this order. In addition, an index going down to the strain is also created for each of the genera (the *genus indexes*) except for the genera containing only one known genome. Querying the genus indexes can be done in a parallel, allowing to save time.
Two tests were performed to create the order index. An index containing the union of the strains q-grams of each genus and an index containing their intersection. In both cases we faced a problem coming from the fact that some genus contain a lot of different genomes. This is for example the case of the *Lactobacillus* genus whose taxonomy was modified in 2020 (see section 1.3.1), which was not yet taken into account in the NCBI taxonomy that we used for this test. As the genomes of *Lactobacillus* were very diverse, the number of remaining q-grams was too small to allow a correct identification when making the intersection of q-gram sets present in each strain of the genus. In the case of the union of all these q-gram sets, the number of different q-grams was on the contrary too important. All requested reads were assigned to *Lactobacillus* because the q-grams of these reads were almost always found in this genus although each q-gram did not come from the same *Lactobacillus* strain.
In general, it is important to remember that the notion of bacterial species is still vague. Classification errors can be made and the union and intersection of genomic content are very sensitive to these errors.

Subsequently we came up with other ideas presented below. This work is still in preliminary but presents interesting research axes to improve the ORI software.

4.3.1 Using the tetranucleotide vector of a sequence

To identify sequences at the taxonomic level of genus or family it is not necessary to be as precise as for species or strains and a global statistics could be sufficient. The use of tetranucleotide composition has be used in metagenomics studies [Brisson *et al.* 2012]

and could be considered as a simple means to achieve identification.
A tetranucleotide is a 4-mer, i.e. a 4-letter word in the alphabet $\{A, T, G, C\}$. There are $4^4 = 256$ possible tetranucleotides. However, as for q-grams, only the canonical form ($can(S)$) of the tetranucleotides need to be considered since it is impossible to determine the stain that is sequenced. There are 136 possible canonical tetranucleotides noted $Tetra[1:136]$ (with 16 palindromes).

In the end, the tetranucleotide composition $comp(S)$ of a sequence S of size n is a vector of size 136 obtained by the formula:

$$compt(S)[i] = \frac{1}{n-3}\sum_{j=1}^{n-3}(can(S[j:j+3]) == Tetra[i]) \tag{C}$$

It is known that the tetranucleotide composition of a genome is consistent with 16S ribosomal RNA sequence analysis [Teeling *et al.* 2004] and allows to find species-specific information in genomic fragments of 10 kb and sometimes even fragments of 1 kb [Abe *et al.* 2003]. In [Sandberg *et al.* 2001], authors showed that by using tetranucleotides there is 85% probability to correctly classify genomic sequences of 400 bases length in a species. It would be interesting to extend this type of work to ambiguous recognition with several candidates (species/genus) because it is possible to query more than one index with the same sequence in parallel.

4.3.2 Classification of reads into a family/genus

If we consider a set of genera $G = \{G_i\}_{i=1..m}$, each genus consists of a variable number of strains sequences S_{ij}. The tetranucleotide composition of the reads that we want to classify into one of the genera must be as close as possible to that of the chosen genus. So, if $G_i = \{S_{ij}\}$, we want to learn a classifier of reads into genera $C(comp(R))$ that, given the set of $comp(S_{ij})$ sequences in each genus G_i and a learning set of reads rs_i whose classification is known, takes as input the tetranucleotide composition of a read $comp(R)$ and produces as output the genus that maximises the number of well-classed reads in the training set.

To better understand the complexity of the problem, tests were conducted using an average vector of the composition vectors on the set of sequences from a same genus of the *Lactobacillales* order. We calculated the Euclidean distance between these vectors and the vector of the considered reads. Figure 4.6 shows that the vectors of the reads of a given genus have a dispersion that can be greater than that of the genera, which makes the method inapplicable in practice. It is therefore necessary to keep a certain variability at the level of the members of a genus.

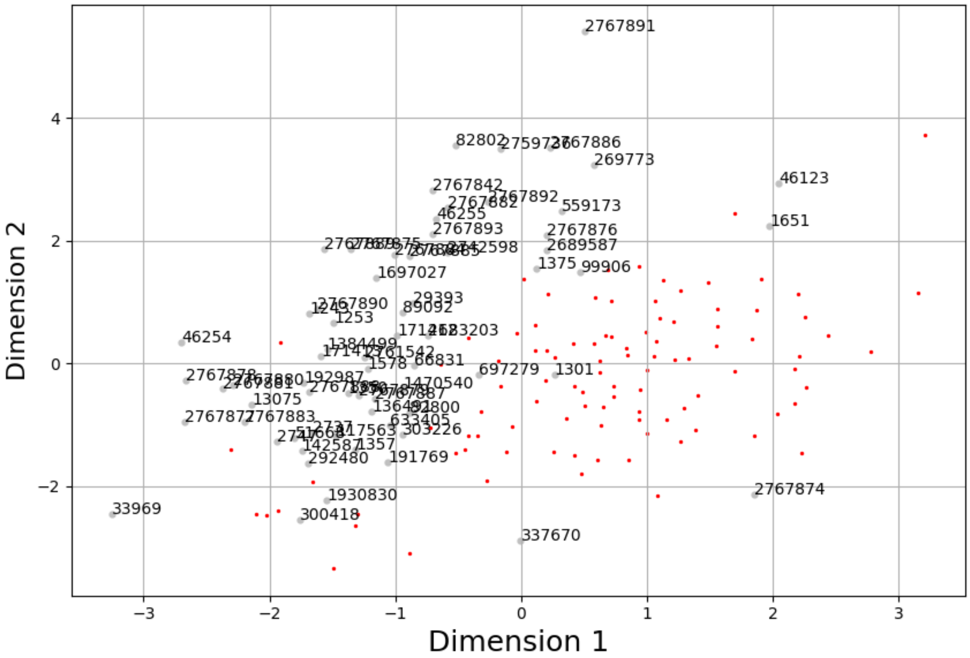


Figure 4.6: *Multi Dimensional Scaling showing the mean tetranucleotides vectors representing bacterial genera.* The red dots represent the projection of 100 vectors corresponding to reads from a strain of genus 1 301 (coordinates near the origin).

Some directions for further research are listed:

- For training data :
  - select the most informative tetranucleotides, for instance by principal component analysis (PCA).
  - work with the full set of genus/family vectors rather than an average vector.
  - split the complete genomes into fragments of size 4 000, then add errors to them according to the nanopore error profiles and consider all the resulting vectors of tetranucleotide composition.
- For learning methods:

- use of deep learning with noisy fragment data
- learning decision forests or discrimination rules on vectors of complete or fragmented genomic sequences.

Chapter 5

Validation of ORI on lactic acid bacteria and first experiments on other species

Contents

Preamble: This chapter presents three experiments to identify bacterial strains from long ONT reads via the use of ORI. The first experiment is the one presented in the paper [Siekaniec *et al.* 2021] and corresponds to the identification of *Streptococcus thermophilus* strains with ORI, Kraken 2 and StrainSeeker. The second experiment corresponds to the identification with ORI of strains from different dairy products. Finally the third experiment corresponds to the identification of *Escherichia coli* strains from pig intestinal microbiota as well as other small tests on *Streptococcus pyogenes* strains with ORI.

5.1 Experiments on *S. thermophilus* strains

Our first bacterial strain identification experiments were to identify strains of our model species *S. thermophilus*. These experiments were performed to:

1. test if ORI was able to perform a fine taxonomic assignment with a species containing close related strains, despite the high error rate of ONT reads.

2. compare our ORI's results with results from other bacterial identification software, Kraken 2 and StrainSeeker (see section 2.1.2).

For this purpose, 46 genomes of *S. thermophilus* from public databases and 31 strains from the CIRM-BIA collection were used. In addition to *S. thermophilus*, the strains *Streptococcus macedonicus* PA and *Lactobacillus delbruecki* subsp. *bulgaricus* ATCC 11842 were also used as controls (see the heatmap in figure 3.8 in section 3.2.1). These two strains represent a different bacterial species and family.
All the *S.thermophilus* strains used in this experiment are presented in appendix table S1 and S2.

5.1.1 Bacterial strains sequencing and read filtering

In order to get nanopore sequences of the 31 strains of *S. thermophilus* from CIRM-BIA as well as *S. macedonicus* PA and *L. delbrueckii* subsp. *bulgaricus* ATCC 11842, these strains were sequenced at INRAE from Rennes in the UMR STLO by Emeline Roux who took care of the wet lab part associated with this thesis.

Briefly, *S. thermophilus* and *S. macedonicus* strains were pre-cultured anaerobically and then grown on LM17 medium [Terzaghi and Sandine 1975] while incubated at 42°C. *L. delbrueckii* was grown on MRS medium [De MAN, Rogosa, and Sharpe 1960] under the same conditions.
One of the important steps in DNA sequencing with nanopore technology is the size of the DNA molecules extracted from the bacteria. The goal is to extract the DNA molecules of the bacteria by fragmenting them as little as possible. For this experiment on *S. thermophilus* strains a long fragment extraction kit was used (Genomic-tip 100/G from Qiagen). Once the DNA molecules extracted, the preparation of the library was done following the protocol "Rapid Barcoding Sequencing (SQK-RBK004)" of ONT and starting with 400 ng genomic DNA per strain and the 12 barcodes in order to sequence 12 strains at each sequencing.

The sequencing part was performed with an R9 flowcell (R9.4.1, FLO-MIND106D) and run for 48 h. The amount of data generated is about 7 to 9 Gbp per sequencing, meaning approximately 583 Mbp per strain, representing theoretically a sequencing depth of more than 300X for each *S. thermophilus* strain whose genome is on average 1.83 Mb. In reality, since concentrated DNA is not very homogeneous, it is complicated to get the same amount of DNA for each sample. The 12 barcodes are therefore not all equally covered. The ONT MinKNOW software (version 19.05.0) was used to monitor the sequencing and generated fast5 files containing the raw signal from the sequencer. These fast5 files were then basecalled (see section 1.2.3) with Guppy (version 4.4.1, default setting) in high-accuracy mode in order to obtain the corresponding nanopore reads.

In addition to testing ORI on real data, the Nanopore reads were also used in conjunction with Illumina reads to assemble the complete genome of *S. thermophilus* strains from the CIRM-BIA collection (see section 1.3.2).

The reads obtained contained an average error rate of about 5% with 3% insertions/deletions (see table 3.1 in section 3.1.1). In order to improve the quality of these reads a filter was applied to keep only those with an average quality score of at least 9 and a size of at least 2000 (see section 3.2.2).

5.1.2 Indexes used for the identification

In order to perform the identification experiments, two ORI indexes were created as explained in section 4.2.1 (without and with sibling strains merging). Both indexes contain the 77 *S. thermophilus* strains mentioned above (46 from databases + 31 from CIRM-BIA) as well as the two strains *Streptococcus macedonicus* PA and *Lactobacillus delbruecki* subsp. *bulgaricus* ATCC 11842 serving as controls. Plasmids were excluded from the indexes because they are considered as a part of the mobilome (set of mobile genetic elements in an organism) and not strain specific. The q-grams of the genomes are inserted into Bloom filters of size 5×10^8 bits using the 111111001111111 spaced seed (see section 3.1.2). The size of the filters used in this experiment is greatly overestimated. If we take the average size of a *S.thermophilus* genome of 1.8 Mbp and consider that all its q-grams of size 15 are different, entering them in an HowDeSBT Bloom filter (1 hash function) of size 5×10^8 will give a false positive rate of about 0.37% which is very low. The difference between the two indexes is that in the second index the sibling strains are merged into one filter using a threshold $\theta = 2e^{-4}$ as explained in section 4.2.1.

As said before, we compared ORI with Kraken 2 (version 2.0.9-beta) and StrainSeeker (version 1.5). These two softwares have common methodological bases with ORI that are the use of k-mers inserted in an index in order to be efficient during the identification. To compare these softwares with ORI, we used the same set of indexed genomes. Furthermore, as StrainSeeker requires a guide tree in addition to the genomes, we used for this purpose the clustering tree generated with MicroScope. This tree contains the 77 strains of *S. thermophilus* (see section 3.2.1), to which the two genomes of *S. macedonicus* and *L. delbrueckii* subsp. *bulgaricus* were added. For Kraken 2 and StrainSeeker, the size of the k-mers used was chosen among the recommended defaults values of the softwares, i.e. a size of 36 with a minimizer of 32 for the Kraken 2 index and a size of 16 for the StrainSeeker index. The final index sizes are 23.4 Mo for the classic ORI index and 22.7 Mo for the ORI index with the sibling strains merged, which are close to the Kraken 2 index of 18.1 Mo and smaller than the StrainSeeker index of 2.1 Gb. The fact that the ORI index is slightly larger than the Kraken 2 index can be easily explained. Indeed, the size of the ORI index depends largely on the proximity of the genomes stored in it and a little bit on the size of the filters used (largely overestimated in this experiment). As an example, if we index 99 complete genomes of *S. thermophilus* from the databases (September 2021) with filters of size 2.1×10^8, the final size of the index is only 1.3 Mo (see table 4.1 in section 4.1.1). The index size is smaller with 99 strains than with the 79 strains because there are only *S. thermophilus* strains among the 99 strains and the main factor of growth is the proximity of the indexed strains, a desirable property if interested by strain level (see section 4.1.1).

Querying indexes

For the ORI indexes the identification of strains in each experiment was performed as presented in section 4.2. Briefly, the filtered reads are assigned to the strains by querying the indexes with a minimum q-gram occurrence threshold of 0.5 (see section 4.2.2). Then an exact optimization step allows the selection of a minimum number of strains explaining the totality of the reads as presented in section 4.2.3.
For Kraken 2 and StrainSeeker the default parameters were used for the bacterial identification.

5.1.3 Experimental design

Parameters

In order to properly test our method and compare it to Kraken 2 and StrainSeeker, 180 strain identification experiments were performed.

These experiments used the reads of *S. thermophilus* previously sequenced (see section 5.1.1) in order to create mixtures of real *S. thermophilus* nanopore reads whose the strains composition and the proportion of reads for each strain are known. The size distribution of the sets of reads have a mean of 8 842 bp, a median of 6 436 bp, a standard deviation of 7 861 bp and ranges from 2 000 bp to 189 000 bp.
Thereafter, each identification result presented is an average of the results on a set of parameters fixed beforehand. The different parameters used in the experiments are the following:

- the number of strains: 4 or 6 strains.
- the number of reads: 1 000, 4 000 or 16 000 reads.
- the proximity of the strains: distant, moderately close, or close strains (see table 5.1).
- the distribution of the abundance of the strains: uniform distribution or distribution with dominant and subdominant strains.

Using these parameters, there are a total of 36 different possibilities to parameterize the experiments. For each possibility, 5 replicates were performed by randomly recovering reads from the sequenced strains.

In order not to bias the results for experiments with an increasing number of reads (second parameter), the 16 000 reads are first randomly selected in the filtered reads from the corresponding strains, then 4 000 reads are randomly selected from this first 16 000 reads and finally the last 1 000 reads are randomly selected from the 4 000 reads, all by keeping the proportion of each strain.
For the strain proximity experiments (third parameter), those labeled "close" are experiments in which each strain has at least one other close strain. A strain is considered to be close if the Hamming distance between the two strains is lower than the distance threshold set for grouping sibling strains. However, there may be distant strains in these experiments. The average Hamming distances of the reads of the "close", "moderately close" and "distant" experiments are presented in table 5.1.

Table 5.1: *Hamming distance between S. thermophilus strains* for strain proximity experiments (identification of strains more or less close). The Hamming distances are multiplied by 10^4.

Distance $\times 10^4$	**close**	**moderately close**	**distant**
Average	7.16	7.72	11.94
Mean std	5.21	2.23	3.81
Average Min/Max	0.0/11.6	2.7/10.3	6.4/16.4

For experiments with different strain abundances (fourth parameter), half of the strains in the sample were arbitrarily considered to represent the subdominant strains. Thus, the subdominant strains are:

- the two strains with the least number of reads for experiments containing four strains. Their associated number of reads represents 12.5% of the total number of reads of the experiment. The two other strains correspond respectively to 50 and 25% of the reads
- the three strains with the least number of reads for the experiments containing six strains. The number of reads of these three strains represents respectively 6.25, 3.12 and 3.12% of the total number of reads of the experiment. The tree other strains correspond respectively to 50, 25 and 12.5% of the reads.

The exact composition in strains, associated with their number of reads of the experiments is shown in table 5.2.

During the presentation of the results the experiments are separated into two parts, first the 90 experiments with a uniform number of reads per strain and then the 90

Table 5.2: *S. thermophilus strains composition of the 180 strains identification experiments and number of reads per strains.*

Strains proximity	**Close**		**Moderately close**		**Distant**	
Strains number	**4**	**6**	**4**	**6**	**4**	**6**
Experiment 1	CIRM65 CIRM23 CIRM1049 CIRM1048	CIRM65 CIRM36 CIRM23 CIRM67 CIRM32 CIRM18	JIM8232 CIRM1358 CIRM1047 CIRM1051	CIRM2101 CIRM1047 CIRM65 CIRM1049 CIRM772 CIRM1035	CIRM1047 CIRM368 CIRM336 CIRM956	CIRM1121 CIRM1055 CIRM1047 CIRM1116 CIRM772 CIRM336
Experiment 2	CIRM67 CIRM36 CIRM29 CIRM1122	CIRM18 CIRM32 CIRM23 CIRM67 CIRM2101 CIRM65	CIRM1050 CIRM65 CIRM1035 CIRM1116	CIRM32 CIRM1055 CIRM67 CIRM1125 CIRM1048 CIRM30	CIRM1050 CIRM2101 CIRM30 CIRM1051	CIRM30 CIRM961 JIM8232 CIRM368 CIRM1125 CIRM772
Experiment 3	CIRM1048 CIRM1049 CIRM1122 CIRM29	CIRM1116 CIRM1122 CIRM961 CIRM967 CIRM67 CIRM2101	CIRM30 CIRM1055 CIRM1051 CIRM67	CIRM19 CIRM2101 CIRM1050 CIRM772 CIRM1116 CIRM23	CIRM1125 CIRM1121 CIRM772 CIRM18	CIRM1050 CIRM1055 CIRM772 CIRM32 CIRM1121 CIRM1116
Experiment 4	CIRM2101 CIRM36 CIRM967 CIRM961	CIRM1116 CIRM29 CIRM65 CIRM2101 CIRM23 CIRM67	CIRM1055 CIRM36 CIRM1046 JIM8232	CIRM30 CIRM32 CIRM1358 CIRM65 CIRM1116 CIRM2101	CIRM1125 CIRM998 CIRM30 CIRM961	CIRM1121 CIRM19 CIRM1046 CIRM961 CIRM1050 CIRM29
Experiment 5	CIRM1116 CIRM29 CIRM961 CIRM967	CIRM36 CIRM2101 CIRM67 CIRM65 CIRM961 CIRM967	CIRM18 CIRM36 CIRM772 CIRM32	CIRM368 CIRM961 CIRM967 CIRM336 CIRM998 CIRM956	CIRM1046 CIRM961 JIM8232 CIRM18	CIRM336 CIRM1049 JIM8232 CIRM32 CIRM772 CIRM998
Distribution	**Uniform**			**Dominant and subdominant**		
4 strains	**1000 reads**	**4000 reads**	**16000 reads**	**1000 reads**	**4000 reads**	**16000 reads**
1	250	1000	4000	500	2000	8000
2				250	1000	4000
3				125	500	2000
4						
6 strains	**1000 reads**	**4000 reads**	**16000 reads**	**1000 reads**	**4000 reads**	**16000 reads**
1	167	667	2667	500	2000	8000
2				250	1000	4000
3				125	500	2000
4				62	250	1000
5	166	666	2666	32	125	500
6				31		

experiments containing low abundance strains.

Validation of results

To validate the identification results of the three programs we decided to use the *sum of the Hamming distances* between the strains predicted by the programs and the closest real strain present in the sample. This measure has the advantage of providing more accurate distance information than a simple false positive calculation (strain present or absent). The results are either 0 in the case of perfect identification, or a positive number otherwise. In this way, even if a software identifies a strain not present in the sample, if it is close to the expected strain, the distance obtained will remain low. However, this distance sum measure poses a problem since unidentified strains are not taken into account in this measure.

To overcome this problem, the *Matthews correlation coefficient* (MCC) was also calculated to measure the precision/sensitivity trade-off of the identification. This MCC is calculated as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{A}$$

with TP the number of true positives (correctly identified strain), FP the number of false positives (falsely identified strain), TN the number of true negatives (strain not identified because not present) and FN the number of false negatives (strain not identified while present). As a correlation coefficient the MCC ranges from 1 for a perfect identification to -1 for a complete disagreement between observation and reality and 0 indicates no relationship. Finally, these two measures are balanced by what we call an *ambiguity ratio*, which is the ratio between the number of predicted strains and the number of actual strains.
In its results, StrainSeeker does not always identify a single isolate but is able to propose groups of strains. In this case, the MCC and the ambiguity ratio were calculated by considering each strain as TP for the correct strains or FP for the others.

5.1.4 Identification results on *S. thermophilus* strains

First we have tested the identification of *S. thermophilus* isolates (only one strain in the sample). Then we have tested more complex mixtures (see table 5.2 in section 5.1.3). We conclude this section by presenting results on the same data, using the ORI index in which the sibling strains are merged.

Identification of isolated strains

The first experiment was to identify a single strain that was quite distant from the other (see figure 3.8 and 3.13) known strains of *S.thermophilus*. 4 000 long ONT reads were randomly selected from all sequenced reads for *S. thermophilus* JIM 8232 (approximately $\sim$ 200 000 reads). All three software programs tested were able to classify almost all reads. The results were as follow:

- For Kraken 2, 50.10% of reads were identified as *S. thermophilus* JIM 8232, 26.56% of reads were identified as *S. thermophilus* species (stop at the species level) and the rest of reads were misidentified.
- For ORI, 99.7% of reads were classified as *S. thermophilus* JIM 8232, with the remainder being unclassified.

- Although StrainSeeker was developed specifically for Illumina reads, it classifies all reads (100%) in the sample as *S. thermophilus* JIM 8232. This experiment therefore proves that it is possible to use it with ONT reads, which had not been tested before to our knowledge.

The second test concerned the *S. thermophilus* CIRM-BIA67 strain, which we will call CIRM67 hereafter. This strain is much more difficult to identify than JIM 8232 because of its proximity to other strains and in particular CIRM-BIA65 (see figure 3.8 and 3.13). Indeed, these strains belong to a cluster of sibling strains identified by our method (see table 3.3). The differences between these strains are mainly due to a contraction in CIRM67 of two tandem repeats of a 13.5 kb fraction of the genome. The identification results for an increasing number of reads are presented in table 5.3.

Table 5.3: *Identification of reads from S. thermophilus CIRM67 from various numbers of reads.*

Software	100 reads	1000 reads	10 000 reads	20 000 reads
Kraken 2	CIRM67 2%	CIRM67 1.70%	CIRM67 1.31%	CIRM67 1.23%
StrainSeeker	One group of many indistinguishable strains 100% (CIRM67 and 66 other strains)	CIRM67 100%	CIRM67 13.13% + 4 groups (other strains)	CIRM67 2.2% + CIRM65 1.86% + 22 other strains + 2 groups
ORI	CIRM67 100%	CIRM67 100%	CIRM65 100%	CIRM65 100%

The results are as follows:

- Kraken 2 classifies most reads as *S. thermophilus* species but less than 2% to the CIRM67 strain.
- StrainSeeker recognized the species level with a large group of *S. thermophilus* strains that cannot be distinguished with 100 reads. Its results are very good with 1,000 reads where it finds 100% CIRM67. Then with more reads, the identification becomes worse. This behavior will also be observed then discussed in the next experiments (see sections 5.1.4 and 5.1.4). In general, StrainSeeker is very sensitive to the amount of reads used for identification.
- The ORI identification is almost perfect in every case, 100% of CIRM67 up to 1 000 reads and 100% of its sibling strain CIRM65 with more reads.

From a general point of view, the error level of the long nanopore reads introduces limitations in the identification of closely related strains for all three programs. However, with ORI, this problem can be corrected by merging the sibling strains (see results in section 5.1.4).

Identification of strains mixture

The results presented in this section follow the experimental design described in section 5.1.3.
The figure 5.1 shows the identification results obtained by the three softwares with the 90 experiments containing a uniform number of reads per strain. The diagrams show boxplots for the sum of the Hamming distances between predicted and actual strains. These boxplots represent how close the identification results are to the expected results. The MCC and ambiguity ratio values are also given below the boxplots. The MCC, which measures the adequacy of the binary prediction (presence/absence of strains), shows the balance between specificity and sensitivity. The ambiguity coefficient is used because Kraken 2 and StrainSeeker propose ambiguous answers with more strains than actually exist in the sample. This overflow of strains tends to artificially increase the MCC, while ORI tries to minimize the number of predicted strains.

Four different representations of the same results based on the computation of the Hamming distances, the MCC and the ambiguity ratio are proposed in figure 5.1. The first representation show the *global identification* results of all 90 experiments (see in figure 5.1*a*). The results were then detailed by separating them with three parameters:

- the number of reads, which allows to observe the effect of *data quantity* on the results.
- the number of strains, which allows to observe the effect of *heterogeneity* in strains on the results.
- the proximity of the strains, which allows to observe the *resolution power* of the software.

Overall, the identification results are very good for ORI (figure 5.1a), both in terms of mean and standard deviation, showing the robustness of our method.
StrainSeeker also performs well and has the best MCC but is penalized by producing multiple solutions and with a larger variations than ORI.
Kraken 2 is less suitable for strain recognition.

Figure 5.1b shows the effect of increasing the number of strains on the results. The results are very similar for both 4 and 6 strains. The difference between 4 and 6 strains may not be large enough for an effect to be observed. It would have been interesting

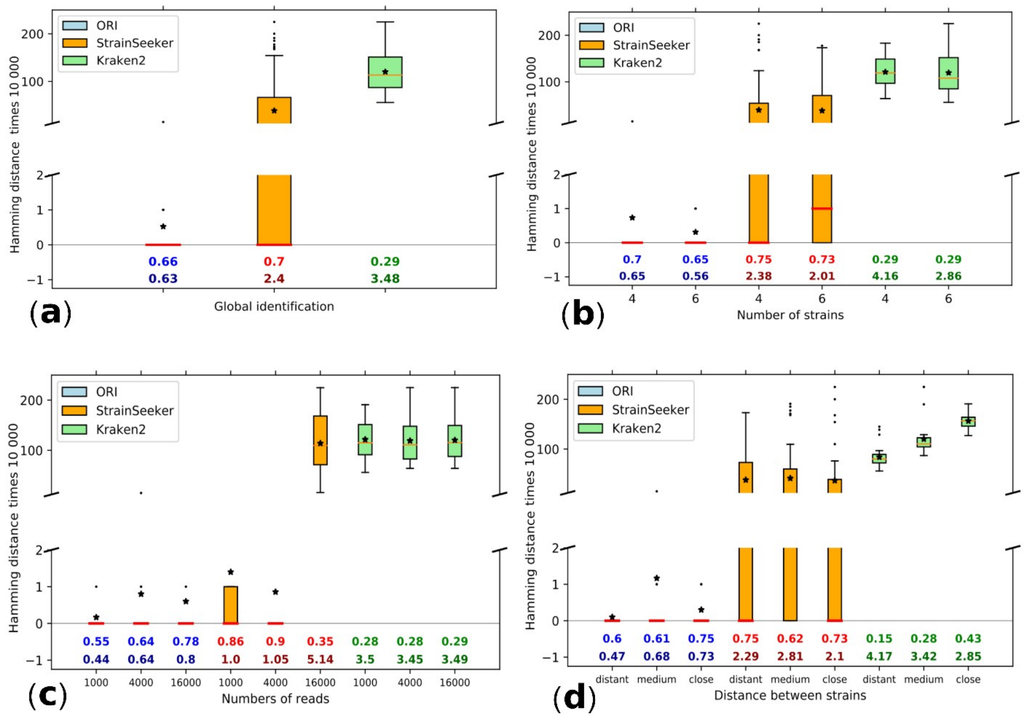


Figure 5.1: *Identification results on a balanced mix of S. thermophilus strains.* The Hamming distance between observed and expected strains, on the y axis, have been multiplied by 10 000 (blue: ORI, orange: StrainSeeker and green: Kraken 2). Stars represent mean values. Matthews correlation coefficient (MCC) values are given on the first line just above the x axis at the bottom of the diagrams. The second line represents the ambiguity ratio (number of strains identified/number of strains present).

to increase this number of strains but it would have require a complete redesign of the experimental set-up.

Using different amounts of data (figure 5.1c) shows that ORI seems to be sensitive to the number of reads. The more data, the better the identification in terms of MCC and ambiguity ratio. If we look at the distance we notice that the best distance is obtained with 1 000 reads. If we associate this with the MCC value and the ambiguity ratio we can then deduce that with few data ORI will tend to identify some strains well but miss others.

When using a large number of reads, StrainSeeker clearly drops in accuracy. Its results are best with 4 000 reads. This has already been observed for the identification of isolates (section 5.1.4), and can be explained by the way StrainSeeker identifies strains. Contrary

to Kraken 2 and ORI, StrainSeeker searches the index in the query, which means that if the number of k-mers in the query increases, notably because of erroneous k-mers, all the nodes in the query become saturated and the strains are all identified.
For Kraken 2, no difference was observed using 1 000, 4 000 or 16 000 reads.

The resolution power (figure 5.1d) of software has been measured with increasingly close mixtures of strains, which increases in turn the difficulty of identification.
Kraken 2 is sensitive to this parameter and provides worse results for a close mixture of strains.
With ORI and StrainSeeker, the effect is not marked but ORI has a slight degradation of the MCC and the ambiguity ratio. This behavior is due to the existence of very close *S. thermophilus* genomes and can be compensated by merging the sibling strains in the index (see section 5.1.4).

The last experiment in this section shows the identification results in the most difficult context; the identification of subdominant strains in a mixture of 4 or 6 strains (see figure 5.2). The 4-strains mixture contains two dominant and two subdominant strains, while the 6-strains mixture contains three dominants and three subdominant strains (see section 5.1.3).
The main point is that Kraken 2, although not as accurate as other software, does a better job at identifying strains in this situation, getting closer to the results of ORI and StrainSeeker.
ORI continues to perform well for strains with at least 500 reads but its results are degraded below that (see table 5.2 for the number of reads).
As before, StrainSeeker seems to be really affected by the number of reads to use for identification. Its results are even better than those of ORI when the number of reads is 4 000 but they are really bad with 16 000 reads.

Identification of mixtures after sibling strains merging

One of the problems of ORI in previous identifications was the presence of very similar strains in the sample. It has been shown that it is useful to obtain an identification of a group of very similar isolates [Van Rossum *et al.* 2020], rather than the exact isolate itself.
To do that, we tested the preprocessing step consisting of clustering those close strains (sibling strains) to measure its effect on the ORI's results. The results obtained are presented in tables 5.4 and 5.5. We have chosen to display these results in tables rather than in diagrams because the Hamming distance values are very low. A more detailed

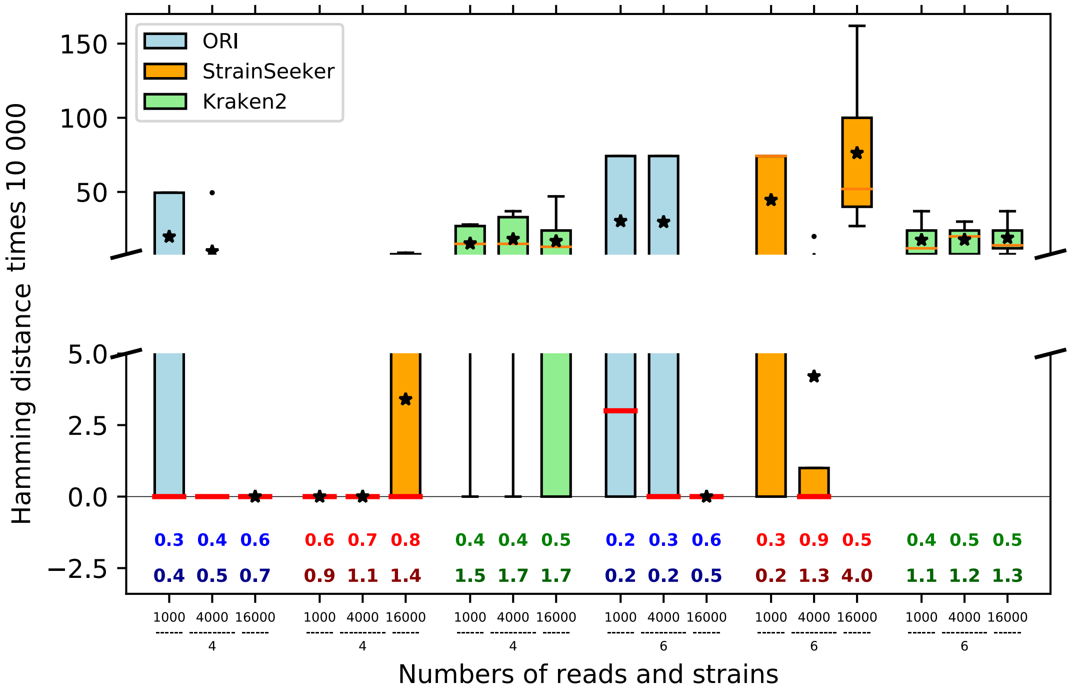


Figure 5.2: *Identification of subdominant strains in a mixture of S. thermophilus strains using various numbers of reads.* The Hamming distance between observed and expected strains, on the y axis, have been multiplied by 10 000 (blue: ORI, orange: StrainSeeker and green: Kraken 2). Matthews correlation coefficient (MCC) values are given on the first line just above the x axis at the bottom of the diagrams. The second line represents the ambiguity ratio (number of strains identified/number of strains present).

version (with median, standard deviation and minimum/maximum) of these results is available in the appendix of the thesis (see tables S3 and S4).

Overall, the merge of sibling strains leads to almost perfect identification results, with a Hamming distance that decreases compared to the classic version of ORI and which is most often zero (detailed results are in appendix table S3). However, this is still not the case for experiments with a very low number of reads per strain. The MCC increases with the number of available reads and is on average close to 1. This proves that ORI merge combines both high accuracy and high sensitivity. In some cases, the average distance is zero and the MCC is less than 1. This simply indicates that the accuracy is perfect (the identified strains are the right ones) but with a loss of sensitivity (missing strains). Overall, the method is more robust because variations in parameter values

Table 5.4: *S. thermophilus strains identification by ORI, without and with merge, in a balanced mixture of 4 or 6 strains more or less genetically close, by using 1 000, 4 000 or 16 000 sequencing reads.* Distance: Hamming distance between the observed and expected strains (0 = perfect identification); MCC: Matthews Correlation Coefficient (1 = perfect correlation); Ambiguity: number of strains identified/number of strains present (1 = perfect number of strains identified). The best results are shown in bold.

a) Global identification		
Method	**ORI**	**ORI merge**
Distance	0.52	**0.41**
(MCC/Ambiguity)	0.66/0.63	**0.92/0.91**

b) Heterogeneity				
Method	**ORI**		**ORI merge**	
Number of strains	**4**	**6**	**4**	**6**
Distance	0.73	0.31	**0.53**	**0.29**
(MCC/Ambiguity)	0.70/0.65	0.65/0.56	**0.94/0.93**	**0.96/0.96**

c) Data quantity						
Method	**ORI**			**ORI merge**		
Number of reads	**1 000**	**4 000**	**16 000**	**1 000**	**4 000**	**16 000**
Distance	0.17	0.8	**0.6**	**0**	**0.43**	0.8
(MCC/Ambiguity)	0.55/0.44	0.64/0.64	0.78/0.80	**0.86/0.77**	**0.93/0.92**	**0.98/1.05**

d) Resolution power						
Method	**ORI**			**ORI merge**		
Proximity	**distant**	**moderately close**	**close**	**distant**	**moderately close**	**close**
Distance	0.10	1.17	**0.30**	**0**	**0.90**	0.33
(MCC/Ambiguity)	0.75/0.73	0.61/0.68	0.6/0.47	**0.93/0.89**	**0.87/0.85**	**0.97/1**

have less influence on the results. The accuracy of the identification of strains present in the samples with and without the merge of sibling strains are presented in table S5 in the appendix. In this table it is clear that merging the bacterial strains improves identification results.

For the identification of subdominant strains presented in table 5.5 (detailed results are in appendix table S4). The results are much better with the merge of sibling strains when the number of reads is high enough (16 000 and 4 000 reads). ORI is then capable to correctly identify strains from only 250 reads (see table 5.1.3). For experiments containing 1 000 reads, where the number of reads for a subdominant strain is at most 125 reads, even ORI merge shows its limits and the gain is lower or almost null for strains with a maximum of 62 reads. The detailed identification precision for each subdominant strain when the number of reads in the sample increases is presented in table S6 in the appendix. Globally, the accuracy of identification of subdominant strains seems to improve when more reads are used, but this is not always the case.

Table 5.5: *Subdominant S. thermophilus strains identification by ORI, without/with merge, in a mixture of 4 or 6 strains, by using 1 000, 4 000 or 16 000 nanopore sequencing reads.* Distance: Hamming distance between the observed and expected strains (0 = perfect identification); MCC: Matthews Correlation Coefficient (1 = perfect correlation); Ambiguity ratio: number of strains identified/number of strains present (1 = perfect number of strains identified). The best results are shown in bold.

Number of strains	**4(ORI/ORI merge)**			**6(ORI/ORI merge)**		
Number of reads	**1 000**	**4 000**	**16 000**	**1 000**	**4 000**	**16 000**
Distance	19.8/19.8	9.9/**0**	**0/0**	30.3/**15.4**	26.7/**0**	**0/0**
MCC	0.28/**0.38**	0.42/**0.78**	0.57/**0.9**	0.22/**0.38**	0.34/**0.65**	0.63/**0.8**
Ambiguity	0.4/0.4	0.5/**0.8**	0.7/**1**	0.2/**0.33**	0.2/**0.47**	0.53/**0.67**

Conclusion and perspectives about *S. thermophilus* identification results

To conclude on *S. thermophilus* strains, ORI seems better than Kraken 2 in terms of bacterial strain identification. Kraken 2 was the worst solution for identification. However, the use of the default k-mer size of Kraken 2 was retrospectively probably not the best idea for a fair comparison with the other two software. It would therefore be interesting to retest Kraken 2 with a smaller k-mer size. Regarding StrainSeeker, which was originally developed for short sequences, behaves surprisingly well on the identification of small mixtures of strains from long nanopore reads when the number of reads is neither too high nor too low. However, the StrainSeeker index requires 100 times more space than ORI and Kraken 2, which will prevent it from scaling up to large datasets.

In the development of ORI, we made choices that take into account the fact that we identify strains from *erroneous long reads*. The first one is the use of q-grams instead of k-mers to be less sensitive to mismatch sequencing errors.
Another choice is the trade-off made between sensitivity and computational cost. For example, the trade-off made by Kraken 2, such as the use of minimizers for example, works well for the rapid identification of species in large metagenomic datasets but causes a loss in sensitivity, which makes identification at the strain level more complicated or even impossible. The trade-off made in ORI makes it more suitable for the identification of small mixtures of strains or isolates than for large metagenomic samples. Another argument in this line is that ORI uses an exact optimization step consisting in keeping the minimum number of strains explaining the majority of the reads. Exact optimization has a high complexity and should likely not scale up to a large number of strains to be identified in the sample. This unique feature of ORI increases the robustness of the

method (few false positives). However, it is also this step associated with the numerous preprocessing used to remove poor quality or uninformative data that make ORI tend to miss some strains sometimes.
The last choice is to add a preprocessing step for the grouping of sibling strains. This choice allows to be more precise in the identification, but to identify groups of close strains rather than isolates. However, as explained before, this choice is not necessarily bad as it is useful to be able to identify clusters of related strains rather than isolates (see the general conclusion on related strains in 6.1 for more details). Moreover, these clusters are generally far smaller than the cluster of strains identified by other methods.

A last interesting point to mention is the fact that ORI works with quite few reads. According to the results obtained in this section we recommend to use it with 4 000 reads and a bit more to identify subdominant strains. However, ORI is sensitive to the number of reads and seems to be less accurate as the number of reads increases too much. The decrease in quality of the results with a higher number of reads can be explained by the increase in the number of errors. However, the noise related to these errors should be removed by the preprocessing steps (see section 3.2.2). Working on the improvement of these preprocessing steps should be an interesting direction for further developments.

5.2 Identification in food: an experiment on dairy products

The experiment presented in this section is an experiment of strain identification in different dairy products such as commercial yoghurts and fermented milk including Ribot milk. The goal is (1) to get a global view on the strains present in different dairy product and (2) to show that this can be done quickly and at low cost (80 € / 810 €) with the use of the rapid barcoding kit and Fongles (mini ONT flowcell with about 60-80 active pores compared to the 1 300-1 600 pores of a classical flowcell).

5.2.1 Preparation of the experiments

The different commercial dairy products are the following:

- 3 commercial full-fat stirred yoghurts that will be noted A, B and C.
- 1 commercial fermented milk product containing *Bifidobacterium* in addition to *S. thermophilus* and *L. delbrueckii* that will be noted D.

- 1 fermented milk product; a traditional Ribot milk that will be noted E (traditional Breton product, fermented buttermilk).

In fact, more sample were tested but not retained for analysis because of the difficulty of picking a good protocol. This is notably the case for 0% fat stirred yoghurts, where the DNA extraction failed probably because of a higher composition of exopolysaccharide (EPS) compensating the absence of fat.
Globally, the DNA extraction from strains in dairy product is complex and requires more steps than a classic DNA extraction from isolates in culture. As a result, the recovered DNA molecules will be more fragmented than with a simple DNA extraction. In our case, we used the method for direct DNA extraction from the food matrix presented in [Parayre *et al.* 2007]. The samples were then barcoded and the library was prepared using the ONT rapid barcoding kit (SQK-RBK004). Finally, the sequencing was performed on a Flongle with the DNA of 5 dairy products (plus one sample exterior to the thesis).

Dairy products A, B and C are yoghurts which means, according to the law, that they are fermented milk containing only the thermophilic lactic acid bacteria *Lactobacillus delbrueckii* subsp. *bulgaricus* and *Streptococcus thermophilus*, which must be alive in the finished product [Légifrance 1988]. We therefore knew in advance which species we were looking for in these dairy products.
This is also the case for the fermented milk product D which is known to contain *Bifidobacterium* in addition to *S. thermophilus* and *L. delbrueckii* subsp. *bulgaricus*.
The study of Ribot milk is more complex because we have no idea of the bacterial species it contains.

Several indexes were created to perform the identifications.
The first test was to create an index containing the complete genomes of both *S. thermophilus* and *L. delbrueckii* (see index in section 4.1.1). Unfortunately, the identification results contained only *S. thermophilus* so it was decided to separate these two indexes.

In the end, the different indexes used were the following:

1. an index (size = 17 Mb) containing 95 complete genomes of *S. thermophilus* whose sibling strains have been merged ($\theta = 2 \times 10^{-4}$).
2. an index (size = 34 Mb) containing 237 complete genomes of *L. delbrueckii* whose sibling strains have been merged ($\theta = 2 \times 10^{-4}$).

3. an index (size = 249 Mb) containing 694 complete genomes of *Bifidobacterium* whose sibling strains have been merged ($\theta = 5 \times 10^{-6}$).

4. an index (size = 114 Mb) containing 272 complete genomes of *Enterococcus*, *Lactococcus* and *Leuconostoc* whose sibling strains have been merged ($\theta = 5 \times 10^{-4}$).

Indexes 1 and 2 were used for all experiments. Index 3 was used on the two fermented products. Index 4 was used only on the reads from the Ribot milk sequencing. It was built according to the results of the Centrifuge software (included in the ONT WIMP program) on these reads. These results rarely go down to the bacterial strain level but the identified genera were used to create the ORI index. We found, a majority of *Lactococcus* ($> 95\%$) and a few *Leuconostoc* ($> 1\%$) and *Enterococcus* ($> 0.4\%$). The other identified genera were not used because their percentage in the results was considered too low (arbitrary).
The new classifications for *Lactobacillaceae* and *Leuconostocaceae* (see section 1.3.1) were not used in these experiments because they have not yet been updated in the taxonomy of the NCBI database used for this experiment.

The number of reads for each experiment before and after filtering out poor quality reads is shown in table 5.6.

Table 5.6: *Reads number in experiments before and after quality and length filter.* The reads are filtered to a minimum quality of 5 and a minimum length of 1 000 bp.

Experiment	**type**	**Reads number**	**Reads number after filter**
A	Stirred yoghurt	9 000	4 300
B	Stirred yoghurt	2 749	1 756
C	Stirred yoghurt	9 722	5 224
D	Fermented dairy product	9 951	4 742
E	Ribot milk	14 418	5 425

As can be seen, the number of reads for each experiment is quite low. As a consequence, the filtering of these reads was reduced to filter only very low quality reads (< 5) and reads with a small size ($< 1\ 000$). Despite of these very lax quality filters, the number of reads in some experiments remains quite low. We thus did not hope to identify the subdominant strains with ORI.
In addition, since commercial products are most often inoculated with strains not present in public databases, the goal is just to identify the strains or strain clusters that are closest to the real strains.

5.2.2 Identification results

Overall, we found strains of *S. thermophilus* and *L. delbrueckii* subsp. *bulgaricus* in all the yoghurts studied. Similarly, *Bifidobacterium animalis* is found in the fermented milk D. Among the strains identified, yoghurts A and C contain a set of *S. thermophilus* strains quite close to each other. It is possible that these two yoghurts are inoculated with the same or at least similar strains of *S. thermophilus*. In yoghurt B, only one strain of *S. thermophilus* is identified, which is also found in the fermented milk D. These results seem quite logical because yoghurts A and C come from the same manufacturer, as well as dairy products B and D.
For the *L. delbrueckii* subsp. *bulgaricus* strains, only one strain was identified in dairy products A, B, C and D. The same cluster of two sibling strains is identified for the samples A, B and D while for the yoghurt C the identified strain is different.
Two Venn diagrams showing the identified strains common to the yoghurts (A, B and C) are presented in figure 5.3. On these diagrams we can see that yoghurt A is quite similar to yoghurt C with respect to the *S. thermophilus* strains and that it resembles yoghurt B with respect to the *L. delbrueckii* subsp. *bulgaricus* strains. Yoghurts B and C, on the other hand, seem to be completely different in terms of strains composition.

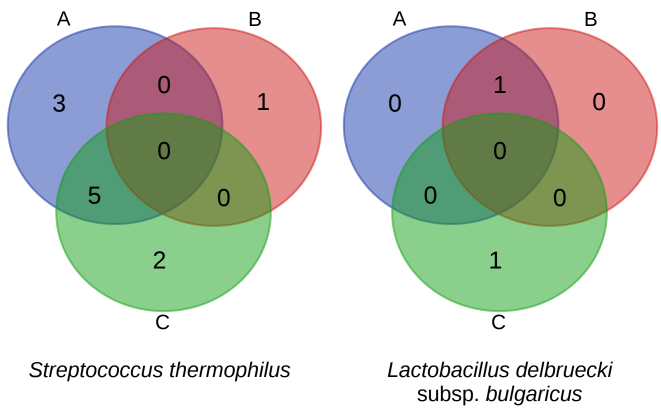


Figure 5.3: *Venn diagrams representing the common identified strains between the three stirred yoghurts.* The identification was done with the ORI software.

The fermented milk product D seems to be quite close in strain composition to yoghurt B from which it differs by the presence of a strain of *S. thermophilus* identified as present at less than 1% and a group of strains of *Bifidobacterium animalis* subsp.

Table 5.7: *Identification results of bacterial strains in different dairy products.* Strains/clusters of strains with a star are results identified in a very low percentage (< 2%) and therefore potentially false positives not present in the sample.

Experiment	Index	Identified strain
Stirred yoghurt		
A	S. thermophilus	ASCC 1275; MN-BM-A02; KLDS SM; ND07; DGCC 7710; 05-34; CS5; CS18 St1-WT; St1-GS-2; IDCC2201 * ATCC 19258; NCTC12958 24739; 13496; 13495 TH1436 GABA EU01 * TK-P3A
B		CIRM-BIA 1047
C		ASCC 1275; MN-BM-A02; KLDS SM; ND07; DGCC 7710; 05-34; CS5; CS18 MN-BM-A01; MN-ZLW-002; CIRM-BIA 1051 St1-WT; St1-GS-2; IDCC2201 * ATCC 19258; NCTC12958 * M17PTZA496 TH1436 EU01
A	*L. delbrueckii* subsp. *bulgaricus*	MBT 92059; KLDS1.1011
B		MBT 92059; KLDS1.1011
C		MGB30-1
Fermented dairy product		
D	*S. thermophilus*	CIRM-BIA 1047 * CS9
D	*L. delbrueckii* subsp. *bulgaricus*	MBT 92059; KLDS1.1011
D	*Bifidobacterium animalis* subsp. *lactis*	cluster of 27 strains (V9; S7; IDCC4301; TF04-14; OM05-7AA; BLC1; BS 01; BSD2780061688st1_E5; J063; J218; B06; BIOML-A2; B420; Bi-07; BIOML-A1; Bl12; TK-J6A; 1001713B170131_170501_H10; RH; Bl-04; H1; H3; i797; MGYG-HGUT-02459; i797; BF052; BB-12)
Ribot milk		
E	S. thermophilus	ASCC 1275; MN-BM-A02; KLDS SM; ND07; DGCC 7710; 05-34; CS5; CS18 24853; 13498; 13492; 13493; 13491; MAG_rmk202_sterm
E	*L. delbrueckii subsp. indicus*	JCM 15610; DSM 15996
E	*Bifidobacterium breve*	DRBB28
E	*Lactococcus lactis*	* L81-E2; L81; L62-G9; L62-C9 IPLA729 QH27_1 * CH_LC01 * MS22337 * MS22333 CF106
E	*Lactococcus cremoris*	* C4; G3-2; EPSC UC109 DPC6856 ATCC 19257 W34 1196
E	*Lactococcus lactis* subsp. *cremoris*	SK11 A76

lactis which are bacteria not present in a yoghurt. It is possible that the subdominant *S. thermophilus* strain is wrongly identified or also that this strain is present in the yogurt *B* but not identified. This second hypothesis is considered because the number of reads in the *B* sample is low (1 756) and if we refer to the experiments performed previously on *S. thermophilus* strains with few reads, there is a risk that ORI misses subdominant strains (see section 5.1.4).

Regarding the *Bifidobacterium animalis* subsp. *lactis* strains, a single big cluster of 27

similar strains was found for the experiment D. A representation of this cluster of 27 strains is given in figure 5.4. These strains are very close to each other because the Hamming distance threshold for clustering sibling strains is much lower than for the other indexes. However, it can be seen on the representation that some strains have been grouped together despite the fact that they have little connection with the whole cluster. This is due to the fact that the clusters correspond to the connected component (see section 3.2.3). The results could therefore be improved by using one of the two more elaborated approaches presented in section 3.2.3 to group sibling strains as dense subgraphs.
A graph representation of all existing proximity relationships between *Bifidobacterium* complete genomes with their assembly identifiers is presented in figure S2 and a representation without identifiers but with the name of the species present in the clusters containing more than 4 strains is presented in figure S3. On this last figure we can see the limits of the existing species classification of some bacterial isolates, since they are considered as two different species, but with genomes so close that they are grouped in the same cluster of sibling strains, even with a low distance threshold. The study of sibling strains containing several species could be a good test to check for possible misclassification of same species.

The last experiment of identification of strains present in Ribot milk (experiment L) was a little more complex because we had no idea of the species to be found in the sample. The results obtained show a *S. thermophilus* strain from a cluster of sibling strains also found in samples A and C as well as a cluster specific to Ribot milk. For Lactobacillus, only one cluster of strains was identified but, unlike the other experiments, it correspond to *L. delbrueckii* subsp. *indicus* and not to *L. delbrueckii* subsp. *bulgaricus* (a different subspecies). Finally, different species of *Lactococcus* have been identified in the sample. Among these species we found *Lactococcus lactis* (7 different strains/clusters) and *Lactococcus cremoris* (6 different strains/clusters). Bacteria of the subspecies *Lactococcus lactis* subsp. *cremoris* (2 strains) which is part of the species *Lactococcus cremoris* are also found.

5.2.3 Conclusion about strains identification in dairy products

As we have just seen, the identification of bacterial strains present in dairy products is much more complex than the identification of bacterial isolates. The challenge was mainly for the wet lab part. DNA extraction is a key step that is made much more com-

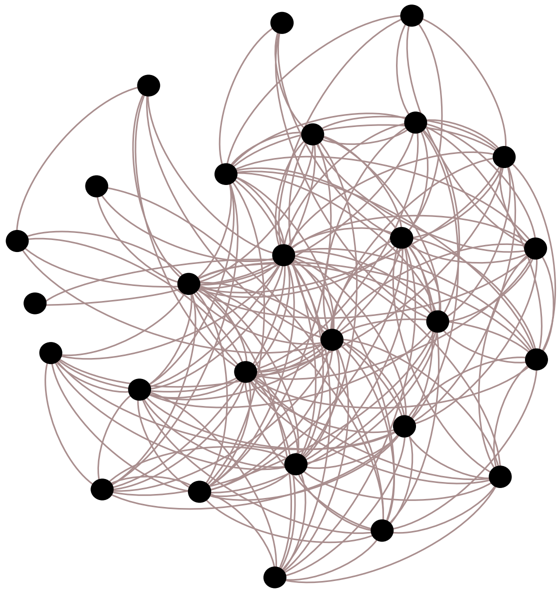


Figure 5.4: *Representation of the cluster of 27 strains of Bifidobacterium animalis subsp. lactis.* A node represents a strain and an edge represents a Hamming distance less than 5×10^{-6} between two strains. Strain identifiers are not displayed here because we are interested in the structure of the cluster and not in the specific strains composing it. The complete graph of all existing proximity relationships between *Bifidobacterium* genomes (with identifiers) is presented in figure S2.

plex by the presence of the the food matrix. Moreover, the particularities of the products may require different extraction methods, as it was observed with 0% yoghurts, where the DNA extraction did not work for this single experiment. Moreover, the extraction being more complicated, the number of steps to extract the DNA increases which will tend to produce shorter DNA fragments.

Regarding the sequencing itself, the samples were sequenced using a Flongle, which is less expensive but contains fewer active pores than a classical flowcell (see section 1.2.3). As a result, the amount of data obtained is reduced. If we combine this with the sequencing of 6 samples simultaneously, via the use of barcodes, the number of reads per sample in output is finally quite low, which makes the identification of bacterial strains more complicated.

ORI failed to identify *L. delbrueckii* subsp. *bulgaricus* when using the index mixing *S. thermophilus* and *L. delbrueckii.* One explanation could be that the number of reads was too low for *L. delbrueckii* which then becomes a subdominant species and is therefore no longer identified. For example if we look at the read assignment results of yoghurt *A* using the two separate indexes, 40 reads were assigned to *L. delbrueckii* subsp. *bulgaricus* while the 3 883 reads were assigned to *S. thermophilus*. As the reads of *S. thermophilus* are not assigned to *L. dellbrueckii*, ORI was able to perform the identification using only the 40 reads assigned. It is however possible that the identification performed with these 40 reads is not robust enough.

There is another problem that can occur with the use of barcoded samples and that we could not detect here: the fact that some of the barcodes may be misclassified while demultiplexing due to sequencing errors. Therefore some of the reads may not belong to the right sample. This is not a problem when the number of reads is very large but it may have an impact when data become sparse.

It was possible, by using separate indexes for the main species, to identify the strains present in the samples. Moreover, the results seem consistent with the background knowledge on the different dairy products studied (e.g. *S. thermophilus* and *L. delbrueckii* subsp. *bulgaricus* in yoghurts (*A*, *B* and *C*) to which *Bifidobacterium animalis* was added to obtain the fermented product *D*). These experiments served as a test bed for the identification of strains directly in food products with ORI, and other experiments are necessary to improve the wet lab part (DNA extraction, sequencing) and better parameterise ORI (discussed in sections 3.2.3 and 4.3 as well as in the conclusion of this thesis). It shows however that it is feasible and worth further investigations.

5.3 Strain identification in other species

In this last part, we will present the results obtained with other species than *S. thermophilus* and a negative control. These experiments present collaborative work to test ORI on other species than our *S. thermophilus* model. These experiences required the training of other people in the use of ORI, which made it possible to improve the software and its documentation.

5.3.1 Identification of *S. pyogenes* strains

This first two experiments were performed in collaboration with Emeline Roux who was working with the Clinical Investigation Center (CIC) of Rennes on the study of *S. pyogenes* strains [Devaere *et al.* 2020].

For this experiment, the DNA extraction was performed by a classical extraction with phenol-chloroform, which keeps DNA molecules long enough without being specific to this task.

Streptococcus pyogenes is a pathogenic streptococcus of clinical interest that is responsible for more than 600 million worldwide infections per year and has a high level of morbidity and mortality [Carapetis *et al.* 2005]. *S. pyogenes* is quite similar to *S. thermophilus* as they are both bacteria of the same genus. There are currently more than 2 000 genomes of *S. pyogenes* in the databases, including 244 complete genomes with some closely related genomes.
The experiment consisted of identifying reads from the *S. pyogenes* STAB14018 (*emm75* type) strain using ORI with long nanopore reads and an index containing 217 complete *S. pyogenes* genomes whose sibling strains were merged ($\theta = 1 \times 10^{-4}$). The size of the Bloom filters used to create the index was 2.5×10^{8} bit. The size of the final index was 33 Mb and included 23 clusters of sibling strains. This experiment is similar to the identification of a bacterial isolate as performed with *S. thermophilus* strains but on another species having more reference genomes.
The identification was performed with the first 4 000 reads filtered on their quality and length (quality > 9 and length $> 2\ 000$). The identification result obtained is 100% of a cluster of 3 close strains containing the desired strain *S. pyogenes* STAB14018 and two other strains: *S. pyogenes* STAB120304 and *S. pyogenes* NCTC13751 (see figure S4 in appendix).

5.3.2 Verification of the ORI specificity

The second experiment consisted in testing the ORI's behaviour in case of contamination by a different but close species. For this purpose, the 4 000 reads of *S. pyogenes* were used to query the index composed of *S. thermophilus* strains (same species) presented in section 5.1.4. In this case no reads are assigned because there are none with more than 50% of q-grams common to any strain in the index, and therefore no identification is performed. Similarly, when searching for *S. thermophilus* reads in the index containing only *S. pyogenes* strains no reads are assigned. This experiment shows that ORI is not impacted by contamination even when using an index containing only one species, which could lead to misclassification [Marcelino, Holmes, and Sorrell 2020].

5.3.3 Identification of *E. coli* strains in a complex microbiota

Finally, the last experience was realized during the internship of a M2 student, Rania Ouazahrou supervised by Emeline Roux and Gaelle Boudry from the NuMeCan Insti-

tute in Rennes. The aim of the presented experiment was to identify the strains of *E. coli* bacteria (see section 1.3.3) from a sequencing of pig intestinal microbiota.
The DNA was extracted using a specific kit for microbiota DNA extraction. However, as there are many species in a microbiota, and these species will be more or less difficult to lyse, the extraction performed with this kind of kit is very drastic and thus fragment the DNA molecules a lot. It is interesting to note that to our knowledge, no protocol exist to date for extracting long DNA from microbiota.
The sequencing of this microbiota produced 30 Gb of nanopore sequences. This represents 3 145 521 reads in total with at least 50% of reads having a size greater than 8 000 bp and an average quality score of 13.5.

The challenge of studying the *E. coli* strains in this metagenomic sequencing comes from both the number of reads to be processed and the number of known *E. coli* strains present in the databases. There are currently (in September 2021) over 22 000 *E. coli* genomes including over 2 000 complete *E. coli* genomes in the public databases (see section 1.3.3). Moreover, this number of complete genomes is increasing very rapidly. As an example, there were 1 693 complete *E. coli* genomes at the end of December 2020. The 1 644 complete *E. coli* genomes present in the databases at the time of the experiment were inserted into an ORI index. The size of the total index is 537 Mb and there was no merge of sibling strains since the creation time of the index was already very long (18h20m04s).

As explained previously (see section 4.1.2) ORI is not adapted to queries with many reads to identify bacterial strains. We tested two approaches to reduce the size of the query:

- using 4 000 random reads from the entire metagenomic sample. Since ORI is not sensitive to the contaminant, it should theoretically be able to assign *E. coli* reads without assigning those not belonging to this species.

- using the Kraken 2 software to extract reads from the species *E. coli* present in the metagenomic sample. On the 3 145 251 initial reads, only 32 178 reads (*approx* 1%) were identified by Kraken 2 as belonging to *E. coli*. Then the identification of the *E. coli* strains was done with ORI using this pre-identified reads.

The first approach with 4 000 randomly selected reads identified a single pathogenic *E. coli* strain; *E. coli* O157:H7. However, overall abundance of *E. coli* in the sample being estimated to 1%, the selection should contained about 40 reads of *E. coli*. This represents very few reads, especially since we expected more than one strain in the pig gut microbiota. Furthermore, it has been shown by experiments on *S. thermophilus* that ORI tends to give poor results with a reduced number of reads per strain (at least 500 reads per strain are required; see section 5.1.4). This explains that only one strain was recognized. A number of random draws should be used to obtain recognition of more strains

A second approach was chosen instead, to select among the total reads those belonging to the *E. coli* species, thanks to Kraken 2. However, as the index containing 1 644 genomes was large, in order to speed up the calculations, the 32 178 reads were separated into 8 files of 4 000 reads. The average computation time for the identification of bacterial strains for each group of 4 000 reads with an index of this size (537 Mb) was approximately 4h23m54s. The identification of the *E. coli* strains was performed from these 8 groups of reads separately. By pooling the results of the 9 identifications we obtained a total identification of 8 *E. coli* strains. Of these 8 strains identified, 5 were close strains that have been observed in pigs [Poulin-Laprade *et al.* 2021] and one strain is known to be a pathogenic strain (Enterotoxigenic *E. coli*) of pigs [Z. Li *et al.* 2018]. The 8 identified strains are presented in table 5.8.

Table 5.8: *Escherichia coli strains identified using 8 groups of 4000 long nanopore reads of E. coli from a pig intestinal metagenomic sample.* The reads corresponding to the species were first identified using Kraken 2 and then the strains were identified using ORI.

Strains	**Size**	**GC%**	**Assembly accession**	**Status**	**Reference**
Res13-Lact-PER12-33-A	5.11 Mb	50.59%	GCA_015571615.1	Complete Genome	[Poulin-Laprade *et al.* 2021]
Res13-Lact-PEA06-10	5.12 Mb	50.59%	GCA_015571775.1	Complete Genome	[Poulin-Laprade *et al.* 2021]
Res13-Lact-PER02-33	5.12 Mb	50.59%	GCA_015571655.1	Complete Genome	[Poulin-Laprade *et al.* 2021]
Res13-Lact-PER04-33	5.11 Mb	50.61%	GCA_015571635.1	Complete Genome	[Poulin-Laprade *et al.* 2021]
Res13-Sevr-PER06-05-b-A	5.11 Mb	50.60%	GCA_015571555.1	Complete Genome	[Poulin-Laprade *et al.* 2021]
AH01	5.22 Mb	50.61%	GCA_013371685.1	Complete Genome	
cq9	5.91 Mb	50.33%	GCA_003402955.1	Complete Genome	
CV839-15	5.26 Mb	50.84%	GCA_002803805.2	Complete Genome	[Z. Li *et al.* 2018]

It would be interesting to test these strains with an index where the sibling strains of *E. coli* would have been grouped together in order to determine if the identification of the different strains from [Poulin-Laprade *et al.* 2021] is due to their proximity or not.

At last, large scale test was achieved on the whole set of *E. coli* reads selected by Kraken 2 running ORI on all 32 178 reads and tests were made on subsamples of 16 000, 8 000 and 4 000 reads randomly selected among the totality of the reads. The results obtained are presented in table 5.9.

Table 5.9: *Escherichia coli strains identified using variant numbers of E. coli reads from a pig intestinal metagenomic sample.* The reads corresponding to the species were first identified using Kraken 2 and then the strains were identified using ORI.

Reads number	Strain	Assembly accession	Reference
32 178	Res13-Lact-PER12-33-A CV839-15	GCA_015571615.1 GCA_002803805.2	[Poulin-Laprade *et al.* 2021] [Z. Li *et al.* 2018]
16 000	Res13-Lact-PER12-33-A AH01	GCA_015571615.1 GCA_013371685.1	[Poulin-Laprade *et al.* 2021]
8 000	Res13-Lact-PER12-33-A	GCA_015571615.1	[Poulin-Laprade *et al.* 2021]
4 000	Res13-Lact-PER12-33-A	GCA_015571615.1	[Poulin-Laprade *et al.* 2021]

As can be seen, these experiments are successful in finding the Res13-Lact-PER12-33-A strain each time. Strains CV839-15 and AH01 also stand out in the experiments with many reads and could be subdominant strains of the sample.
For the identification of 32 178 reads with the index containing 1 644 *E. coli* genomes the identification took 1d8h7m19s which shows the limit of ORI in terms of computation time when the index and the number of reads increase.

In addition to *E. coli*, ORI was used to find strains of the majority species of the microbiota and gave good results in most cases except for the species *Phocaecola vulgatus* identified as present by Kraken 2 but where ORI did not identify any genome.

Globally, this experiment demonstrated the use of ORI in combination with Kraken 2 for the purpose of strain identification in a metagenomic sample. This is a first step towards the use of ORI on a larger scale data for the identification of bacterial strains in metagenomic samples.

5.3.4 Conclusion on the identification of strains not belonging to *S. thermophilus*

In general, the three experiments presented in this last section show that ORI is able to identify bacterial strains of different species including pathogenic species of clinical interest such as *Streptococcus pyogenes* or *Escherichia coli*.

The second experiment shows that ORI is robust with respect to the presence of genomes not present in the database. This characteristic is important when studying mixture of strains from different species.

Moreover, the experience with *E. coli* strains shows that ORI can be used in a larger context than the identification of bacterial isolates or small mixtures of strains. It is a starting point for strain identification directly from metagenomic data.

Chapter 6

Conclusions and perspectives

Contents

Preamble: This chapter presents some conclusions on the thesis as well as perspectives for the development of ORI and the identification of bacterial strains from ONT sequencing which seems interesting to improve both the speed and accuracy of our method.

This thesis brings some thoughts on the field of bacterial identification from Nanopore reads but also on adapting comparative genomics to the study of bacterial strains.

6.1 Conclusion and perspectives on ORI

The major result of the thesis is the design of a new identification method and the implementation of an associated software called ORI. ORI is based on a new association of:

- a k-mer set indexing technique.
- the use of spaced seeds.
- a preprocessing step for the clustering of sibling strains.

- an exact optimization step for the selection of a minimum number of strains explaining the totality of the reads.

This association allows an efficient storage of the reference strain genomes and a more error-tolerant assignment of the long Nanopore reads, while being memory efficient.
In addition to the development of ORI, part of the thesis have consisted in making it available to the scientific community, presenting it and training biologist and bioinformatician users to its use whose feedback has allowed and still allows its improvement.

Conclusion on the use of an HowDeSBT index with spaced seeds

From one side, using an extension of the HowDeSBT-based index allows ORI to be space and time efficient. Furthermore the use of q-grams instead of k-mers allows to be less sensitive to erroneous long reads generated by the Oxford nanopore technology.
From another side, the index query time is currently the limiting point of the ORI's method. Indeed, the creation of an index is always a matter of trade-offs. In the case of HowDeSBT the compromise is clearly in favour of the size of the index and the memory usage. This choice has an impact on the calculation time of the query. In order to improve it and to scale to a bigger number of strains there are several possible improvements.

Perspectives for improving the search time of the ORI index

We propose direction for further research based on three main ideas:

- A divide and conquer strategy based on a partition of the index allowing to quickly query it by going down in the taxonomy. It would be interesting to first quickly identify the genus (or family) of bacteria present in the sample, before identifying more precisely in a second stage the isolates or group of isolates present for each genus identified. Two propositions based on either using a fast existing species identification software or making use of a tetranucleotide vector representation have already been presented in section 4.3 of the thesis.

- An improved exploitation of system resources for handling the current HowDeSBT index. An easy modification would include the loading of the whole index in main memory, which would allow to slightly speed up the query at the expense of the used memory.
  It is also possible to use the very recent modular tool suite developed in our

team named kmtricks [Lemane *et al.* 2021], allowing the rapid construction of BFs for large collections of sequencing data. A kmtricks compatible version of HowDeSBT is already available and it should be possible to merge this version with our HowDeSBT extension which is expected to greatly speed up the index construction and query time.
In addition, another tool in development in our team, findere [Robidou and Peterlongo 2021] seems interesting to be considered. It enables to speed-up the queries time and to decrease the false-positive rate of approximate membership queries (such as BF queries) without modifying the original indexing data structure. Since it does not cause memory overhead, it is then theoretically possible to use it in conjunction with HowDeSBT and kmtricks to further accelerate our approach.

- The replacement of the index with a new one, faster to query with many long reads. In their recent review [Marchet, Boucher, *et al.* 2020] concluded that the color-aggregative methods and BIGSI/COBS seem better suited to query large sequences which is the case when working with long reads. In general COBS seems to be a good compromise because it is efficient in query time, indexing and uses quite little memory. The index created will be a bit larger than with HowDeSBT but it would remain acceptable for bacterial genomes. It would therefore be interesting to modify COBS to adapt it to the use of spaced seeds in order to test the trade-off between time savings during the query and the increase in the size of the indexes.

Perspectives about the spaced seed use

The use of spaced seeds allows to be less sensitive to mismatch errors. As seen in section 3.1.3, it is possible to take into account insertion deletion errors with indel seeds but at the cost of an important increase in terms of space and time. Currently there is a lack of k-mer indexing structure that would allow to efficiently index genomes using these indel seeds. There is still work to do in this area which would undoubtedly benefit our ORI software and more generally any treatment of erroneous long reads.

Conclusion on sibling strain clustering

The identification of bacteria is based on their classification but this classification is far from being stable and established. At the level of detail studied in this thesis, it is even more difficult and fuzzy. The definitions of bacterial strain, bacterial subspecies and bacterial clone remain very vague and, depending on the chosen definition, they may even overlap.

However, it is useful to obtain an identification resolution of a group of very similar isolates [Van Rossum *et al.* 2020], rather than the exact isolate itself. For example, in the medical field, the identification of epidemic strains is very important and represents a major use of MinION sequencing because of its portability allowing rapid sequencing in the field. However, in general, identification at the level of the individual isolate is of little practical value. As an illustration, a recent article [Gori *et al.* 2020] presents identification results on nearly 2000 samples of *Streptococcus agalactiae* characterized by different degrees of virulence and preferred host. The authors choose to distinguish only a few dozen different types at most despite the requirement of specialists for strain distinction.

One of the other major ideas of ORI is the clustering of sibling strains. This clustering has the advantage of allowing robust identification of clusters of close related strains and prevents misidentification of difficult strains. It is based on the calculation of genomic distances between strains and our experiments show that it leads to better identification results when processing long erroneous reads. Furthermore, as the strains are grouped during the index creation, the created clusters can be discussed with an expert knowing the precise characteristics of the grouped strains before any identification. It helps to determine the best clustering threshold to use and allows to plan other biological tests to further discriminate strains within a group.

For example, a recent application of clustering of closely related strains is the inference of antibiotic resistance and susceptibility [Břinda, Callendrello, *et al.* 2020]. In general, the identification of sibling clusters with ORI is quite robust. In the identification results, ORI errors are more likely to miss strains present in a sample (FN) than to identify strains that are actually absent (FP). These results therefore allow a more detailed investigation to discriminate the strain among a limited number of possibilities. When studying the iTOL tree containing the *S. thermophilus* strains, it was observed that the sibling strains were annotated mainly by specific genes of unknown function, unlike other strains (e.g. *S. thermophilus* JIM8232). However, two differently labeled unknown genes may be related. The sibling strains could therefore indicate groups with a compact pangenome useful for the identification of these strains.

Perspectives on sibling strain clustering

Based on these observations, an interesting perspective would be to verify that the clustering of nearby strains allows to distinguish strains with important phenotypic traits

such as antimicrobial resistance profiles [Greig *et al.* 2018].
It could also be interesting, in the case where genes of the different strains of the index are known, to add in ORI the computation of the maximal biclusters $\{strains\} \times \{genes\}$ and to use them in the identification of the strains by looking for the genes specific to a strain or a cluster of strains present in the reads. This study of the genes present in the strains could validate and in some cases refine the identification performed by ORI.
An interesting idea to study in more detail the clusters of sibling strains once identified would be to create the graph of variation of these strain and to realign the corresponding reads on this graph to determine more precisely the strains actually present. The construction of such a variation graph is too complex to be applicable to the totality of the index strains but for few strains of a cluster it becomes possible.
Another valuable point would be to integrate in ORI the creation of a heatmap such as the one presented in figure 3.8. It can be produced from the use of the Hamming distance matrix and include the sibling strains clusters according to a given threshold. This kind of representation would be easier to use than the current histogram to set a relevant clustering threshold.
Finally, the quadratic strain clustering step is the most time and memory consuming step during the index creation. In the case the index query time is accelerated, it would be the limiting step. We are mostly interested in closest strains and it could help to reduce the number of needed comparisons, for instance by making comparisons below the genus or family level. Moreover, other indexing method may lead to a faster estimation of distances.

Global discussion on bacterial strains characterisation

In general, the study of bacterial strains is necessary in many fields but the strain definition remains unclear. If we consider only the complete sequence of the genomes during the comparative study of strains of the same species and if we base the strain definition on a distance threshold between them, then this threshold will be different depending on the species studied. An interesting issue is then to find a threshold where the formed strain clusters allows to separate the individuals according to a studied phenotypic property (e.g. antimicrobial resistance, pathogenicity, probiotic effects).

Conclusion and perspectives about the ORI's strain identification part

For the last stage of strain identification, we have shown that the search for a minimum number of strains explaining most of the reads may be solved exactly and provides good results when considering a limited number of reads and strains. In case these numbers become too large the process will become too long. In order to prevent this, different filters of reads and strains are performed (see section 3.2.2). Furthermore, according to our experiments, too many reads seem to decrease the accuracy of the ORI identification. The preprocessing with filters is used to reduce the information quantity (reads and strains) to be processed, both to decrease computation time and to increase data quality, by keeping only the most relevant information to identify the strains really present. A good parameterization of these filters should allow to obtain the best possible information/noise ratio.

Estimating strain abundance

It is possible to estimate strain abundance in ORI using an Expectation-Maximization (EM) algorithm similar to that used in Centrifuge. However, ORI already provides access to indicative abundance values in addition to the strain identification that can help to interpret the identification results. To demonstrate the utility of this quantification, using the example from [Siekaniec *et al.* 2021], let's consider one of the *S. thermophilus* strain experiments on a sample containing 16,000 reads of 6 *S. thermophilus* strains equally distributed and forming 2 clusters of sibling strains: A={CIRM-BIA18, CIRM-BIA32} and B={CIRM-BIA2101, CIRM-BIA23, CIRM-BIA65, CIRM-BIA67}. The sample is thus composed of 1/3 of A and 2/3 of B. The identification made by ORI is as follows: 25% of A, 72% of B and 3% of a third cluster C={CIRM-BIA1116, CIRM-BIA1122, CIRM-BIA16, CIRM-BIA29, CNRZ1066, CS8, EPS, S9} not really present in the sample. However the current abundance of C provided by ORI is quite low and C is quite close to cluster B (see figure 3.8), so it is reasonable to assume that C is not actually present in the sample. By increasing the merging threshold of the nearby strains, B and C would indeed be fused, which would lead to perfect identification.

Comparison of ORI with other identification tools

In the course of this thesis, ORI was compared to two software, Kraken 2 and StrainSeeker, that have in common a methodological background based on a k-mer genome partition inserted in an index. However, these programs were originally developed to identify bacteria from short reads. Then, it would be interesting to compare the identification results of ORI with results from identification software based on the alignment of long erroneous reads such as NanoMAP, MetaMaps. These comparisons were not performed during the thesis because:

- NanoMAP is a new software that did not exist at the time of our experiments.
- MetaMaps was a recent and therefore unstable software at the time of our experiments. Therefore, we did not manage to install it. The authors have been contacted and a conda package allowing a simplified installation has since been made available.

Therefore, these two softwares will be compared to ORI in future identification experiments that we will perform.

To go further with a strain-based annotation

The identification of bacterial strains present in a sample allows for more detailed analyses. For example, the results of the identification of *E. coli* strains presented in section 5.3 were then used by a Masters trainee (Rania Ouazahrou) to reconstruct the metabolic pathways present in the sample from annotation of the genomes of identified strains.

6.2 Short conclusion on *S. thermophilus*

During the thesis, the genomes of 31 strains of *S. thermophilus* were sequenced and assembled. These genomes are added to the reference genomes already present in the databases thus increasing our available knowledge of this species of food interest. Moreover, the comparative study of these strains carried out during the thesis led to the creation of the clustering tree associated with the maximal biclusters $\{strains\} \times \{genes\}$ available in iTOL (see section 3.2.1). This tree allows to have a general view of the species and should allow further comparative studies on *S. thermophilus* genomes.

6.3 Perspectives on increasing the quantity and sensitivity of sequenced ONT long reads

With the advancement of Nanopore technology and especially basecallers, the error rate of reads is decreasing month after month. If this trend continues until it reaches the error rate of the illumina technology, the identification of strains from these reads will

become more accurate. Actually, in order to decrease the error rate of reads, basecalling software can be trained on a sample of reference genomes by using Taiyaki [*Taiyaki* 2021]. It would then be very interesting in our case to train the basecaller to recognize lactic bacteria (by default the ONT basecaller guppy is trained on human and *E.coli* for bacteria) in order to decrease as much as possible the errors in the sequenced reads.
In addition, tools are available that work in real time during sequencing, whether it is at the level of the basecalled reads or the raw signal. In our case, the possibilities offered by ONT sequencing such as Read Until and the washing and reuse of flowcells are two points of interest in the perspective of a fast and cheap identification process. Indeed the Reads Until capacity (see section 1.2.3), associated to a fast identification of the dominant strains of a sample, would allow to artificially amplify during the sequencing the minority strains by rejecting the reads from already identified strains.

Compromise made during the wet lab part

The wet lab part was realized by Emeline Roux during the thesis. This part is also to be taken into account in the bacterial strain identification time. The portability of the MinION sequencer means that it can be used directly in each laboratory (and even outside laboratories [Johnson *et al.* 2017; Castro-Wallace *et al.* 2017]) without having to send the sample to a sequencing platform. This is a big practical improvement for faster identification in situ.
However, the wet lab part may be relatively long to set up and should therefore be optimized. The trade-offs in this part are mainly between time and quality/length of reads as well as production cost.
During our experiments, DNA was extracted via a specific kit for long reads, using mostly ONT's R9.4 flowcell for sequencing (Flongles have also been tried and allow to decrease the cost of sequencing at the expense of a lower number of reads in the output). The extraction of long fragments DNA requires tedious passage times in the gravity column (from 6 to 8 hours, depending on the strain) and is quite expensive. The choice of the extraction method is crucial to reduce costs and times as much as possible while maintaining good ORI performance. A DNA extraction kit designed for short reads and therefore faster and cheaper would tends to fragment more the sampled DNA. A good compromise between speed and quality could however be obtained with a fast DNA extraction with magnetic beads.
Concerning the nanopore library sequencing kit, the rapid barcoding kit was mostly used. This kit reduces a little the size of the reads obtained (factor 3) but also decreases

a lot the preparation time (factor 20) compared to a classical nanopore kit SQK-LSK109. Note that the incubation times for all protocols have been increased to match those of the New England Biolabs (NEB) enzyme supplier
Regarding the sequencing time required by ORI to perform identification, less than 10 minutes of sequencing are sufficient to produce 4,000 reads. Overall, from sample collection to sequence files, for a sample containing 4 strains, it takes less than 6 hours (for a total cost of 200 € maximum using Flongles). From this sequence file, the identification time achieved by ORI is dependent on the size of the index used (about 30 minutes for an index containing all 95 *S. thermophilus* available in the current databases and about 50 minutes for an index containing 1 026 lactic bacterial genomes from *S. thermophilus*, *Lactobacillus delbruecki* and *Bifidobacterium*).

Taking this into account, the long term goal of bacterial identification from nanopore sequencing would be to have an analysis report of the bacterial strains present in a sample directly at the output of the sequencing using a simple laptop computer. This would allow the rapid study of samples in the field and thus properly exploit the advantage brought by the portability of the Oxford MinION nanopore sequencer. The development of ORI is helping to move in this direction.

Appendices

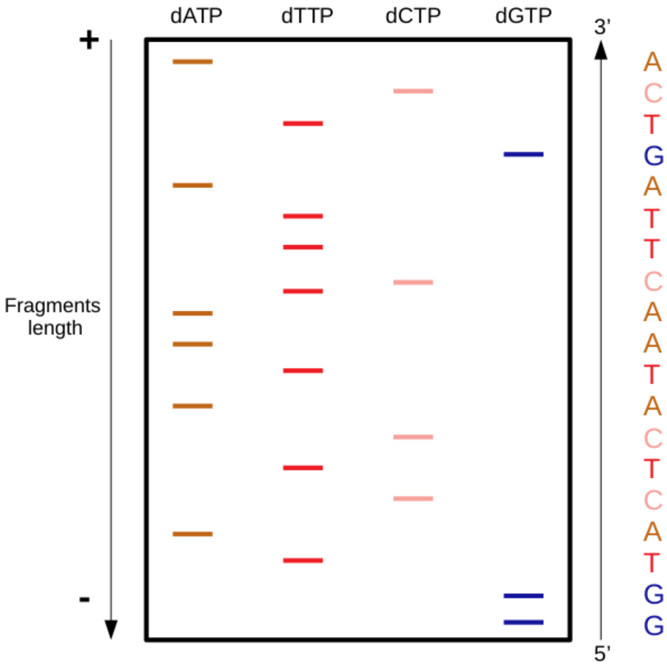


Figure S1: *Simplified diagram showing the fragments obtained by the first Sanger sequencing visible as bands on the polyacrylamide gel.*

Strain	Size (Mb)	GC%	Contig	Status	Accession	WGS	Source	Year	Country	Reference
05-34	1.80032	38.9	43	Scaffold		QFLC01	kefir grains	2010	China	Qin et al., 2011
1F8CT	1.74694	39.1	60	Chromosome	NZ_CM003138	AZTK01	curd from Grana Padano cheese	2012	Italy	Treu et al., 2014c
ACA-DC 2	1.73184	39.2	1	Complete Genome	NZ_LT604076		Greek yogurt		Greece	Alexandraki et al., 2017
APC151	1.83913	39.1	1	Complete Genome	NZ_CP019935		digestive tract of marine fish	2015	Ireland	Linares et al., 2016, 2017
ASCC 1275	1.8455	39.1	1	Complete Genome	NZ_CP006819				Australia	Wu et al., 2014
ATCC 19258	2.0747	38.9	88	Scaffold		PRKU01	culture	2017	Denmark	
B59671	1.82117	39.1	1	Complete Genome	NZ_CP022547		raw milk		USA	Renye et al., 2017
BIO1488	1.74857	39.0	52	Contig		WMLD01				
C106	1.76901	38.9	85	Contig		LGRS01	cheese	2010	Ireland	Wels et al., 2015
CAG:236	1.63196	39.0	66	Scaffold	GCA_000434755.1	CBBT01	distal human gut microbiota	2013		
CNRZ1066	1.79623	39.1	1	Complete Genome	NC_006449		yogurt		France	Bolotin et al., 2004
CS8	1.79166	39.0	1	Complete Genome	NZ_CP016439		rubin	2015	China	
DGCC 7710	1.85121	39.0	1	Complete Genome	CP025216		dairy culture			Hatmaker et al., 2018
EPS	1.81231	39.0	1	Complete Genome	CP025400		milk	2016	China	
FAM 13496	1.83635	38.9	62	Contig		VBTK01	cheese starter culture	1988	Switzerland	
GABA	1.85747	39.1	1	Complete Genome	CP025399		milk	2016	China	
IDCC2201	1.79483	39.2	1	Complete Genome	NZ_CP035306		home-made cheese	2018	South Korea	
JIM 8232	1.9299	38.9	1	Complete Genome	NC_017581		Milk	2002	France	Delorme et al., 2010, 2011
KLDS 3.1003	1.89996	38.9	1	Complete Genome	NZ_CP016877		traditional yogurt culture		Inner Mongolia	Evivie et al., 2017
KLDS3.1012	1.86802	39.2	10	Scaffold		LHSK01	traditional sour milk	2009	Inner Mongolia	Tian et al., 2018
KLDS SM	1.85679	39.1	1	Complete Genome	NZ_CP016026		naturally fermented yogurt		Inner Mongolia	Li et al., 2018; Tian et al., 2018
KU30	1.79076	38.9	43	Contig		QFFS01	milk	2009	China	
LMD-9	1.85637	39.1	3	Complete Genome	NC_008532					Makarova et al., 2006
Plasmid 1	4449 bp	37.0								
Plasmid 2	3361 bp	35.1								
LMG 18311	1.79685	39.1	1	Complete Genome	NC_006448		commercial yogurt	1974	UK	Bolotin et al., 2004
LMG S-29186	1.79829	38.9	42	Contig		NIGT01		2017	Italy	
M17PTZA496	2.06546	38.8	70	Chromosome	NZ_CM002372	AZJT01	Fontina cheese	1996	Italy	Treu et al., 2014a
MN-BM-A01	1.87652	39.1	1	Complete Genome	NZ_CP012588		traditional yogurt block	2012	China	Bai et al., 2016
MN-BM-A02	1.85043	39.0	1	Complete Genome	NZ_CP010999		dairy Fan	2008	China	Shi et al., 2015
MN-ZLW-002	1.84852	39.1	1	Complete Genome	NC_017927		traditional yogurt block		China	Kang et al., 2012
MTCC 5460	1.60936	39.3	143	Contig		ALIK01	fermented milk product (curd)	1984	India	
MTCC 5461	1.6199	39.3	144	Contig		ALIL01	fermented milk product (curd)	1984	India	Prajapati et al., 2013
MTH17CL396	1.82822	38.9	49	Chromosome	NZ_CM002371	AZJS01	Fontina cheese	1996	Italy	Treu et al., 2014a
N4L	1.83175	39.1	1	Complete Genome	NZ_LS974444					Proust et al., 2018
NCTC12958	2.10227	39.0	1	Complete Genome	NZ_LS483339		milk			
ND03	1.83195	39.0	1	Complete Genome	NC_017563		traditional dairy product	2010	China	Sun et al., 2010,2011
ND07	1.86951	39.0	1	Complete Genome	NZ_CP016394		fermented Yak milk	2005	China	
S9	1.78744	39.1	1	Complete Genome	NZ_CP013939		traditional dairy product	2013	China	
SMQ-301	1.86179	39.1	1	Complete Genome	NZ_CP011217		industrial cheese starter	1996		Labrie et al., 2015
ST106	1.85608	39.3	1	Complete Genome	NZ_CP031881		raw milk		USA	Renye et al., 2019
ST109	1.78886	39.2	1	Complete Genome	NZ_CP031545		raw milk		USA	Renye et al., 2019
ST3	1.86506	39.0	1	Complete Genome	NZ_CP017064		commerc. Diet. Supplements		South Korea	
TH1435	1.75346	38.8	36	Chromosome	NZ_CM002369	AYSG01	raw goat milk	2011	Italy	Treu et al., 2014b
TH1436	1.78213	38.8	28	Chromosome	NZ_CM002370	AYTT01	raw goat milk	2011	Italy	Treu et al., 2014b
TH1477	1.88956	38.9	56	Chromosome	NZ_CM003135	AZTJ01	cow milk	2012	Italy	Treu et al., 2014c
TH982	1.78931	38.8	52	Chromosome	NZ_CM003136	AZTL01	buffalo mozzarella curd	2003	Italy	Treu et al., 2014c
TH985	1.84443	39.0	84	Chromosome	NZ_CM003139	AZTM01	buffalo mozzarella whey	2003	Italy	Treu et al., 2014c

Table S1: *Streptococcus thermophilus genomes extracted from public databases.*

Strain	Short name	Size (Mb)	GC%	Contig	Status	Accession	Source	Year	Country
CIRM-BIA1035	CIRM1035	1.81944	39.1	1	Complete Genome	NZ_LR822029	artisanal lactic starter (Gruyere de Comte)	1973	France
CIRM-BIA1046	CIRM1046	1.88980	39.0	1	Complete Genome	NZ_LR822030	yoghurt	1971	Japan
CIRM-BIA1047	CIRM1047	1.81490	39.0	1	Complete Genome	NZ_LR822031	lactic starter (for yoghurt making)	1978	France
CIRM-BIA1048	CIRM1048	1.86536	39.0	1	Complete Genome	NZ_LR822033	lactic starter (for yoghurt making)	1978	Bulgaria
CIRM-BIA1049	CIRM1049	1.86535	39.0	1	Complete Genome	NZ_LR822034	lactic starter	1978	France
CIRM-BIA1050	CIRM1050	1.84723	39.0	1	Complete Genome	NZ_LR822032	yoghurt (ewe milk)	1973	Greece
CIRM-BIA1051	CIRM1051	1.85757	39.1	1	Complete Genome	NZ_LR822035	yoghurt (Tarag)	1974	Mongolia
CIRM-BIA1055	CIRM1055	1.83916	39.1	1	Complete Genome	NZ_LR822036	yoghurt	1990	France
CIRM-BIA1116	CIRM1116	1.80437	39.1	1	Complete Genome	NZ_LR822039	yoghurt	1963	France
CIRM-BIA1121	CIRM1121	1.78180	39.2	2	Complete Genome	NZ_LR822037	lactic starter (for yoghurt making)	1978	France
Plasmid		3530 bp	32.3						
CIRM-BIA1122	CIRM1122	1.80842	39.0	1	Complete Genome	NZ_LR822041	lactic starter	<1979	France
CIRM-BIA1125	CIRM1125	1.76728	39.2	1	Complete Genome	NZ_LR822040	yoghurt	1987	Greece
CIRM-BIA1358	CIRM1358	1.80090	39.1	1	Complete Genome	NZ_LR822042	artisanal lactic starter (Gruyere de Comte)	1974	France
CIRM-BIA16	CIRM16	1.80404	39.0	1	Complete Genome	NZ_LR822006	lactic starter (for yoghurt making)	1954	France
CIRM-BIA18	CIRM18	1.86007	39.1	1	Complete Genome	NZ_LR822008	artisanal lactic starter (Gruyere de Comte)	1964	France
CIRM-BIA19	CIRM19	1.82804	39.0	1	Complete Genome	NZ_LR822009	artisanal lactic starter (Gruyere de Comte)	1963	France
CIRM-BIA2101	CIRM2101	1.78894	39.1	1	Complete Genome		commercial lactic starter	1975	Germany
CIRM-BIA23	CIRM23	1.79608	39.0	1	Complete Genome	NZ_LR822011	yoghurt	<1955	France
CIRM-BIA29	CIRM29	1.80465	39.0	1	Complete Genome	NZ_LR822010	yoghurt	1960	Netherlands
CIRM-BIA30	CIRM30	1.80141	39.1	1	Complete Genome	NZ_LR822012	artisanal lactic starter (Gruyere de Comte)	1963	France
CIRM-BIA32	CIRM32	1.86007	39.1	1	Complete Genome	NZ_LR822013	artisanal lactic starter (Emmental)	1964	France
CIRM-BIA336	CIRM336	1.87117	39.1	2	Complete Genome	NZ_LR822017	milk after contact with traditional sicilian wooden vat (Tina)	2007	Italy
Plasmid		3823 bp	33.4						
CIRM-BIA368	CIRM368	2.02646	38.8	3	Complete Genome	NZ_LR822023	milk after contact with traditional sicilian wooden vat (Tina)	2007	Italy
Plasmid 1		4451 bp	39.4						
Plasmid 2		2162 bp	36.8						
CIRM-BIA36	CIRM36	1.78366	39.1	1	Complete Genome	NZ_LR822014	yoghurt	1971	France
CIRM-BIA65	CIRM65	1.78829	39.1	2	Complete Genome	NZ_LR822015	lactic starter (for yoghurt making)	1971	France
Plasmid		3345 bp	35.5						
CIRM-BIA67	CIRM67	1.77325	39.1	2	Complete Genome	NZ_LR824002	lactic starter (for yoghurt making)	1971	France
Plasmid		3345 bp	35.5						
CIRM-BIA772	CIRM772	1.77249	39.2	1	Complete Genome	NZ_LR822019	fermented milk	1963	France
CIRM-BIA956	CIRM956	1.99901	38.7	3	Complete Genome	NZ_LR822020	cow raw milk	2009	Italy
Plasmid 1		4403 bp	39.4						
Plasmid 2		2162 bp	36.8						
CIRM-BIA961	CIRM961	2.01958	38.9	1	Complete Genome	NZ_LR822025	cow raw milk	2009	Italy
CIRM-BIA967	CIRM967	2.01753	38.9	1	Complete Genome	NZ_LR822026	cow raw milk	2009	Italy
CIRM-BIA998	CIRM998	1.97776	38.9	2	Complete Genome	NZ_LR822027	cow raw milk	2009	Italy
Plasmid		4395 bp	39.3						

Table S2: *Streptococcus thermophilus genomes sequenced from the CIRM-BIA collection.*

Table S3: *S. thermophilus strains identification by ORI, with and without merge index, in a balanced mixture of 4 or 6 strains more or less genetically close, by using 1000, 4000 or 16000 Nanopore sequencing reads.* Best results are in bold. Values of Hamming distance (0 = perfect identification); MCC: Matthews Correlation Coefficient (1 = perfect correlation); ratio: number of strains identified / number of strains present (1 = perfect number); std: standard deviation.

(a) Heterogeneity results (variable number of strains mixed):

Method		**ORI**		**ORI_merge**	
Number of strains		**4**	**6**	**4**	**6**
Distance	**median**	**0**	**0**	**0**	**0**
	mean	0.73	0.31	**0.53**	**0.29**
	std	2.64	**1.13**	**1.78**	1.19
	[min,max]	[0,15]	**[0,7]**	**[0,7]**	**[0,7]**
MCC	**mean**	0.70	0.65	**0.94**	**0.96**
	std	0.18	0.15	**0.10**	**0.08**
Ambiguity	**mean**	0.65	0.56	**0.93**	**0.96**
	std	0.25	0.21	**0.19**	**0.15**

(b) Data quantity (different number of .fastq reads):

Method		**ORI**			**ORI_merge**		
Number of reads		**1000**	**4000**	**16000**	**1000**	**4000**	**16000**
Distance	**median**	**0**	**0**	**0**	**0**	**0**	**0**
	mean	0.17	0.80	**0.60**	**0**	**0.43**	0.80
	std	0.73	2.93	**1.80**	**0**	**1.43**	2.14
	[min,max]	[0,4]	[0,15]	**[0,7]**	**[0,0]**	**[0,7]**	**[0,7]**
MCC	**mean**	0.55	0.64	0.78	**0.86**	**0.93**	**0.98**
	std	0.20	0.18	0.14	**0.14**	**0.12**	**0.05**
Ambiguity	**mean**	0.44	0.64	0.80	**0.77**	**0.92**	**1.05**
	std	**0.17**	0.20	0.20	0.21	**0.12**	**0.13**

(c) Resolution power (proximity between strains within the mixture):

Method		**ORI**			**ORI_merge**		
Proximity		**distant**	**medium**	**close**	**distant**	**medium**	**close**
Distance	**median**	**0**	**0**	**0**	**0**	**0**	**0**
	mean	0.10	1.17	**0.30**	**0**	**0.90**	0.33
	std	0.54	3.16	**1.27**	**0**	**2.17**	1.35
	[min,max]	[0,3]	[0,15]	**[0,7]**	**[0,0]**	**[0,7]**	**[0,7]**
MCC	**mean**	0.75	0.61	0.6	**0.93**	**0.87**	**0.97**
	std	0.17	0.24	0.14	**0.09**	**0.15**	**0.09**
Ambiguity	**mean**	0.73	0.68	0.47	**0.89**	**0.85**	**1**
	std	0.20	0.23	0.19	**0.15**	**0.22**	**0.18**

Table S4: **Subdominant *S. thermophilus* strains identification by ORI, without/with merge in a mixture of 4 or 6 strains, by using 1000, 4000 or 16000 Nanopore sequencing reads.** Best results are in bold. Values of Hamming distance: in all experiments, minimum value is 0 (perfect identification); MCC: Matthews Correlation Coefficient (1 = perfect correlation); Ambiguity ratio: number of strains identified / number of strains present; std: standard deviation.

Number of strains		**4 (ORI/ORI_merge)**			**6 (ORI/ORI_merge)**		
Number of reads		**1000**	**4000**	**16000**	**1000**	**4000**	**16000**
Distance	**median**	**0/0**	**0/0**	**0/0**	3/**0**	**0/0**	**0/0**
	mean	19.8/19.8	9.9/**0**	**0/0**	30.3/**15.4**	26.7/**0**	**0/0**
	std	24.2/24.2	19.8/**0**	**0/0**	35.9/**29.4**	36.3/**0**	**0/0**
	max	49/49	49/**0**	**0/0**	74.2/74.2	74.2/**0**	**0/0**
MCC	**mean**	0.28/**0.38**	0.42/**0.78**	0.57/**0.9**	0.22/**0.38**	0.34/**0.65**	0.63/**0.8**
	std	0.35/**0.32**	0.35/**0.2**	0.34/**0.21**	**0.28**/0.34	0.28/**0.17**	0.16/**0.14**
Ambiguity	**mean**	0.4/0.4	0.5/**0.8**	0.7/**1**	0.2/**0.33**	0.2/**0.47**	0.53/**0.67**
	std	0.37/0.37	0.32/**0.24**	0.24/**0**	**0.16**/0.21	**0.16**/0.27	**0.16**/0.21

Table S5: *Precision of the Streptococcus thermophilus strains identification using ORI with and without clustering of sibling strains for all the experiments.* The clusters are the same as in Table S5. The percentage represents the number of times the strain was correctly identified divided by the number of times it was identified. This corresponds to the precision: TP/(TP+FP) with TP = true positives and FP = false positives.

<table>
<tr><th>Cluster id</th><th>Strain</th><th>Precision without merge (%)</th><th>Precision with merge (%)</th></tr>
<tr><td rowspan="2">2</td><td>CIRM1048</td><td>54.5</td><td rowspan="2">100.0</td></tr>
<tr><td>CIRM1049</td><td>81.2</td></tr>
<tr><td>3</td><td>CIRM1051</td><td>100.0</td><td>100.0</td></tr>
<tr><td>4</td><td>CIRM1055</td><td>100.0</td><td>100.0</td></tr>
<tr><td rowspan="3">5</td><td>CIRM29</td><td>100.0</td><td rowspan="3">91.8</td></tr>
<tr><td>CIRM1116</td><td>100.0</td></tr>
<tr><td>CIRM1122</td><td>100.0</td></tr>
<tr><td rowspan="2">6</td><td>CIRM18</td><td>33.3</td><td rowspan="2">100.0</td></tr>
<tr><td>CIRM32</td><td>80.6</td></tr>
<tr><td rowspan="5">7</td><td>CIRM67</td><td>50.0</td><td rowspan="5">100.0</td></tr>
<tr><td>CIRM36</td><td>60.0</td></tr>
<tr><td>CIRM65</td><td>77.3</td></tr>
<tr><td>CIRM2101</td><td>100.0</td></tr>
<tr><td>CIRM23</td><td>100.0</td></tr>
<tr><td rowspan="2">8</td><td>CIRM967</td><td>100.0</td><td rowspan="2">100.0</td></tr>
<tr><td>CIRM961</td><td>100.0</td></tr>
<tr><td rowspan="15">No cluster</td><td>CIRM1125</td><td>62.5</td><td>76.0</td></tr>
<tr><td>CIRM19</td><td>83.3</td><td>100.0</td></tr>
<tr><td>CIRM368</td><td>89.5</td><td>94.4</td></tr>
<tr><td>CIRM1050</td><td>100.0</td><td>100.0</td></tr>
<tr><td>CIRM30</td><td>100.0</td><td>100.0</td></tr>
<tr><td>CIRM1046</td><td>100.0</td><td>100.0</td></tr>
<tr><td>JIM_8232</td><td>100.0</td><td>100.0</td></tr>
<tr><td>CIRM998</td><td>100.0</td><td>100.0</td></tr>
<tr><td>CIRM772</td><td>100.0</td><td>100.0</td></tr>
<tr><td>CIRM1121</td><td>100.0</td><td>100.0</td></tr>
<tr><td>CIRM1035</td><td>100.0</td><td>100.0</td></tr>
<tr><td>CIRM336</td><td>100.0</td><td>100.0</td></tr>
<tr><td>CIRM1047</td><td>100.0</td><td>100.0</td></tr>
<tr><td>CIRM1358</td><td>100.0</td><td>100.0</td></tr>
<tr><td>CIRM956</td><td>100.0</td><td>100.0</td></tr>
</table>

Table S6: *Precision of the Streptococcus thermophilus subdominant strains identification using ORI without clustering of sibling strains for experiments with dominant and subdominant strains.* The clusters are the same as in Table S5. The percentage represents the number of times the strain was correctly identified divided by the number of times it was identified. This corresponds to the precision: TP/(TP+FP) with TP = true positives and FP = false positives. Unpredicted means that the strain is not identified in any experiment.

Precision without merge (%)				
Cluster id	**Strain**	**1000 reads**	**4000 reads**	**16000 reads**
2	CIRM1048	0.0	0.0	0.0
	CIRM1049	50.0	100.0	33.33
3	CIRM1051	100.0	100.0	100.0
4	CIRM1055	Unpredicted	Unpredicted	Unpredicted
5	CIRM1116	0.0	0.0	0.0
	CIRM1122	Unpredicted	66.67	66.67
	CIRM29	50.0	50.0	33.33
6	CIRM18	0.0	50.0	0.0
	CIRM32	33.33	0.0	42.86
7	CIRM2101	0.0	0.0	0.0
	CIRM23	Unpredicted	0.0	33.33
	CIRM65	0.0	14.29	0.0
	CIRM67	50.0	Unpredicted	Unpredicted
8	CIRM961	40.0	50.0	55.56
	CIRM967	75.0	100.0	100.0
No cluster	CIRM1125	0.0	0.0	0.0
	CIRM1121	0.0	0.0	0.0
	CIRM1050	0.0	0.0	20.0
	CIRM30	0.0	25.0	25.0
	CIRM1047	0.0	25.0	25.0
	CIRM998	0.0	50.0	66.67
	CIRM956	Unpredicted	Unpredicted	100.0
	CIRM1035	Unpredicted	100.0	100.0
	CIRM368	25.0	33.33	33.33
	JIM_8232	40.0	40.0	40.0
	CIRM1046	50.0	33.33	33.33
	CIRM336	66.67	66.67	75.0
	CIRM772	100.0	50.0	75.0

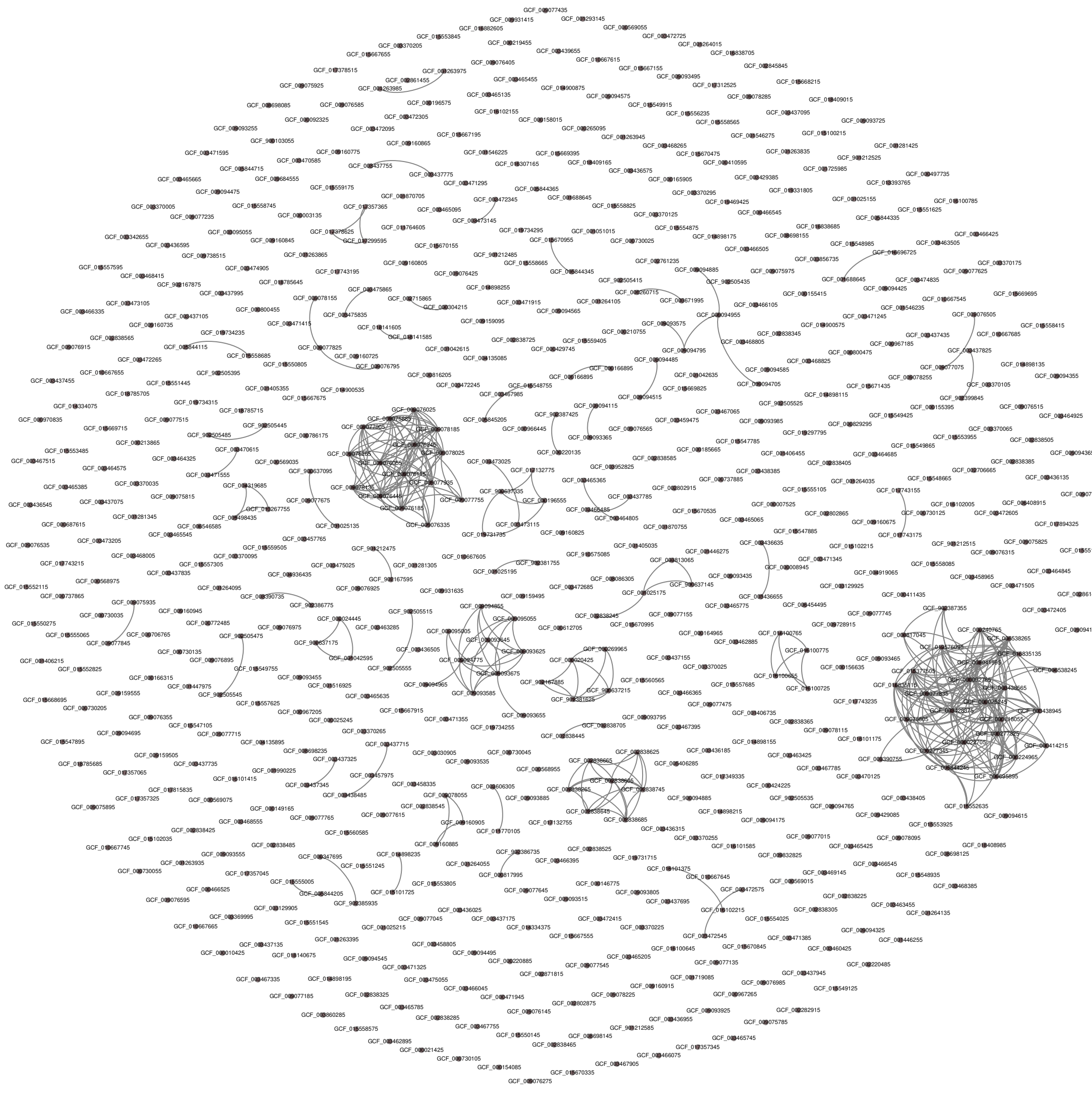


Figure S2: *Graphical representation of the graph representing all existing proximity relationships between the complete genomes of Bifidobacterium with their assembly identifiers.*

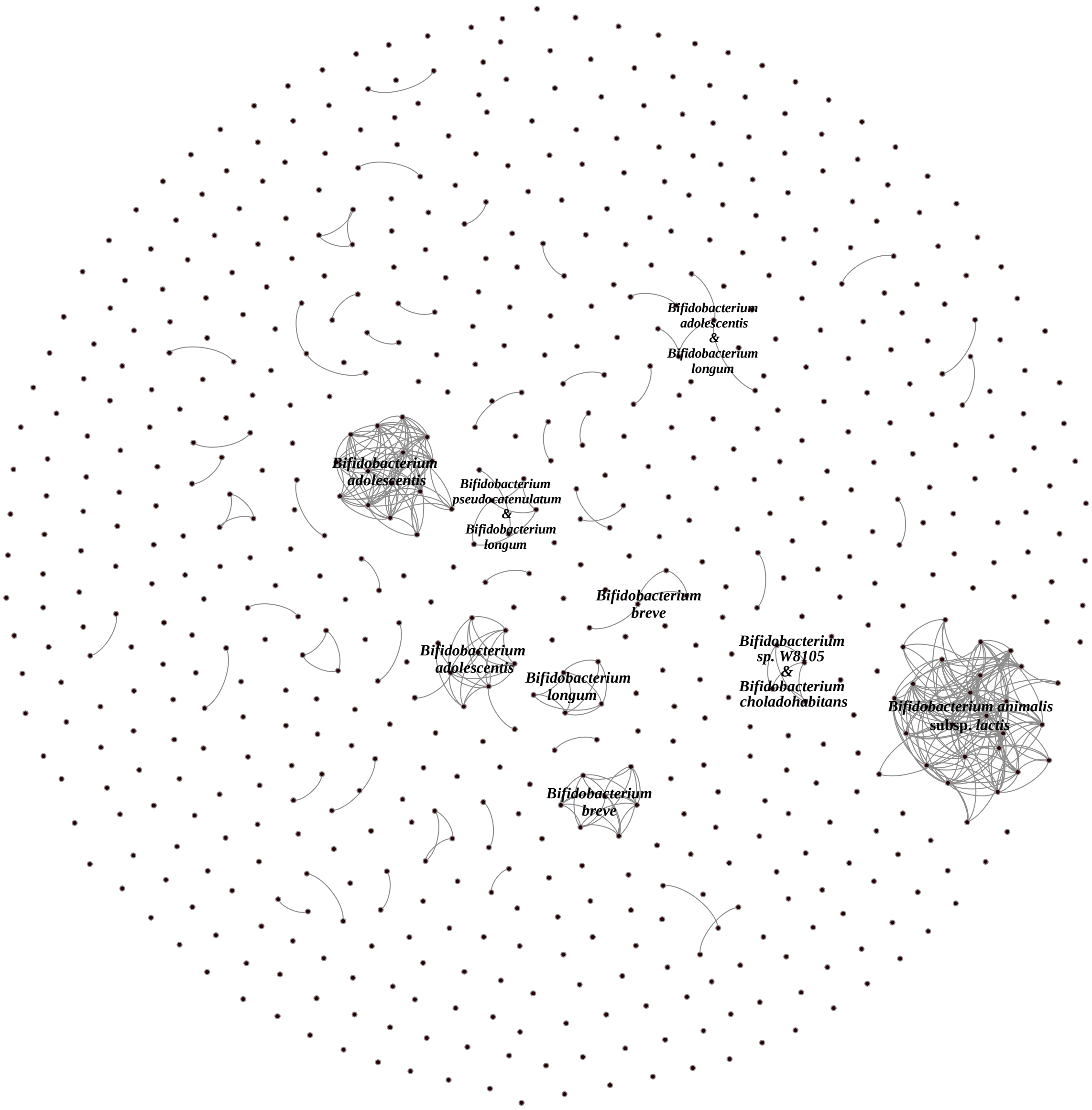


Figure S3: *Graphical representation of the graph representing all existing proximity relationships between the complete genomes of Bifidobacterium. For the related components of more than 4 strains the name of the species present in the cluster is given.*

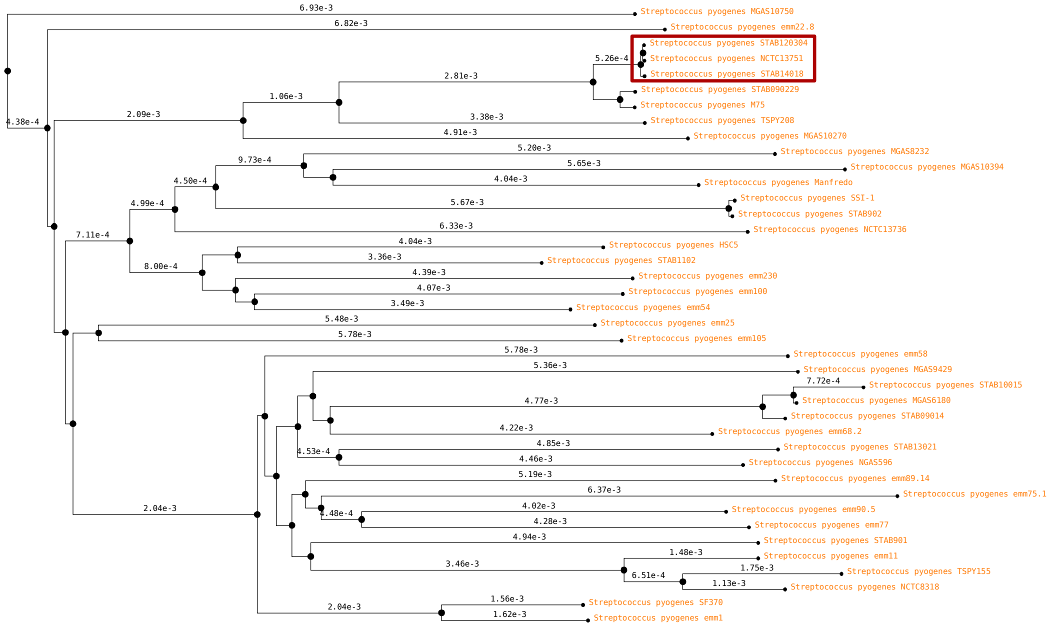


Figure S4: *Clustering tree created from MicroScope presenting 40 strains of S. pyogenes including the cluster identified in the results in section 5.3.*

Bibliography

Abe, T. *et al.* (Apr. 1, 2003). “Informatics for Unveiling Hidden Genome Signatures”. In: *Genome Research* 13.4, pp. 693–702 (cit. on p. 122).

Adessi, C. *et al.* (Oct. 2000). “Solid phase DNA amplification: characterisation of primer attachment and amplification mechanisms”. In: *Nucleic Acids Research* 28.20, e87 (cit. on p. 34).

Ahn, T.-H. *et al.* (Jan. 15, 2015). “Sigma: Strain-level inference of genomes from metagenomic analysis for biosurveillance”. In: *Bioinformatics* 31.2, pp. 170–177 (cit. on p. 55).

Ainsworth, D. *et al.* (Feb. 28, 2017). “k-SLAM: accurate and ultra-fast taxonomic classification and gene identification for large metagenomic data sets”. In: *Nucleic Acids Research* 45.4, pp. 1649–1656 (cit. on p. 71).

Alexandraki, V. *et al.* (2019). “Comparative Genomics of Streptococcus thermophilus Support Important Traits Concerning the Evolution, Biology and Technological Properties of the Species”. In: *Frontiers in Microbiology* 10. Publisher: Frontiers (cit. on pp. 4, 45).

Almodaresi, F., P. Pandey, M. Ferdman, *et al.* (Apr. 1, 2020). “An Efficient, Scalable, and Exact Representation of High-Dimensional Color Information Enabled Using de Bruijn Graph Search”. In: *Journal of Computational Biology* 27.4, pp. 485–499 (cit. on p. 69).

Almodaresi, F., H. Sarkar, *et al.* (July 1, 2018). “A space and time-efficient index for the compacted colored de Bruijn graph”. In: *Bioinformatics* 34.13, pp. i169–i177 (cit. on p. 69).

Almodaresi, F. *et al.* (May 15, 2017). *Rainbowfish: A Succinct Colored de Bruijn Graph Representation.* Company: Cold Spring Harbor Laboratory Distributor: Cold Spring Harbor Laboratory Label: Cold Spring Harbor Laboratory Section: New Results Type: article, p. 138016 (cit. on p. 69).

Altschul, S. F. *et al.* (Oct. 1990). “Basic local alignment search tool”. en. In: *Journal of Molecular Biology* 215.3, pp. 403–410 (cit. on pp. 5, 29, 49, 53).

Ames, S. K. *et al.* (Sept. 15, 2013). “Scalable metagenomic taxonomy classification using a reference genome database”. In: *Bioinformatics* 29.18, pp. 2253–2260 (cit. on p. 71).

Anaconda Software Distribution (2020). Version Vers. 2-2.4.0. URL: https://docs.anaconda.com/ (cit. on p. 115).

Ardui, S. *et al.* (Mar. 2018). “Single molecule real-time (SMRT) sequencing comes of age: applications and utilities for medical diagnostics”. In: *Nucleic Acids Research* 46.5, pp. 2159–2168 (cit. on p. 34).

Au, K. F. *et al.* (Oct. 4, 2012). “Improving PacBio Long Read Accuracy by Short Read Alignment”. In: *PLOS ONE* 7.10. Publisher: Public Library of Science, e46679 (cit. on pp. 74, 82).

Auch, A. F. *et al.* (Jan. 2010). “Digital DNA-DNA hybridization for microbial species delineation by means of genome-to-genome sequence comparison”. In: *Standards in Genomic Sciences* 2.1, pp. 117–134 (cit. on p. 29).

Bao, E. *et al.* (Oct. 15, 2019). “FLAS: fast and high-throughput algorithm for PacBio long-read self-correction”. In: *Bioinformatics* 35.20, pp. 3953–3960 (cit. on p. 74).

Bell, T. *et al.* (Aug. 2005). “The contribution of species richness and composition to bacterial services”. en. In: *Nature* 436.7054 (cit. on p. 20).

Benyoussef, A. (May 24, 2013). *Image depicting an agglomeration of Streptococcus thermophilus price chains using a ZEISS LSM 780 confocal microscope (with scale)* (cit. on p. 44).

Bingmann, T. *et al.* (2019). "COBS: A Compact Bit-Sliced Signature Index". In: *String Processing and Information Retrieval*. Ed. by N. R. Brisaboa and S. J. Puglisi. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 285–303 (cit. on pp. 62, 69).

Bispo, P. J. M. *et al.* (Feb. 1, 2011). "Detection and Gram Discrimination of Bacterial Pathogens from Aqueous and Vitreous Humor Using Real-Time PCR Assays". In: *Investigative Ophthalmology & Visual Science* 52.2. Publisher: The Association for Research in Vision and Ophthalmology, pp. 873–881 (cit. on p. 22).

Blattner, F. R. *et al.* (Sept. 1997). "The Complete Genome Sequence of Escherichia coli K-12". en. In: *Science* 277.5331. Publisher: American Association for the Advancement of Science Section: Articles, pp. 1453–1462 (cit. on p. 24).

Bocs, S. *et al.* (July 1, 2003). "AMIGene: Annotation of MIcrobial Genes". In: *Nucleic Acids Research* 31.13, pp. 3723–3726 (cit. on p. 91).

Bodor, A. *et al.* (Mar. 2020). "Challenges of unculturable bacteria: environmental perspectives". en. In: *Reviews in Environmental Science and Bio/Technology* 19.1, pp. 1–22 (cit. on pp. 22, 24).

Bolotin, A. *et al.* (Dec. 2004). "Complete sequence and comparative genome analysis of the dairy bacterium Streptococcus thermophilus". In: *Nature Biotechnology* 22.12, pp. 1554–1558 (cit. on p. 44).

Boucher, C. *et al.* (Apr. 2015). "Variable-Order de Bruijn Graphs". In: *2015 Data Compression Conference*. 2015 Data Compression Conference. ISSN: 2375-0359, pp. 383–392 (cit. on pp. 5, 58).

Bowe, A. *et al.* (2012). "Succinct de Bruijn Graphs". In: *Algorithms in Bioinformatics*. Ed. by B. Raphael and J. Tang. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 225–235 (cit. on p. 58).

Bradley, P. *et al.* (Feb. 2019). "Ultra-fast search of all deposited bacterial and viral genomic data". In: *Nature biotechnology* 37.2, pp. 152–159 (cit. on pp. 62, 69).

Breitwieser, F. P. *et al.* (Nov. 16, 2018). "KrakenUniq: confident and fast metagenomics classification using unique k-mer counts". In: *Genome Biology* 19.1, p. 198 (cit. on p. 71).

Breitwieser, F. P. *et al.* (July 19, 2019). "A review of methods and databases for metagenomic classification and assembly". In: *Briefings in Bioinformatics* 20.4, pp. 1125–1136 (cit. on p. 49).

Břinda, K., A. Callendrello, *et al.* (Mar. 2020). "Rapid inference of antibiotic resistance and susceptibility by genomic neighbour typing". In: *Nature Microbiology* 5.3, pp. 455–464 (cit. on p. 156).

Břinda, K. *et al.* (Nov. 15, 2015). "Spaced seeds improve k-mer-based metagenomic classification". In: *Bioinformatics* 31.22. Publisher: Oxford Academic, pp. 3584–3592 (cit. on pp. 8, 83).

Brisson, V. L. *et al.* (Sept. 2012). "Metagenomic analysis of a stable trichloroethene-degrading microbial community". In: *The ISME Journal* 6.9, pp. 1702–1714 (cit. on p. 121).

Broder, A. (June 1997). "On the resemblance and containment of documents". In: pp. 21–29 (cit. on p. 89).

Buchfink, B. *et al.* (Jan. 2015). "Fast and sensitive protein alignment using DIAMOND". In: *Nature Methods* 12.1, pp. 59–60 (cit. on p. 53).

Burkhardt, S. and J. Kärkkäinen (Jan. 1, 2003). "Better Filtering with Gapped q-Grams". In: *Fundamenta Informaticae* 56.1, pp. 51–70 (cit. on p. 83).

Cai, H. *et al.* (Sept. 2003). "16S Ribosomal RNA Sequence—Based Identification of Veterinary Clinical Bacteria". en. In: *Journal of Veterinary Diagnostic Investigation* 15.5. Publisher: SAGE Publications Inc, pp. 465–469 (cit. on p. 28).

Caporaso, J. G. *et al.* (May 2010). "QIIME allows analysis of high-throughput community sequencing data". In: *Nature Methods* 7.5, pp. 335–336 (cit. on p. 28).

Carapetis, J. R. *et al.* (Nov. 2005). "The global burden of group A streptococcal diseases". In: *The Lancet. Infectious Diseases* 5.11, pp. 685–694 (cit. on p. 148).

Carbone, A. *et al.* (Mar. 1, 2005). "Codon Bias Signatures, Organization of Microorganisms in Codon Space, and Lifestyle". In: *Molecular Biology and Evolution* 22.3, pp. 547–561 (cit. on p. 25).

Carding, S. *et al.* (Feb. 2015). "Dysbiosis of the gut microbiota in disease". In: *Microbial Ecology in Health and Disease* 26 (cit. on p. 20).

Caro-Quintero, A. and K. T. Konstantinidis (Feb. 2012). "Bacterial species may exist, metagenomics reveal". In: *Environmental Microbiology* 14.2, pp. 347–355 (cit. on p. 30).

Castro-Wallace, S. L. *et al.* (Dec. 21, 2017). "Nanopore DNA Sequencing and Genome Assembly on the International Space Station". In: *Scientific Reports* 7.1, p. 18022 (cit. on p. 160).

Check Hayden, E. (Feb. 6, 2009). "Genome sequencing: the third generation". In: *Nature* (cit. on p. 36).

Chen, K. *et al.* (Nov. 10, 2009). "An efficient way of finding good indel seeds for local homology search". In: *Chinese Science Bulletin* 54.20, p. 3837 (cit. on p. 86).

Chen, L.-L. and C.-T. Zhang (June 20, 2003). "Seven GC-rich microbial genomes adopt similar codon usage patterns regardless of their phylogenetic lineages". In: *Biochemical and Biophysical Research Communications* 306.1, pp. 310–317 (cit. on p. 25).

Chikhi, R. *et al.* (Mar. 8, 2021). "Data Structures to Represent a Set of k-long DNA Sequences". In: *ACM Computing Surveys* 54.1, 17:1–17:22 (cit. on p. 57).

Chu, J., H. Mohamadi, *et al.* (July 21, 2020). "Mismatch-tolerant, alignment-free sequence classification using multiple spaced seeds and multiindex Bloom filters". In: *Proceedings of the National Academy of Sciences* 117.29. ISBN: 9781903436110 Publisher: National Academy of Sciences Section: PNAS Plus, pp. 16961–16968 (cit. on pp. 62, 63, 69, 84).

Chu, J., S. Sadeghi, *et al.* (Dec. 1, 2014). "BioBloom tools: fast, accurate and memory-efficient host species sequence screening using bloom filters". In: *Bioinformatics* 30.23, pp. 3402–3404 (cit. on pp. 62, 63, 69).

Chun, J. and F. A. 2. Rainey (n.d.). "Integrating genomics into the taxonomy and systematics of the Bacteria and Archaea". In: *International Journal of Systematic and Evolutionary Microbiology* 64.Pt_2 (). Publisher: Microbiology Society, pp. 316–324 (cit. on pp. 28, 29).

Collins, R. E. and P. G. Higgs (Nov. 2012). "Testing the infinitely many genes model for the evolution of the bacterial core genome and pangenome". In: *Molecular Biology and Evolution* 29.11, pp. 3413–3425 (cit. on p. 88).

Conway, T. and P. S. Cohen (June 2015). "Commensal and Pathogenic Escherichia coli Metabolism in the Gut". In: *Microbiology spectrum* 3.3, 10.1128/microbiolspec.MBP–0006–2014 (cit. on p. 20).

Da Silva, K. D. *et al.* (Feb. 13, 2021). "StrainFLAIR: Strain-level profiling of metagenomic samples using variation graphs". In: *bioRxiv*. Publisher: Cold Spring Harbor Laboratory Section: New Results, p. 2021.02.12.430979 (cit. on p. 73).

Dadi, T. H. *et al.* (Sept. 1, 2018). "DREAM-Yara: an exact read mapper for very large databases with short update time". In: *Bioinformatics* 34.17, pp. i766–i772 (cit. on pp. 63, 69).

Darling, A. E. *et al.* (Jan. 9, 2014). "PhyloSift: phylogenetic analysis of genomes and metagenomes". In: *PeerJ* 2, e243 (cit. on p. 55).

De MAN, J. C. *et al.* (1960). "A Medium for the Cultivation of Lactobacilli". In: *Journal of Applied Bacteriology* 23.1. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1365-2672.1960.tb00188.x, pp. 130–135 (cit. on p. 127).

De Vuyst, L. *et al.* (Dec. 2002). "The Biodiversity of Lactic Acid Bacteria in Greek Traditional Wheat Sourdoughs Is Reflected in Both Composition and Metabolite Formation". In: *Applied and Environmental Microbiology* 68.12. Publisher: American Society for Microbiology, pp. 6059–6069 (cit. on p. 21).

Delahaye, C. and J. Nicolas (Oct. 1, 2021). "Sequencing DNA with nanopores: Troubles and biases". In: *PLOS ONE* 16.10. Publisher: Public Library of Science, e0257521 (cit. on pp. 39, 81, 82, 97).

Delorme, C., C. Bartholini, *et al.* (Oct. 2011). "Complete genome sequence of the pigmented Streptococcus thermophilus strain JIM8232". eng. In: *Journal of Bacteriology* 193.19, pp. 5581–5582 (cit. on pp. 31, 45).

Delorme, C., N. Legravet, *et al.* (Feb. 2, 2017). "Study of Streptococcus thermophilus population on a world-wide and historical collection by a new MLST scheme". In: *International Journal of Food Microbiology* 242, pp. 70–81 (cit. on pp. 28, 45).

Denamur, E. *et al.* (Jan. 2021). "The population genetics of pathogenic Escherichia coli". In: *Nature Reviews Microbiology* 19.1, pp. 37–54 (cit. on p. 45).

Devaere, M. *et al.* (Mar. 12, 2020). "Complete Genome Sequences of Two Strains of Streptococcus pyogenes Belonging to an Emergent Clade of the Genotype emm89 in Brittany, France". In: *Microbiology Resource Announcements* 9.11, e00129–20 (cit. on p. 147).

Dijk, E. L. van *et al.* (Sept. 2018). “The Third Revolution in Sequencing Technology”. eng. In: *Trends in genetics: TIG* 34.9, pp. 666–681 (cit. on pp. 36, 37, 42).

Dijkshoorn, L. *et al.* (2000). “Strain, clone and species: comments on three basic concepts of bacteriology”. In: *Journal of Medical Microbiology* 49.5. Publisher: Microbiology Society, pp. 397–401 (cit. on pp. 30, 32).

Dilthey, A. T. *et al.* (July 11, 2019). “Strain-level metagenomic assignment and compositional estimation for long reads with MetaMaps”. In: *Nature Communications* 10.1, p. 3066 (cit. on pp. 7, 75).

Dohm, J. C. *et al.* (May 25, 2020). “Benchmarking of long-read correction methods”. In: *NAR Genomics and Bioinformatics* 2.2, lqaa037 (cit. on p. 42).

Doolittle, W. F. (June 5, 2012). “Population Genomics: How Bacterial Species Form and Why They Don’t Exist”. In: *Current Biology* 22.11, R451–R453 (cit. on p. 30).

Dröge, J. *et al.* (Mar. 15, 2015). “Taxator-tk: precise taxonomic assignment of metagenomes by fast approximation of evolutionary neighborhoods”. In: *Bioinformatics* 31.6, pp. 817–824 (cit. on p. 55).

Eddy, S. R. (1998). “Profile hidden Markov models”. In: *Bioinformatics (Oxford, England)* 14.9, pp. 755–763 (cit. on p. 53).

Edgar, R. C. (July 2018). “Updating the 97% identity threshold for 16S ribosomal RNA OTUs”. eng. In: *Bioinformatics (Oxford, England)* 34.14, pp. 2371–2375 (cit. on p. 28).

Eid, J. *et al.* (Jan. 2, 2009). “Real-time DNA sequencing from single polymerase molecules”. In: *Science (New York, N.Y.)* 323.5910, pp. 133–138 (cit. on p. 37).

El Soda, M. A. (Sept. 1, 1993). “The role of lactic acid bacteria in accelerated cheese ripening”. In: *FEMS Microbiology Reviews* 12.1, pp. 239–251 (cit. on p. 43).

Elias, P. (Apr. 1, 1974). “Efficient Storage and Retrieval by Content and Address of Static Files”. In: *Journal of the ACM* 21.2, pp. 246–260 (cit. on p. 61).

FDA (2002). *GRN No. 49 Bifidobacterium lactis strain Bb12 and Streptococcus thermophilus strain Th4*. GRAS Notice inventory (cit. on p. 44).

— (2012). *GRN No. 378 Cultured [dairy sources, sugars, wheat, malt, and fruit- and vegetable-based sources] fermented by [Streptococcus thermophilus, Bacillus coagulans, Lactobacillus acidophilus, Lactobacillus paracasei subsp. paracasei, Lactobacillus plantarum, Lactobacillus sakei, Lactobacillus bulgaricus and Proprionibacterium freudenreichii subsp. shermanii or mixtures of these strains]*. GRAS Notice inventory (cit. on p. 44).

Fedurco, M. *et al.* (Feb. 2006). “BTA, a novel reagent for DNA attachment on glass and efficient generation of solid-phase amplified DNA colonies”. In: *Nucleic Acids Research* 34.3, e22–e22 (cit. on p. 34).

Ferragina, P. and G. Manzini (Nov. 2000). “Opportunistic data structures with applications”. In: *Proceedings 41st Annual Symposium on Foundations of Computer Science*. Proceedings 41st Annual Symposium on Foundations of Computer Science. ISSN: 0272-5428, pp. 390–398 (cit. on pp. 5, 49–51).

Ferragina, P. and G. Manzini (July 1, 2005). “Indexing compressed text”. In: *Journal of the ACM* 52.4, pp. 552–581 (cit. on p. 52).

Firtina, C. *et al.* (Nov. 30, 2018). “Hercules: a profile HMM-based hybrid error correction algorithm for long reads”. In: *Nucleic Acids Research* 46.21, e125–e125 (cit. on p. 74).

Francis, O. E. *et al.* (Oct. 2013). “Pathoscope: species identification and strain attribution with unassembled sequencing data”. In: *Genome Research* 23.10, pp. 1721–1729 (cit. on p. 55).

Freitas, T. A. K. *et al.* (May 26, 2015). “Accurate read-based metagenome characterization using a hierarchical suite of unique signatures”. In: *Nucleic Acids Research* 43.10, e69 (cit. on pp. 54, 55).

Fu, S. *et al.* (Feb. 4, 2019). “A comparative evaluation of hybrid error correction methods for error-prone long reads”. In: *Genome Biology* 20.1, p. 26 (cit. on p. 73).

Galia, W. *et al.* (Sept. 1, 2016). “Acquisition of PrtS in Streptococcus thermophilus is not enough in certain strains to achieve rapid milk acidification”. In: *Dairy Science & Technology* 96.5, pp. 623–636 (cit. on p. 31).

Garrison, E. *et al.* (Oct. 2018). “Variation graph toolkit improves read mapping by representing genetic variation in the reference”. In: *Nature Biotechnology* 36.9, pp. 875–879 (cit. on p. 72).

Gautreau, G. *et al.* (Mar. 19, 2020). "PPanGGOLiN: Depicting microbial diversity via a partitioned pangenome graph". In: *PLOS Computational Biology* 16.3. Publisher: Public Library of Science, e1007732 (cit. on p. 88).

Gebser, M. *et al.* (Dec. 15, 2012). "Answer Set Solving in Practice". In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6.3. Publisher: Morgan & Claypool Publishers, pp. 1–238 (cit. on pp. 9, 91).

Girotto, S. *et al.* (Nov. 30, 2018a). "Efficient computation of spaced seed hashing with block indexing". In: *BMC Bioinformatics* 19.15, p. 441 (cit. on p. 84).

— (Mar. 22, 2018b). "FSH: fast spaced seed hashing exploiting adjacent hashes". In: *Algorithms for Molecular Biology* 13.1, p. 8 (cit. on p. 84).

Gori, A. *et al.* (June 9, 2020). "Pan-GWAS of Streptococcus agalactiae Highlights Lineage-Specific Genes Associated with Virulence and Niche Adaptation". In: *mBio* 11.3, e00728–20 (cit. on p. 156).

Goris, J. *et al.* (n.d.). "DNA–DNA hybridization values and their relationship to whole-genome sequence similarities". In: *International Journal of Systematic and Evolutionary Microbiology* 57.1 (). Publisher: Microbiology Society, pp. 81–91 (cit. on pp. 26, 29, 88).

Greig, D. R. *et al.* (Oct. 2018). "MinION nanopore sequencing identifies the position and structure of bacterial antibiotic resistance determinants in a multidrug-resistant strain of enteroaggregative Escherichia coli". In: *Microbial Genomics* 4.10 (cit. on p. 157).

Grozdanov, L. *et al.* (Aug. 2004). "Analysis of the Genome Structure of the Nonpathogenic Probiotic Escherichia coli Strain Nissle 1917". In: *Journal of Bacteriology* 186.16. Publisher: American Society for Microbiology, pp. 5432–5441 (cit. on pp. 2, 31).

Gupta, G. *et al.* (July 17, 2020). "RAMBO: Repeated And Merged BloOm Filter for Ultra-fast Multiple Set Membership Testing (MSMT) on Large-Scale Data". In: *arXiv:1910.02611 [cs]*. arXiv: `1910.02611` (cit. on pp. 63, 69).

Hackl, T. *et al.* (Nov. 1, 2014). "proovread : large-scale high-accuracy PacBio correction through iterative short read consensus". In: *Bioinformatics* 30.21, pp. 3004–3011 (cit. on p. 74).

Haghshenas, E. *et al.* (Sept. 1, 2016). "CoLoRMap: Correcting Long Reads by Mapping short reads". In: *Bioinformatics* 32.17, pp. i545–i551 (cit. on p. 74).

Hahn, L. *et al.* (Oct. 19, 2016). "rasbhari: Optimizing Spaced Seeds for Database Searching, Read Mapping and Alignment-Free Sequence Comparison". In: *PLOS Computational Biology* 12.10. Publisher: Public Library of Science, e1005107 (cit. on p. 85).

Hahn, M. W. *et al.* (Mar. 2003). "Isolation of Novel Ultramicrobacteria Classified as Actinobacteria from Five Freshwater Habitats in Europe and Asia". In: *Applied and Environmental Microbiology* 69.3, pp. 1442–1451 (cit. on p. 20).

Hall, G. A. *et al.* (Oct. 19, 2020). *Strain-level sample characterisation using long reads and MAPQ scores*. Company: Cold Spring Harbor Laboratory Distributor: Cold Spring Harbor Laboratory Label: Cold Spring Harbor Laboratory Section: New Results Type: article, p. 2020.10.18.344739 (cit. on pp. 7, 75).

Harris, R. S. and P. Medvedev (Feb. 1, 2020). "Improved representation of sequence bloom trees". In: *Bioinformatics (Oxford, England)* 36.3, pp. 721–727 (cit. on pp. 6, 63, 66, 69).

Herms, I. and S. Rahmann (2008). "Computing Alignment Seed Sensitivity with Probabilistic Arithmetic Automata". In: *Algorithms in Bioinformatics*. Ed. by K. A. Crandall and J. Lagergren. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 318–329 (cit. on p. 84).

Holley, G. and P. Melsted (Sept. 17, 2020). "Bifrost: highly parallel construction and indexing of colored and compacted de Bruijn graphs". In: *Genome Biology* 21.1, p. 249 (cit. on p. 69).

Holley, G. *et al.* (Apr. 14, 2016). "Bloom Filter Trie: an alignment-free and reference-free data structure for pan-genome storage". In: *Algorithms for Molecular Biology* 11.1, p. 3 (cit. on pp. 61, 69).

Hols, P. *et al.* (Aug. 1, 2005). "New insights in the molecular biology and physiology of Streptococcus thermophilus revealed by comparative genomics". In: *FEMS Microbiology Reviews* 29.3, pp. 435–463 (cit. on p. 44).

Hoon, M. de *et al.* (June 12, 2004). "Open source clustering software". In: *Bioinformatics* 20.9, pp. 1453–1454 (cit. on p. 63).

Hu, T. *et al.* (2020). "Genome Analysis and Physiological Characterization of Four Streptococcus thermophilus Strains Isolated From Chinese Traditional Fermented Milk". In: *Frontiers in Microbiology* 11. Publisher: Frontiers (cit. on pp. 4, 45).

Huson, D. H., B. Albrecht, *et al.* (Apr. 20, 2018). "MEGAN-LR: new algorithms allow accurate binning and easy interactive exploration of metagenomic long reads and contigs". In: *Biology Direct* 13.1, p. 6 (cit. on pp. 7, 74).

Huson, D. H., S. Beier, *et al.* (June 21, 2016). "MEGAN Community Edition - Interactive Exploration and Analysis of Large-Scale Microbiome Sequencing Data". In: *PLoS Computational Biology* 12.6, e1004957 (cit. on p. 55).

Ignatov, D. I. (2015). *Introduction to Formal Concept Analysis and Its Applications in Information Retrieval and Related Fields.* Ed. by P. Braslavski *et al.* Communications in Computer and Information Science. Cham: Springer International Publishing, pp. 42–141 (cit. on pp. 8, 91).

Iqbal, Z. *et al.* (Feb. 2012). "De novo assembly and genotyping of variants using colored de Bruijn graphs". In: *Nature Genetics* 44.2, pp. 226–232 (cit. on p. 61).

Jackson, D. A. *et al.* (Oct. 1972). "Biochemical Method for Inserting New Genetic Information into DNA of Simian Virus 40: Circular SV40 DNA Molecules Containing Lambda Phage Genes and the Galactose Operon of Escherichia coli". In: *Proceedings of the National Academy of Sciences* 69.10, pp. 2904–2909 (cit. on p. 34).

Jain, C., A. Dilthey, *et al.* (2017). "A Fast Approximate Algorithm for Mapping Long Reads to Large Reference Databases". In: *Research in Computational Molecular Biology.* Ed. by S. C. Sahinalp. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 66–81 (cit. on pp. 75, 93).

Jain, C., L. M. Rodriguez-R, *et al.* (Nov. 30, 2018). "High throughput ANI analysis of 90K prokaryotic genomes reveals clear species boundaries". In: *Nature Communications* 9.1, p. 5114 (cit. on pp. 9, 93).

Jain, M., I. T. Fiddes, *et al.* (Apr. 2015). "Improved data analysis for the MinION nanopore sequencer". In: *Nature Methods* 12.4, pp. 351–356 (cit. on pp. 3, 38).

Jain, M., S. Koren, *et al.* (Apr. 2018). "Nanopore sequencing and assembly of a human genome with ultra-long reads". In: *Nature Biotechnology* 36.4, pp. 338–345 (cit. on p. 39).

Jenkins, C. *et al.* (Apr. 2012). "Detection and identification of bacteria in clinical samples by 16S rRNA gene sequencing: comparison of two different approaches in clinical practice". en. In: *Journal of Medical Microbiology* 61.4, pp. 483–488 (cit. on p. 28).

Johnson, S. S. *et al.* (Apr. 2017). "Real-Time DNA Sequencing in the Antarctic Dry Valleys Using the Oxford Nanopore Sequencer". In: *Journal of Biomolecular Techniques : JBT* 28.1, pp. 2–7 (cit. on p. 160).

Junjua, M. *et al.* (July 2016). "A large scale in vitro screening of Streptococcus thermophilus strains revealed strains with a high anti-inflammatory potential". en. In: *LWT* 70, pp. 78–87 (cit. on pp. 31, 45, 98).

Karasikov, M. *et al.* (Jan. 17, 2019). *Sparse Binary Relation Representations for Genome Graph Annotation.* Company: Cold Spring Harbor Laboratory Distributor: Cold Spring Harbor Laboratory Label: Cold Spring Harbor Laboratory Section: New Results Type: article, p. 468512 (cit. on p. 69).

Karsenti, E. *et al.* (Oct. 2011). "A Holistic Approach to Marine Eco-Systems Biology". en. In: *PLOS Biology* 9.10. Publisher: Public Library of Science, e1001177 (cit. on p. 32).

Kiełbasa, S. M. *et al.* (Mar. 2011). "Adaptive seeds tame genomic sequence comparison". In: *Genome Research* 21.3, pp. 487–493 (cit. on pp. 49, 53).

Kiliç, A. O. *et al.* (June 1, 1996). "Analysis of Lactobacillus phages and bacteriocins in American dairy products and characterization of a phage isolated from yogurt". In: *Applied and Environmental Microbiology* 62.6. Publisher: American Society for Microbiology, pp. 2111–2116 (cit. on p. 44).

Kim, D. *et al.* (Dec. 2016). "Centrifuge: rapid and sensitive classification of metagenomic sequences". In: *Genome Research* 26.12, pp. 1721–1729 (cit. on pp. 7, 54, 55).

Kim, M. *et al.* (Feb. 1, 2014). "Towards a taxonomic coherence between average nucleotide identity and 16S rRNA gene sequence similarity for species demarcation of prokaryotes". In: *International*

Journal of Systematic and Evolutionary Microbiology 64 (Pt_2). Publisher: Microbiology Society, pp. 346–351 (cit. on p. 28).

Koh, D. W.-S. *et al.* (Apr. 2021). "Yet Another Quick Assembly, Analysis and Trimming Tool (YAQAAT): A Server for the Automated Assembly and Analysis of Sanger Sequencing Data". In: *Journal of Biomolecular Techniques : JBT* 32.1, pp. 10–14 (cit. on p. 34).

Konstantinidis, K. T. and J. M. Tiedje (Feb. 2005). "Genomic insights that advance the species definition for prokaryotes". eng. In: *Proceedings of the National Academy of Sciences of the United States of America* 102.7, pp. 2567–2572 (cit. on pp. 3, 29).

Koonin, E. V. and Y. I. Wolf (Dec. 2008). "Genomics of bacteria and archaea: the emerging dynamic view of the prokaryotic world". In: *Nucleic Acids Research* 36.21, pp. 6688–6719 (cit. on p. 88).

Koren, S. *et al.* (Mar. 15, 2017). "Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation". In: *Genome Research*. Company: Cold Spring Harbor Laboratory Press Distributor: Cold Spring Harbor Laboratory Press Institution: Cold Spring Harbor Laboratory Press Label: Cold Spring Harbor Laboratory Press Publisher: Cold Spring Harbor Lab, gr.215087.116 (cit. on p. 74).

Kucherov, G. *et al.* (Apr. 1, 2006). "A unifying framework for seed sensitivity and its application to subset seeds". In: *Journal of Bioinformatics and Computational Biology* 04.2. Publisher: World Scientific Publishing Co., pp. 553–569 (cit. on p. 83).

Kumar, N. *et al.* (Jan. 2015). "Bacterial genospecies that are not ecologically coherent: population genomics of Rhizobium leguminosarum". In: *Open Biology* 5.1, p. 140133 (cit. on p. 30).

Langmead, B. and S. L. Salzberg (Mar. 4, 2012). "Fast gapped-read alignment with Bowtie 2". In: *Nature methods* 9.4, pp. 357–359 (cit. on pp. 52, 53).

Lannoy, C. de *et al.* (2017). "The long reads ahead: de novo genome assembly using the MinION". In: *F1000Research* 6, p. 1083 (cit. on p. 38).

Larsen, M. V. *et al.* (Apr. 2012). "Multilocus Sequence Typing of Total-Genome-Sequenced Bacteria". In: *Journal of Clinical Microbiology* 50.4, pp. 1355–1361 (cit. on p. 28).

Lawrence, J. G. and H. Ochman (Apr. 1, 1997). "Amelioration of Bacterial Genomes: Rates of Change and Exchange". In: *Journal of Molecular Evolution* 44.4, pp. 383–397 (cit. on p. 25).

Lee, H. *et al.* (June 18, 2014). *Error correction and assembly complexity of single molecule sequencing reads*. Company: Cold Spring Harbor Laboratory Distributor: Cold Spring Harbor Laboratory Label: Cold Spring Harbor Laboratory Section: New Results Type: article, p. 006395 (cit. on p. 73).

Lee, J.-Y. *et al.* (Dec. 2017). "Comparative genomics of Lactobacillus salivarius strains focusing on their host adaptation". In: *Microbiological Research* 205, pp. 48–58 (cit. on p. 54).

Légifrance (1988). *Décret n°88-1203 du 30 décembre 1988 relatif aux laits fermentés et au yaourt ou yoghourt* (cit. on p. 141).

Leimeister, C.-A. *et al.* (July 15, 2014). "Fast alignment-free sequence comparison using spaced-word frequencies". In: *Bioinformatics* 30.14, pp. 1991–1999 (cit. on pp. 83, 84).

Lemane, T. *et al.* (Feb. 17, 2021). *kmtricks: Efficient construction of Bloom filters for large sequencing data collections*. Company: Cold Spring Harbor Laboratory Distributor: Cold Spring Harbor Laboratory Label: Cold Spring Harbor Laboratory Section: New Results Type: article, p. 2021.02.16.429304 (cit. on p. 155).

Letunic, I. and P. Bork (July 2, 2019). "Interactive Tree Of Life (iTOL) v4: recent updates and new developments". In: *Nucleic Acids Research* 47 (W1), W256–W259 (cit. on pp. 9, 92).

Levene, M. J. *et al.* (Jan. 31, 2003). "Zero-mode waveguides for single-molecule analysis at high concentrations". In: *Science (New York, N.Y.)* 299.5607, pp. 682–686 (cit. on p. 37).

Li, H. (Sept. 15, 2018). "Minimap2: pairwise alignment for nucleotide sequences". In: *Bioinformatics* 34.18, pp. 3094–3100 (cit. on p. 74).

Li, H. and R. Durbin (July 15, 2009). "Fast and accurate short read alignment with Burrows-Wheeler transform". In: *Bioinformatics (Oxford, England)* 25.14, pp. 1754–1760 (cit. on pp. 52, 53).

Li, Z. *et al.* (Feb. 22, 2018). "Complete Genome Sequences of Two Porcine Enterotoxigenic Escherichia coli Strains". In: *Genome Announcements* 6.8, e00059–18 (cit. on pp. 16, 150, 151).

Logsdon, G. A. *et al.* (Oct. 2020). "Long-read human genome sequencing and its applications". In: *Nature Reviews Genetics* 21.10, pp. 597–614 (cit. on p. 37).

Louca, S. *et al.* (Feb. 2019). "A census-based estimate of Earth's bacterial and archaeal diversity". en. In: *PLOS Biology* 17.2. Publisher: Public Library of Science, e3000106 (cit. on p. 22).

Ma, B. *et al.* (Mar. 1, 2002). "PatternHunter: faster and more sensitive homology search". In: *Bioinformatics* 18.3, pp. 440–445 (cit. on pp. 8, 82).

Ma, Z. (*et al.* (Dec. 2019). "Hybrid assembly of ultra-long Nanopore reads augmented with 10x-Genomics contigs: Demonstrated with a human genome". en. In: *Genomics* 111.6, pp. 1896–1901 (cit. on p. 36).

Mak, D. *et al.* (July 15, 2006). "Indel seeds for homology search". In: *Bioinformatics* 22.14, e341–e349 (cit. on p. 86).

Mäkinen, V. and G. Navarro (Nov. 22, 2007). "Rank and select revisited and extended". In: *Theoretical Computer Science*. The Burrows-Wheeler Transform 387.3, pp. 332–347 (cit. on p. 66).

Mallik, A. and L. Ilie (May 1, 2021). "ALeS: adaptive-length spaced-seed design". In: *Bioinformatics* 37.9, pp. 1206–1210 (cit. on p. 85).

Marçais, G., A. L. Delcher, *et al.* (Jan. 26, 2018). "MUMmer4: A fast and versatile genome alignment system". In: *PLoS Computational Biology* 14.1, e1005944 (cit. on p. 53).

Marçais, G. and C. Kingsford (Mar. 15, 2011). "A fast, lock-free approach for efficient parallel counting of occurrences of k-mers". In: *Bioinformatics* 27.6, pp. 764–770 (cit. on p. 68).

Marcelino, V. R. *et al.* (Feb. 27, 2020). "The use of taxon-specific reference databases compromises metagenomic classification". In: *BMC Genomics* 21.1, p. 184 (cit. on pp. 87, 120, 148).

Marchet, C., C. Boucher, *et al.* (Dec. 16, 2020). "Data structures based on k-mers for querying large collections of sequencing data sets". In: *Genome Research*. Company: Cold Spring Harbor Laboratory Press Distributor: Cold Spring Harbor Laboratory Press Institution: Cold Spring Harbor Laboratory Press Label: Cold Spring Harbor Laboratory Press Publisher: Cold Spring Harbor Lab (cit. on pp. 58, 61, 66, 69, 155).

Marchet, C., Z. Iqbal, *et al.* (July 1, 2020). "REINDEER: efficient indexing of k-mer presence and abundance in sequencing datasets". In: *Bioinformatics* 36 (Supplement_1), pp. i177–i185 (cit. on p. 69).

Marchet, C. *et al.* (Apr. 3, 2021). "BLight: efficient exact associative structure for k-mers". In: *Bioinformatics* (btab217) (cit. on p. 69).

Martinović, A. *et al.* (Aug. 2020). "Streptococcus thermophilus: To Survive, or Not to Survive the Gastrointestinal Tract, That Is the Question!" In: *Nutrients* 12.8. Number: 8 Publisher: Multidisciplinary Digital Publishing Institute, p. 2175 (cit. on pp. 4, 43).

McMaster, G. K. *et al.* (Apr. 1, 1981). "Comparison of canine parvovirus with mink enteritis virus by restriction site mapping". In: *Journal of Virology* 38.1. Publisher: American Society for Microbiology, pp. 368–371 (cit. on p. 26).

Meier-Kolthoff, J. P. *et al.* (2014). "Complete genome sequence of DSM 30083(T), the type strain (U5/41(T)) of Escherichia coli, and a proposal for delineating subspecies in microbial taxonomy". eng. In: *Standards in Genomic Sciences* 9, p. 2 (cit. on p. 31).

Menzel, P. *et al.* (Apr. 13, 2016). "Fast and sensitive taxonomic classification for metagenomics with Kaiju". In: *Nature Communications* 7.1, p. 11257 (cit. on p. 55).

Michener, C. D. *et al.* (1970). *Systematics in Support of Biological Research*. en. Google-Books-ID: XTsrAAAAYAAJ. Division of Biology and Agriculture, National Research Council (cit. on p. 21).

Miele, V. *et al.* (Apr. 22, 2011). "Ultra-fast sequence clustering from similarity networks with SiLiX". In: *BMC Bioinformatics* 12, p. 116 (cit. on p. 91).

Mitchell, T. J. (Dec. 2003). "The pathogenesis of streptococcal infections: from Tooth decay to meningitis". In: *Nature Reviews Microbiology* 1.3, pp. 219–230 (cit. on p. 44).

Mohamadi, H. *et al.* (May 1, 2017). "ntCard: a streaming algorithm for cardinality estimation in genomics data". In: *Bioinformatics* 33.9, pp. 1324–1330 (cit. on p. 118).

Moore, W. E. C. *et al.* (Oct. 1987). "Report of the Ad Hoc Committee on Reconciliation of Approaches to Bacterial Systematics". en. In: *International Journal of Systematic and Evolutionary Microbiology* 37.4, pp. 463–464 (cit. on p. 26).

Morgenstern, B. *et al.* (Feb. 11, 2015). "Estimating evolutionary distances between genomic sequences from spaced-word matches". In: *Algorithms for Molecular Biology* 10.1, p. 5 (cit. on p. 83).

Morisse, P., C. Marchet, *et al.* (Jan. 12, 2021). "Scalable long read self-correction and assembly polishing with multiple sequence alignment". In: *Scientific Reports* 11.1, p. 761 (cit. on p. 74).

Morisse, P. *et al.* (Dec. 15, 2018). "Hybrid correction of highly noisy long reads using a variable-order de Bruijn graph". In: *Bioinformatics (Oxford, England)* 34.24, pp. 4213–4222 (cit. on p. 73).

Muggli, M. D., A. Bowe, *et al.* (Oct. 15, 2017). "Succinct colored de Bruijn graphs". In: *Bioinformatics (Oxford, England)* 33.20, pp. 3181–3187 (cit. on pp. 61, 69).

Muggli, M. D. *et al.* (July 15, 2019). "Building large updatable colored de Bruijn graphs via merging". In: *Bioinformatics (Oxford, England)* 35.14, pp. i51–i60 (cit. on p. 69).

Mustafa, H. *et al.* (2017). "Metannot: A succinct data structure for compression of colors in dynamic de Bruijn graphs". In: *bioRxiv.* Accepted: 2018-01-30T11:11:41Z Publisher: Cold Spring Harbor Laboratory, p. 236711 (cit. on p. 69).

Muyzer, G. *et al.* (1995). "Phylogenetic relationships of Thiomicrospira species and their identification in deep-sea hydrothermal vent samples by denaturing gradient gel electrophoresis of 16S rDNA fragments | SpringerLink". In: *Archives of Microbiology* 164, pp. 165–172 (cit. on p. 28).

Nathans, D. and H. O. Smith (June 1975). "Restriction Endonucleases in the Analysis and Restructuring of DNA Molecules". In: *Annual Review of Biochemistry* 44.1. Publisher: Annual Reviews, pp. 273–293 (cit. on p. 34).

Navarro-Garcia, F. (Dec. 2014). "Escherichia coli O104:H4 Pathogenesis: an Enteroaggregative E. coli/Shiga Toxin-Producing E. coli Explosive Cocktail of High Virulence". eng. In: *Microbiology Spectrum* 2.6 (cit. on pp. 2, 31).

Needleman, S. B. and C. D. Wunsch (Mar. 28, 1970). "A general method applicable to the search for similarities in the amino acid sequence of two proteins". In: *Journal of Molecular Biology* 48.3, pp. 443–453 (cit. on p. 49).

Noé, L. (Feb. 14, 2017). "Best hits of 11110110111: model-free selection and parameter-free sensitivity calculation of spaced seeds". In: *Algorithms for Molecular Biology* 12.1, p. 1 (cit. on pp. 8, 85).

Noé, L. and G. Kucherov (July 1, 2005). "YASS: enhancing the sensitivity of DNA similarity search". In: *Nucleic Acids Research* 33 (suppl_2), W540–W543 (cit. on p. 85).

Noé, L. and D. E. Martin (Dec. 1, 2014). "A Coverage Criterion for Spaced Seeds and Its Applications to Support Vector Machine String Kernels and k-Mer Distances". In: *Journal of Computational Biology* 21.12. Publisher: Mary Ann Liebert, Inc., publishers, pp. 947–963 (cit. on pp. 84, 85).

Ondov, B. D. *et al.* (June 20, 2016). "Mash: fast genome and metagenome distance estimation using MinHash". In: *Genome Biology* 17.1, p. 132 (cit. on pp. 89, 90, 92).

Ørskov, F. and I. Ørskov (Aug. 1983). "Summary of a Workshop on the Clone Concept in the Epidemiology, Taxonomy, and Evolution of the Enterobacteriaceae and other Bacteria". In: *The Journal of Infectious Diseases* 148.2, pp. 346–357 (cit. on p. 32).

Ottaviano, G. and R. Venturini (July 3, 2014). "Partitioned Elias-Fano indexes". In: *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval.* SIGIR '14. New York, NY, USA: Association for Computing Machinery, pp. 273–282 (cit. on p. 61).

Ounit, R. and S. Lonardi (Dec. 15, 2016). "Higher classification sensitivity of short metagenomic reads with CLARK-S". In: *Bioinformatics (Oxford, England)* 32.24, pp. 3823–3825 (cit. on pp. 71, 83).

Ounit, R., S. Wanamaker, *et al.* (Mar. 25, 2015). "CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers". In: *BMC Genomics* 16.1, p. 236 (cit. on pp. 70, 71, 83).

Pandey, P., F. Almodaresi, *et al.* (Aug. 22, 2018). "Mantis: A Fast, Small, and Exact Large-Scale Sequence-Search Index". In: *Cell Systems* 7.2, 201–207.e4 (cit. on p. 69).

Pandey, P., M. A. Bender, *et al.* (May 9, 2017). "A General-Purpose Counting Filter: Making Every Bit Count". In: *Proceedings of the 2017 ACM International Conference on Management of Data.* SIGMOD '17. New York, NY, USA: Association for Computing Machinery, pp. 775–787 (cit. on p. 60).

Parayre, S. *et al.* (June 2007). "Easy DNA extraction method and optimisation of PCR-Temporal Temperature Gel Electrophoresis to identify the predominant high and low GC-content bacteria from dairy products". In: *Journal of Microbiological Methods* 69.3, pp. 431–441 (cit. on p. 141).

Parks, D. H., M. Chuvochina, P.-A. Chaumeil, *et al.* (Sept. 2020). "A complete domain-to-species taxonomy for Bacteria and Archaea". en. In: *Nature Biotechnology* 38.9. Number: 9 Publisher: Nature Publishing Group, pp. 1079–1086 (cit. on p. 22).

Parks, D. H., M. Chuvochina, D. W. Waite, *et al.* (Nov. 2018). "A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life". eng. In: *Nature Biotechnology* 36.10, pp. 996–1004 (cit. on p. 22).

Petrucci, E. *et al.* (Feb. 1, 2020). "Iterative Spaced Seed Hashing: Closing the Gap Between Spaced Seed Hashing and k-mer Hashing". In: *Journal of Computational Biology* 27.2. Publisher: Mary Ann Liebert, Inc., publishers, pp. 223–233 (cit. on p. 84).

Pierce, N. T. *et al.* (July 4, 2019). "Large-scale sequence comparisons with sourmash". In: *F1000Research* 8, p. 1006 (cit. on p. 89).

Piro, V. C., T. H. Dadi, *et al.* (July 1, 2020). "ganon: precise metagenomics classification against large and up-to-date sets of reference sequences". In: *Bioinformatics* 36 (Supplement_1). Publisher: Oxford Academic, pp. i12–i20 (cit. on pp. 70, 78).

Piro, V. C. *et al.* (Aug. 1, 2016). "DUDes: a top-down taxonomic profiler for metagenomics". In: *Bioinformatics* 32.15, pp. 2272–2280 (cit. on pp. 54, 55).

Pollard, M. O. *et al.* (Aug. 1, 2018). "Long reads: their purpose and place". In: *Human Molecular Genetics* 27 (R2), R234–R241 (cit. on p. 37).

Poretsky, R. *et al.* (Apr. 2014). "Strengths and Limitations of 16S rRNA Gene Amplicon Sequencing in Revealing Temporal Microbial Community Dynamics". In: *PLoS ONE* 9.4, e93827 (cit. on p. 31).

Poulin-Laprade, D. *et al.* (Mar. 26, 2021). "Resistance Determinants and Their Genetic Context in Enterobacteria from a Longitudinal Study of Pigs Reared under Various Husbandry Conditions". In: *Applied and Environmental Microbiology* 87.8, e02612–20 (cit. on pp. 16, 150, 151).

Radke-Mitchell, L. C. and W. E. Sandine (Oct. 1, 1986). "Influence of Temperature on Associative Growth of Streptococcus thermophilus and Lactobacillus bulgaricus1". In: *Journal of Dairy Science* 69.10, pp. 2558–2568 (cit. on p. 43).

Raman, R. *et al.* (Nov. 2007). "Succinct Indexable Dictionaries with Applications to Encoding k-ary Trees, Prefix Sums and Multisets". In: *ACM Transactions on Algorithms* 3.4, p. 43. arXiv: 0705.0552 (cit. on pp. 61, 66).

Rasko, D. A. *et al.* (Oct. 15, 2008). "The Pangenome Structure of Escherichia coli: Comparative Genomic Analysis of E. coli Commensal and Pathogenic Isolates". In: *Journal of Bacteriology* 190.20. Publisher: American Society for Microbiology, pp. 6881–6893 (cit. on p. 45).

Reinert, K. *et al.* (2015). "Alignment of Next-Generation Sequencing Reads". In: *Annual Review of Genomics and Human Genetics* 16, pp. 133–151 (cit. on p. 78).

Relman, D. A. (May 1999). "The Search for Unrecognized Pathogens". en. In: *Science* 284.5418. Publisher: American Association for the Advancement of Science Section: Special Viewpoints, pp. 1308–1310 (cit. on p. 28).

Rhoads, A. and K. F. Au (Oct. 1, 2015). "PacBio Sequencing and Its Applications". In: *Genomics, Proteomics & Bioinformatics*. SI: Metagenomics of Marine Environments 13.5, pp. 278–289 (cit. on p. 37).

Robidou, L. and P. Peterlongo (May 2021). *findere: fast and precise approximate membership query* (cit. on p. 155).

Roosaare, M. *et al.* (2017). "StrainSeeker: fast identification of bacterial strains from raw sequencing reads using user-provided guide trees". In: *PeerJ* 5, e3353 (cit. on pp. 6, 71).

Salmela, L. and E. Rivals (Dec. 15, 2014). "LoRDEC: accurate and efficient long read error correction". In: *Bioinformatics* 30.24, pp. 3506–3514 (cit. on p. 73).

Salmela, L., R. Walve, *et al.* (Mar. 15, 2017). "Accurate self-correction of errors in long reads using de Bruijn graphs". In: *Bioinformatics* 33.6, pp. 799–806 (cit. on p. 74).

Sandberg, R. *et al.* (Aug. 2001). "Capturing Whole-Genome Characteristics in Short Sequences Using a Naïve Bayesian Classifier". In: *Genome Research* 11.8, pp. 1404–1409 (cit. on pp. 80, 122).

Sanger, F. *et al.* (Dec. 1977). "DNA sequencing with chain-terminating inhibitors". In: *Proceedings of the National Academy of Sciences of the United States of America* 74.12, pp. 5463–5467 (cit. on p. 33).

Schleifer, K. H. (Dec. 2009). “Classification of Bacteria and Archaea: Past, present and future”. en. In: *Systematic and Applied Microbiology* 32.8, pp. 533–542 (cit. on p. 22).

Schloss, P. D. *et al.* (Dec. 2009). “Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities”. In: *Applied and Environmental Microbiology* 75.23, pp. 7537–7541 (cit. on p. 28).

Schulz, H. N. *et al.* (Apr. 1999). “Dense populations of a giant sulfur bacterium in Namibian shelf sediments”. eng. In: *Science (New York, N.Y.)* 284.5413, pp. 493–495 (cit. on p. 20).

Segata, N. *et al.* (June 10, 2012). “Metagenomic microbial community profiling using unique clade-specific marker genes”. In: *Nature Methods* 9.8, pp. 811–814 (cit. on pp. 53, 55).

Shendure, J. and H. Ji (Oct. 2008). “Next-generation DNA sequencing”. en. In: *Nature Biotechnology* 26.10, pp. 1135–1145 (cit. on pp. 33, 34, 36).

Singh, A. (May 2, 2020). *Probabilistic Approach to Understand Errors in Sequencing and its based Applications.* Company: Cold Spring Harbor Laboratory Distributor: Cold Spring Harbor Laboratory Label: Cold Spring Harbor Laboratory Section: New Results Type: article, p. 2020.05.02.073650 (cit. on p. 79).

Smith, T. F. and M. S. Waterman (Mar. 25, 1981). “Identification of common molecular subsequences”. In: *Journal of Molecular Biology* 147.1, pp. 195–197 (cit. on p. 49).

Solomon, B. and C. Kingsford (Mar. 2016). “Fast search of thousands of short-read sequencing experiments”. In: *Nature Biotechnology* 34.3, pp. 300–302 (cit. on pp. 62, 63, 69).

— (July 1, 2018). “Improved Search of Large Transcriptomic Sequencing Databases Using Split Sequence Bloom Trees”. In: *Journal of Computational Biology* 25.7, pp. 755–765 (cit. on pp. 63, 66, 69, 110).

Sović, I. *et al.* (Apr. 15, 2016). “Fast and sensitive mapping of nanopore sequencing reads with GraphMap”. In: *Nature Communications* 7.1, p. 11307 (cit. on pp. 74, 86).

Stackebrandt, E. (May 2002). “Report of the ad hoc committee for the re-evaluation of the species definition in bacteriology”. en. In: *INTERNATIONAL JOURNAL OF SYSTEMATIC AND EVOLUTIONARY MICROBIOLOGY* 52.3, pp. 1043–1047 (cit. on p. 28).

Staden, R. (June 1979). “A strategy of DNA sequencing employing computer programs”. In: *Nucleic Acids Research* 6.7, pp. 2601–2610 (cit. on p. 32).

Stoler, N. and A. Nekrutenko (Mar. 1, 2021). “Sequencing error profiles of Illumina sequencing instruments”. In: *NAR Genomics and Bioinformatics* 3.1 (cit. on p. 36).

Subedi, D. *et al.* (Oct. 23, 2018). “Comparative genomics of clinical strains of Pseudomonas aeruginosa strains isolated from different geographic sites”. In: *Scientific Reports* 8.1, p. 15668 (cit. on p. 54).

Sukumaran, J. and M. T. Holder (June 15, 2010). “DendroPy: a Python library for phylogenetic computing”. In: *Bioinformatics* 26.12, pp. 1569–1571 (cit. on p. 94).

Sun, C. *et al.* (Mar. 23, 2017). “AllSome Sequence Bloom Trees”. In: *bioRxiv*. Publisher: Cold Spring Harbor Laboratory Section: New Results, p. 090464 (cit. on pp. 63, 69).

Sunagawa, S. *et al.* (Dec. 2013). “Metagenomic species profiling using universal phylogenetic marker genes”. In: *Nature Methods* 10.12, pp. 1196–1199 (cit. on p. 55).

Taiyaki (Sept. 30, 2021). original-date: 2019-03-12T16:23:55Z (cit. on p. 160).

Teeling, H. *et al.* (2004). “Application of tetranucleotide frequencies for the assignment of genomic fragments”. In: *Environmental Microbiology* 6.9. _eprint: https://sfamjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1462-2920.2004.00624.x, pp. 938–947 (cit. on p. 122).

Tenover, F. C. *et al.* (Sept. 1995). “Interpreting chromosomal DNA restriction patterns produced by pulsed-field gel electrophoresis: criteria for bacterial strain typing”. eng. In: *Journal of Clinical Microbiology* 33.9, pp. 2233–2239 (cit. on p. 30).

Terzaghi, B. E. and W. E. Sandine (June 1975). “Improved Medium for Lactic Streptococci and Their Bacteriophages”. In: *Applied Microbiology* 29.6, pp. 807–813 (cit. on p. 127).

Tettelin, H. (Dec. 2004). “Streptococcal genomes provide food for thought”. In: *Nature Biotechnology* 22.12, pp. 1523–1524 (cit. on p. 44).

Thompson, C. C. *et al.* (Mar. 1, 2013). “Streptococcal taxonomy based on genome sequence analyses”. In: *F1000Research* 2, p. 67 (cit. on p. 28).

Tindall, B. J. *et al.* (n.d.). "Notes on the characterization of prokaryote strains for taxonomic purposes". In: *International Journal of Systematic and Evolutionary Microbiology* 60.1 (). Publisher: Microbiology Society, pp. 249–266 (cit. on p. 28).

Titarenko, V. and S. Titarenko (June 10, 2021). *VSTseed: periodic spaced seeds for reads with substitutions*. Company: Cold Spring Harbor Laboratory Distributor: Cold Spring Harbor Laboratory Label: Cold Spring Harbor Laboratory Section: New Results Type: article, p. 2021.06.09.447791 (cit. on p. 85).

Toft, C. and S. G. E. Andersson (July 2010). "Evolutionary microbial genomics: insights into bacterial host adaptation". In: *Nature Reviews Genetics* 11.7, pp. 465–475 (cit. on p. 54).

Truong, D. T. *et al.* (Apr. 2017). "Microbial strain-level population structure and genetic diversity from metagenomes". In: *Genome Research* 27.4, pp. 626–638 (cit. on pp. 54, 55).

Uriot, O. *et al.* (Oct. 1, 2017). "Streptococcus thermophilus: From yogurt starter to a new promising probiotic candidate?" In: *Journal of Functional Foods* 37, pp. 74–89 (cit. on pp. 4, 43).

Vallenet, D. *et al.* (Jan. 8, 2020). "MicroScope: an integrated platform for the annotation and exploration of microbial gene functions through genomic, pangenomic and metabolic comparative analysis". In: *Nucleic Acids Research* 48 (D1), pp. D579–D589 (cit. on pp. 8, 91).

Van Rossum, T. *et al.* (Sept. 2020). "Diversity within species: interpreting strains in microbiomes". In: *Nature Reviews. Microbiology* 18.9, pp. 491–506 (cit. on pp. 136, 156).

Vandamme, P. *et al.* (June 1996). "Polyphasic taxonomy, a consensus approach to bacterial systematics". en. In: *Microbiological Reviews* 60.2 (cit. on p. 22).

Voskoboynik, A. *et al.* (July 2013). "The genome sequence of the colonial chordate, Botryllus schlosseri". In: *eLife* 2. Ed. by M. E. Bronner. Publisher: eLife Sciences Publications, Ltd, e00569 (cit. on p. 36).

Wagner, J. *et al.* (Nov. 14, 2016). "Evaluation of PacBio sequencing for full-length bacterial 16S rRNA gene classification". In: *BMC Microbiology* 16.1, p. 274 (cit. on p. 37).

Wenger, A. M. *et al.* (Oct. 2019). "Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome". In: *Nature biotechnology* 37.10, pp. 1155–1162 (cit. on p. 37).

Wick, R. R., L. M. Judd, C. L. Gorrie, *et al.* (June 8, 2017). "Unicycler: Resolving bacterial genome assemblies from short and long sequencing reads". In: *PLOS Computational Biology* 13.6. Publisher: Public Library of Science, e1005595 (cit. on pp. 4, 44).

Wick, R. R. *et al.* (June 24, 2019). "Performance of neural network basecalling tools for Oxford Nanopore sequencing". In: *Genome Biology* 20.1, p. 129 (cit. on pp. 38, 39).

Woese, C. R. (June 1987). "Bacterial evolution." In: *Microbiological Reviews* 51.2, pp. 221–271 (cit. on p. 27).

Wood, D. E. and S. L. Salzberg (2014). "Kraken: ultrafast metagenomic sequence classification using exact alignments". In: *Genome Biology* 15.3, R46 (cit. on pp. 6, 68, 71).

Wood, D. E. *et al.* (Nov. 28, 2019). "Improved metagenomic analysis with Kraken 2". In: *Genome Biology* 20.1, p. 257 (cit. on pp. 6, 68, 71).

Yang, W. *et al.* (2019). "UltraStrain: An NGS-Based Ultra Sensitive Strain Typing Method for Salmonella enterica". In: *Frontiers in Genetics* 10, p. 276 (cit. on p. 98).

Yarza, P. *et al.* (Sept. 2014). "Uniting the classification of cultured and uncultured bacteria and archaea using 16S rRNA gene sequences". en. In: *Nature Reviews Microbiology* 12.9. Number: 9 Publisher: Nature Publishing Group, pp. 635–645 (cit. on p. 22).

Yu, Y. *et al.* (Oct. 19, 2018). "SeqOthello: querying RNA-seq experiments at scale". In: *Genome Biology* 19.1, p. 167 (cit. on pp. 60, 62, 69).

Zhang, H. *et al.* (Dec. 21, 2020). "A comprehensive evaluation of long read error correction methods". In: *BMC Genomics* 21.6, p. 889 (cit. on p. 73).

Zheng, G. X. Y. *et al.* (Mar. 2016). "Haplotyping germline and cancer genomes with high-throughput linked-read sequencing". en. In: *Nature Biotechnology* 34.3, pp. 303–311 (cit. on p. 36).

Zheng, J. *et al.* (2020). "A taxonomic note on the genus Lactobacillus: Description of 23 novel genera, emended description of the genus Lactobacillus Beijerinck 1901, and union of Lactobacillaceae and Leuconostocaceae". In: *International Journal of Systematic and Evolutionary Microbiology* 70.4. Publisher: Microbiology Society, pp. 2782–2858 (cit. on pp. 15, 43).

List of published contributions

Siekaniec, G. *et al.* (2021). “Identification of isolated or mixed strains from long reads: a challenge met on Streptococcus thermophilus using a MinION sequencer”. In: *Microbial Genomics* 7.11. Publisher: Microbiology Society, p. 000654 (cit. on pp. 10, 11, 80, 126, 158).

DOCTORAT BRETAGNE LOIRE MATHSTIC

UNIVERSITÉ DE RENNES 1

Titre : Identification de souches d'une espèce bactérienne à partir de longues lectures

Mots-clés : Bioinformatique, Identification de souches bactériennes, *Streptococcus thermophilus*, lecture longue, indexation, graine espacée, ORI

Résumé : Actuellement, l'identification à partir de séquences génomiques de souches d'une espèce bactérienne présentes dans un échantillon reste un processus complexe et chronophage. Cette difficulté provient de la grande similarité génomique entre ces souches. Cependant, pouvoir les différencier rapidement est crucial dans de nombreux domaines, que ce soit en agroalimentaire (comme *Streptococcus thermophilus*) ou en santé publique. Récemment, la troisième génération de technologies de séquençage, et plus particulièrement les séquenceurs d'Oxford Nanopore Technologies, permettent d'obtenir des séquences longues mais erronées à partir d'échantillons contenant des souches bactériennes. Ces lectures contiennent plus d'informations que les anciennes lectures courtes de seconde génération. Or, actuellement, il existe encore assez peu de logiciels bioinformatiques développés pour identifier les souches bactériennes à partir de longues lectures erronées.

Cette thèse propose donc une nouvelle méthode d'identification de souches bactériennes basée sur l'hypothèse qu'une lecture nanopore est suffisamment longue pour permettre de distinguer une souche (ou un groupe de souches) des autres. Cette méthode utilise une technique d'indexation particulièrement compacte d'une base de données de génomes connus. Elle repose également sur l'utilisation d'une graine espacée afin de rechercher les séquences dans l'index en étant moins sensible aux erreurs des lectures longues. La méthode est implémentée dans un logiciel appelé ORI (Oxford nanopore Reads Identification) qui a montré des résultats robuste d'identification bactérienne sur des données réelles de *Streptococcus thermophilus*.

Title: Identification of strains of a bacterial species from long reads

Keywords: Bioinformatics, Bacterial strains identification, *Streptococcus thermophilus*, long read, indexing, spaced seed, ORI

Abstract: Currently, the identification from genomic sequences of strains of a bacterial species present in a sample remains a complex and time consuming process. This difficulty comes from the genomic similarity between these strains. However, being able to differentiate them quickly is crucial in many fields, whether in agri-food (such as *Streptococcus thermophilus*) or in public health.

Recently, the third generation of sequencing technologies, and more specifically the Oxford Nanopore Technologies sequencers, make it possible to obtain long but erroneous sequences from samples containing bacterial strains. These reads contain more information than the short reads from the second generation. However, currently, there are still few bioinformatics softwares developed to identify bacterial strains from erroneous long reads.

This thesis therefore proposes a new method of bacterial strain identification based on the assumption that a nanopore read is long enough to distinguish one strain (or group of strains) from others. This method uses a particularly compact indexing technique of a known genome database. It also relies on the use of a spaced seed in order to search for sequences in the index while being less sensitive to long reads errors. The method is implemented in a software called ORI (Oxford nanopore Reads Identification) which has shown robust bacterial identification results on real data of *Streptococcus thermophilus*.